МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ОБРАЗОВАНИЕ

НАЦИОНАЛЬНЫЕ ПРОЕКТЫ РОССИИ

СЕТЬ ЦЕНТРОВ ЦИФРОВОГО ОБРАЗОВАНИЯ ДЕТЕЙ «IT-КУБ»

РЕАЛИЗАЦИЯ ДОПОЛНИТЕЛЬНОЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ ПО ТЕМАТИЧЕСКОМУ НАПРАВЛЕНИЮ «МОБИЛЬНАЯ РАЗРАБОТКА» С ИСПОЛЬЗОВАНИЕМ ОБОРУДОВАНИЯ ЦЕНТРА ЦИФРОВОГО ОБРАЗОВАНИЯ ДЕТЕЙ «IT-КУБ»

МОСКВА 2021

С. Г. Григорьев

Р.А. Сабитов

Г. С. Смирнова

Ш.Р. Сабитов

Реализация дополнительной общеобразовательной программы по тематическому направлению «Мобильная разработка» с использованием оборудования центра цифрового образования детей «IT-куб»

Методическое пособие

под ред. С. Г. Григорьева

Москва, 2021

Содержание

Пояснительная записка

Трудно представить современный мир без мобильных устройств и разного рода гаджетов. То, что казалось ещё 20 лет назад фантастикой, сейчас распахнуло двери и стремительно врывается в наш мир, который даже по человеческим меркам ещё совсем недавно пользовался дисковыми телефонными аппаратами. А сейчас всё вокруг неумолимо и стремительно переходит к новому технологическому укладу. Согласно Элвину Тоффлеру, следующий мировой технологический и социальный уклад установит ценность человеческого ума и талантов как высший приоритет. При этом новом укладе мобильные устройства являются не только предтечей и воплощением будущего, и должны быть не только инструментом постижения мира, но и проводником, способствующим нашей трансформации. Посредством этих инструментов человечество должно преодолеть непростые ступени нового мира и застолбить своё место в грядущем новом мире.

Первые мобильные приложения появились еще в далёком 1993 году. А первый мобильный телефон появился за 20 лет до этого, в ещё более далёком 1973 году, когда 3 апреля два инженера-разработчика компаний Bell Labs и Motorola осуществили первый разговор.

На сегодняшний день мир мобильной разработки представлен двумя основными операционными системами и технологиями на их базе: Android и iOS. С большим отрывом превалирует Android.

Средства разработки под ОС Андроид можно поделить на две группы. Первая группа использует непосредственно Android SDK (пакет разработчика Андроид) и языки, соответственно Java или Kotlin. В этом случае разработка ведётся в среде Android Studio (реже используется Eclipse, или, как вариант, можно использовать обычную версию IntelliJ и настроить специальный плагин для платформы Андроид). Удобнее использовать среду Android Studio, которая является специальной сборкой IntelliJ для создания мобильных приложений Андроид.

Вторая группа средств активно развивается и представляет мобильную разработку на базе фреймворков. Например, для разработки Android-приложений уже давно существует фреймворк Xamarin, в котором можно программировать на базе .Net-технологий. Также можно упомянуть React.js, с помощью которого можно создавать оптимизированные по потреблению ресурсов Андроид-приложения. Существуют и прочие технологии, которые позволяют подгонять Web-приложения под формат мобильных приложений. Стоит отметить Flutter, как средство быстрого прототипирования малоэкранных приложений.

В данном курсе рассматривается разработка Андроид-приложений на базе облачного средства AppInventor. AppInventor находится на промежуточной стадии между no code платформой и фреймворком для разработки мобильных Android-приложений. АИ является no code платформой, потому что можно создать мобильное приложение, не запрограммировав ни строчки. В то же время АИ предоставляет достаточно большой механизм расширений и плагинов, которые сближают функционал АИ с фреймворками.

Для достижения поставленной цели планируется выполнение следующих задач:

Образовательные:

- Формировать общее представление о создании мобильных приложений на базе платформы Андроид.
- Формировать представления о структуре и функционировании среды App Inventor.
- Формировать умения и навыки построения различных видов алгоритмов в среде АИ.
- Формировать умение использовать инструменты и компоненты среды АИ для создания мобильных приложений.

- Формировать умения создавать типовые мобильные приложения.
- Формировать ключевые компетенции проектной и исследовательской деятельности.

Развивающие:

- Развивать алгоритмическое и логическое мышление.
- Развивать умение постановки задачи, выделения основных объектов, математическое модели задачи.
- Развивать умение поиска необходимой учебной информации.
- — Формировать мотивацию к изучению программирования.

Воспитательные:

- Воспитывать умение работать индивидуально и в группе для решения поставленной задачи.
- Воспитывать трудолюбие, упорство, желание добиваться поставленной цели.
- Воспитывать информационную культуру.

Программа рассчитана на учащихся в возрасте от 11 до 15 лет, не требует предварительных знаний и входного тестирования.

Занятия проводятся в группах до 12 человек, продолжительность занятия 45 минут, общая продолжительность программы — 36 часов.

Нормативная база

1. Конституция Российской Федерации (принята всенародным голосованием 12.12.1993 с изменениями, одобренными в ходе общероссийского голосования 01.07.2020) — URL: http://www.consultant.ru/document/cons_doc_LAW_28399/ (дата обращения: 10.03.2021).

2. Федеральный закон от 29.12.2012 № 273-ФЗ (ред. от 31.07.2020) «Об образовании в Российской Федерации» (с изм. и доп., вступ. в силу с 01.09.2020) — URL: http://www.consultant.ru/document/cons_doc_LAW_140174 (дата обращения: 28.09.2020).

3. Паспорт национального проекта «Образование» (утв. президиумом Совета при Президенте РФ по стратегическому развитию и национальным проектам, протокол от 24.12.2018 № 16) — URL: //https://login.consultant.ru link ?req=doc&base=LAW-&n=319308&demo=1 (дата обращения: 10.03.2021).

4. Государственная программа Российской Федерации «Развитие образования» (Утверждена Постановлением Правительства РФ от 26.12.2017 № 1642 (ред. от 22.02.2021) «Об утверждении государственной программы Российской Федерации «Развитие образования» — URL: http: //www.consultant.ru document cons_doc_LAW_286474 (дата обращения: 10.03.2021).

5. Стратегия развития воспитания в Российской Федерации на период до 2025 года (Утверждена распоряжением Правительства РФ от 29.05.2015 № 996-р «Об утверждении Стратегии развития воспитания в Российской Федерации на период до 2025 года») — URL: http://www.consultant.ru/document/cons_doc_LAW_180402/ — (дата обращения: 10.03.2021).

6. Профессиональный стандарт «Педагог (педагогическая деятельность в дошкольном, начальном общем, основном общем, среднем общем образовании), (воспитатель, учитель)» (ред. от 16.06.2019 г.) (Приказ Министерства труда и социальной защиты РФ от 18 октября 2013 г. № 544н, с изменениями, внесенными приказом Министерства труда и соцзащиты РФ от 25 декабря 2014 г. № 1115н и от 5 августа 2016 г. № 422н) —URL: // http://профстандартпедагога.рф — (дата обращения: 10.03.2021).

7. Профессиональный стандарт «Педагог дополнительного образования детей и взрослых» (Приказ Министерства труда и социальной защиты РФ от 5 мая 2018 г. № 298н «Об утверждении профессионального стандарта «Педагог дополнительного образования детей и взрослых») — URL: //https://profstandart.rosmintrud.ru/obshchiy-informatsionnyy-blok/natsionalnyy-reestr-professionalnykh-standartov/reestr-professionalnykh-standartov/index.php?ELEMENT_ID=48583 (дата обращения: 10.03.2021).

8. Федеральный государственный образовательный стандарт основного общего образования (утверждён приказом Министерства образования и науки Российской Федерации от 17 декабря 2010 г. № 1897) (ред. 21.12.2020) — URL: https://fgos.ru (дата обращения: 10.03.2021).

9. Федеральный государственный образовательный стандарт среднего общего образования (утверждён приказом Министерства образования и науки Российской Федерации от 17 мая 2012 г. № 413) (ред. 11.12.2020) — URL: https://fgos.ru (дата обращения: 10.03.2021).

10. Методические рекомендации по созданию и функционированию детских технопарков «Кванториум» на базе общеобразовательных организаций (утверждены распоряжением Министерства просвещения Российской Федерации от 12 января 2021 г. № Р-4) — URL: http://www.consultant.ru/document/cons_doc_LAW_374695/ (дата обращения: 10.03.2021).

11. Методические рекомендации по созданию и функционированию центров цифрового образования «IT-куб» (утверждены распоряжением Министерства просвещения Российской Федерации от 12 января 2021 г. № Р-5) — URL: http://www.consultant.ru/document/cons_doc_LAW_374572/ (дата обращения: 10.03.2021).

12. Методические рекомендации по созданию и функционированию в общеобразовательных организациях, расположенных в сельской местности и малых городах, центров образования естественно-научной и технологической направленностей («Точка роста») — (утверждены распоряжением Министерства просвещения Российской Федерации от 12 января 2021 г. № Р-6) — URL: http://www.consultant.ru/document/cons_doc_LAW_374694/ (дата обращения: 10.03.2021).

Основные понятия и термины

Активности — это специальные классы, представляющие и контролирующие работу одного экрана приложения Андроид.

Блок АИ — это визуальный программный блок, похожий на пазл. Обычно работает в комбинации с другими блоками.

Класс — это базовая структурная единица языка Java. Представлена в виде файла с расширением *.java.

Компонента АИ — это визуальная компонента, размещаемая на экране приложения. Может быть видимой, как кнопка или изображение. Иногда бывает невидимой, например компонента, представляющая собой функционал камеры или сенсора.

ЛКМ, ПКМ — это левая, правая кнопка мыши.

ОС Андроид — это операционная система Андроид.

Платформа Андроид — это библиотеки и компоненты для разработки Андроид-приложений.

Приложение для сканирования QR-кода — это специальное мобильное приложение для распознавания QR-кода.

Расположения — это особые компоненты АИ, представляющие собой контейнеры для других компонент.

Сенсоры — это датчики мобильного устройства.

Список (массив) — это упорядоченная изменяемая последовательность элементов различного типа.

Текстура — это изображение, близкое по визуальным свойствам к реальным объектам.

Файл манифеста — это файл с базовыми настройками Андроид-приложения.

Эмулятор — это система программных средств, которая копирует функции мобильного устройства на базе платформы Андроид с целью максимально близкой имитации эмулятором поведения мобильного устройства. Это позволяет запускать Андроид-приложения при отсутствии физического мобильного устройства.

Язык программирования — это набор формальных правил, по которым пишут программы.

AI — App Inventor.

APK — это формат архивных исполняемых файлов-приложений для Android и ряда других операционных систем, основанных на Android. Каждое приложение Android скомпилировано и упаковано в один файл, который включает в себя весь код приложения, ресурсы, активности, файл манифеста и пр.

Google Play — это магазин приложений Google, куда Андроид-разработчики могут выставлять свои приложения. Соответственно пользователи мобильных устройств на базе Андроид могут оттуда скачивать приложения и устанавливать их на свои устройства.

IDE — это интегрированная среда разработки.

IntelliJ IDEA — это интегрированная среда разработки компании JetBrains.

IoT — это Internet of Things (Интернет Вещей).

Java — это объектно-ориентированный язык высокого уровня со строгой типизацией.

JVM — это Java Virtual Machine, виртуальная машина Java, специальная среда для выполнения байт-кода.

QR-код — это двумерный штрихкод.

Структурирование материалов

Содержание обучения может быть представлено следующими подразделами:

- Ознакомление со средой AI
- Создание первого приложения
- Работа с базовыми компонентами и блоками
- Анимация
- Web
- Многоэкранные приложения
- Структуры данных
- Сенсоры
- Отправка сообщений
- Хранилища

Для каждого раздела подготовлены лабораторные работы с необходимым теоретически материалом, заданиями и указаниями к их выполнению. Также имеются дидактическим материалы общей направленности, которые можно использовать при подготовке преподавателей и учащихся к занятиям, при выполнении лабораторных работ.

Рис. 1. Использование оборудования IT-куб при организации занятий по курсу «Мобильная разработка»

Примерная рабочая программа для организации работы по тематическому направлению «Мобильная разработка»

Планируемые результаты освоения программы обучающимися с описанием процедур итоговой и промежуточной аттестации

Как было сказано ранее, целью программы «Мобильная разработка» является развитие умений и навыков создания простых мобильных приложений для ОС Андроид на базе визуального конструктора среды App Inventor, а также развитие алгоритмического мышления учащихся, творческих способностей, аналитических и логических компетенций.

Планируемые результаты обучения

Личностные:

- Формирование умения самостоятельной деятельности.
- Формирование умения работать в команде.
- Формирование коммуникативных навыков.
- Формирование навыков анализа и самоанализа.
- Формирование эстетического отношения к языкам программирования, осознание их выразительных возможностей.
- Формирование целеустремлённости и усидчивости в процессе творческой, исследовательской работы и учебной деятельности.

Предметные:

- Формировать общее представление о создании мобильных приложений на базе платформы Андроид.
- Формировать представления о структуре и функционировании среды App Inventor.
- Формировать умения и навыки построения различных видов алгоритмов с помощью блоков в среде АИ.
- Формировать умение использовать компоненты, блоки и их комбинации в среде АИ для создания мобильных приложений.
- Формировать умения создавать типовые мобильные приложения на базе компонент среды АИ.
- Формировать ключевые компетенции проектной и исследовательской деятельности.

Метапредметные:

- Формирование умения ориентировки в системе знаний.
- Формирование умения выбора наиболее эффективных способов решения задач на компьютере в зависимости от конкретных условий.
- Формирование приёмов проектной деятельности, включая умения видеть проблему, формулировать тему и цель проекта, составлять план своей деятельности, осуществлять действия по реализации плана, соотносить результат своей деятельности с целью, классифицировать, наблюдать, проводить эксперименты, делать выводы и заключения, доказывать, защищать свои идеи, оценивать результаты своей работы.
- Формирование умения распределения времени.
- Формирование умений успешной самопрезентации.

Тематическое планирование

№ п/п	Тема	Содержание	Целевая установка урока	Кол-во часов	Основные виды деятельности обучающихся на уроке/ внеурочном занятии	Используемое оборудование
1	Знакомство со средой АИ. Создание первого проекта	Ознакомление со средой. Установка и запуск эмулятора. Создание первого приложения	Научиться настраивать окружение среды АИ и создавать проекты	2	Наблюдение за работой учителя, самостоятельная работа в среде App Inventor, ответы на контрольные вопросы, участие в дискуссии. Выполнение лабораторных работ	Компьютер, проектор, интерактивная доска
2	Работа с базовыми компонентами интерфейса приложения и блоками	Базовые компоненты разделов Интерфейс пользователя и Расположения. Знакомство с базовыми блоками. Создание типовых приложений	Научиться применять базовые компоненты АИ для построения интерфейса. Научиться использовать основные блоки (переменные, математика, логика, процедуры) для создания программной логики приложений	8	Наблюдение за работой учителя, самостоятельная работа в среде App Inventor, ответы на контрольные вопросы, участие в дискуссии. Выполнение лабораторных работ	Компьютер, проектор, интерактивная доска
3	Анимация	Компоненты Холст, Шар, Спрайт. Создание игр	Научиться использовать компоненты анимации для создания игровых приложений	4	Наблюдение за работой учителя, самостоятельная работа в среде App Inventor, ответы на контрольные вопросы, участие в дискуссии. Выполнение лабораторных работ	Компьютер, проектор, интерактивная доска

Продолжение

№ п/п	Тема	Содержание	Целевая установка урока	Кол-во часов	Основные виды деятельности обучающихся на уроке/ внеурочном занятии	Используемое оборудование
4	Web-приложения	Организация доступа в Интернет при помощи компоненты Web-Просмотрщик	Создание интернет-приложений	2	Наблюдение за работой учителя, самостоятельная работа в среде App Inventor, ответы на контрольные вопросы, участие в дискуссии. Выполнение лабораторных работ	Компьютер, проектор, интерактивная доска
5	Работа с несколькими экранами	Переход и передача информации между экранами	Научиться создавать многоэкранные приложения	4	Наблюдение за работой учителя, самостоятельная работа в среде App Inventor, ответы на контрольные вопросы, участие в дискуссии. Выполнение лабораторных работ	Компьютер, проектор, интерактивная доска
6	Тестирование	Создание приложений	Проверка полученных навыков по теме «Работа с компонентами интерфейса и программными блоками в среде АИ»	2	Наблюдение за работой учителя, самостоятельная работа в среде App Inventor, ответы на контрольные вопросы, участие в дискуссии. Выполнение лабораторных работ	Компьютер, проектор, интерактивная доска
7	Структуры данных	Работа с блоками разделов Dictionary и Массив	Научиться использовать массивы и словари для эффективного управления данными	2	Наблюдение за работой учителя, самостоятельная работа в среде App Inventor, ответы на контрольные вопросы, участие в дискуссии. Выполнение лабораторных работ	Компьютер, проектор, интерактивная доска

В содержание

8	Сенсоры. Передача сообщений	Сенсор местоположения, акселерометр. Отправка сообщений и фото	Изучить базовый функционал среды по отправке СМС и почты, использования камеры, акселерометра	2	Наблюдение за работой учителя, самостоятельная работа в среде App Inventor, ответы на контрольные вопросы, участие в дискуссии. Выполнение лабораторных работ	Компьютер, проектор, интерактивная доска
9	Хранилища данных	Компонента TinyDB	Научиться сохранять и извлекать информацию при помощи локального хранилища	2	Наблюдение за работой учителя, самостоятельная работа в среде App Inventor, ответы на контрольные вопросы, участие в дискуссии. Выполнение лабораторных работ	Компьютер, проектор, интерактивная доска
10	Творческое задание	Создание приложений	Проверка полученных навыков по темам «Компоненты сенсоров и общения», «Хранилища данных»	1	Самостоятельное выполнение контрольных заданий	Компьютер, проектор, интерактивная доска
11	Индивидуальное задание	Разработка индивидуального или группового проекта	Создание индивидуального приложения в среде АИ	6	Самостоятельная индивидуальная или групповая проектная деятельность	Компьютер, проектор, интерактивная доска
11	Итоги	Защита индивидуальных или групповых проектов, подведение итогов курса	Защита проекта	1	Самостоятельная индивидуальная или групповая проектная деятельность	Компьютер, проектор, интерактивная доска
Итого				36		

Содержание и форма организации учебных занятий

Планы учебных занятий

1. Знакомство со средой АИ. Создание первого проекта.

Рекомендуемое количество часов на данную тему — 2 часа.

Планируемые результаты

Предметные: получение навыков работы в среде АИ, освоение основных инструментов среды; получение умений установки MIT AppInventor Tools и запуска эмулятора.

Метапредметные: умение пользоваться справками программ и интернет-поиском; способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные), делать выводы в процессе работы и по её окончании, корректировать намеченный план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи.

Личностные: готовность и способность обучающихся к саморазвитию и личностному самоопределению; сформированность их мотивации к обучению и целенаправленной познавательной деятельности.

Оборудование и материалы: компьютер, презентационное оборудование.

Распределение лабораторных работ:

Занятия 1, 2 — выполнение лабораторной работы 1

2. Работа с компонентами интерфейса и программными блоками в среде АИ

Рекомендуемое количество часов на данную тему — 8 часов.

Планируемые результаты

Предметные: получение навыков работы с базовыми компонентами разделов Интерфейс пользователя и Расположения; получение навыков работы с базовыми блоками разделов Управление, Математика, Логика, Текст, Переменные для организации программной логики мобильных приложений.

Метапредметные: способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные), делать выводы в процессе работы и по её окончании, корректировать намеченный план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи.

Личностные: готовность и способность обучающихся к саморазвитию и личностному самоопределению; сформированность их мотивации к обучению и целенаправленной познавательной деятельности.

Оборудование и материалы: компьютер, презентационное оборудование.

Распределение лабораторных работ:

Занятия 1, 2 — выполнение лабораторной работы 2
Занятия 3, 4 — выполнение лабораторной работы 3
Занятия 5, 6 — выполнение лабораторной работы 4
Занятие 7 — выполнение лабораторной работы 5
Занятие 8 — выполнение лабораторной работы 6

3. Анимация

Рекомендуемое количество часов на данную тему — 4 часа.

Планируемые результаты

Предметные: получение навыков создания интерактивных игровых приложений с использованием компонент анимации в среде АИ.

Метапредметные: способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные), делать выводы в процессе работы и по её окончании, корректировать намеченный план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи.

Личностные: эстетическое отношение к языкам программирования, осознание их выразительных возможностей, готовность и способность обучающихся к саморазвитию и личностному самоопределению; сформированность их мотивации к обучению и целенаправленной познавательной деятельности.

Оборудование и материалы: компьютер, презентационное оборудование.

Распределение лабораторных работ:

Занятия 1, 2 — выполнение лабораторной работы 7

Занятия 3, 4 — выполнение лабораторной работы 8

4. Web-приложения

Рекомендуемое количество часов на данную тему — 2 часа.

Планируемые результаты

Предметные: получение навыков создания мобильных интернет-приложений.

Метапредметные: способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные), делать выводы в процессе работы и по её окончании, корректировать намеченный план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи.

Личностные: эстетическое отношение к языкам программирования, осознание их выразительных возможностей, готовность и способность обучающихся к саморазвитию и личностному самоопределению; сформированность их мотивации к обучению и целенаправленной познавательной деятельности.

Оборудование и материалы: компьютер, презентационное оборудование.

Распределение лабораторных работ:

Занятия 1, 2 — выполнение лабораторной работы 9

5. Работа с несколькими экранами

Рекомендуемое количество часов на данную тему — 4 часа.

Планируемые результаты

Предметные: получение навыков создания многоэкранных приложений; научиться переключаться и передавать данные между экранами.

Метадредметные: способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрица-

тельные), делать выводы в процессе работы и по её окончании, корректировать намеченный план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи.

Личностные: эстетическое отношение к языкам программирования, осознание их выразительных возможностей, готовность и способность обучающихся к саморазвитию и личностному самоопределению; сформированность их мотивации к обучению и целенаправленной познавательной деятельности.

Оборудование и материалы: компьютер, презентационное оборудование.

Распределение лабораторных работ:

Занятия 1, 2 — выполнение лабораторной работы 10

Занятия 3, 4 — выполнение лабораторной работы 11

6. Структуры данных

Рекомендуемое количество часов на данную тему — 2 часа.

Планируемые результаты

Предметные: получить навыки работы с массивами и словарями в среде АИ.

Метапредметные: способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные), делать выводы в процессе работы и по её окончании, корректировать намеченный план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи.

Личностные: эстетическое отношение к языкам программирования, осознание их выразительных возможностей, готовность и способность обучающихся к саморазвитию и личностному самоопределению; сформированность их мотивации к обучению и целенаправленной познавательной деятельности.

Оборудование и материалы: компьютер, презентационное оборудование.

Распределение лабораторных работ:

Занятия 1, 2 — выполнение лабораторной работы 12

7. Сенсоры. Передача сообщений

Рекомендуемое количество часов на данную тему — 2 часа.

Планируемые результаты

Предметные: научиться встраивать функции сенсоров и передачи сообщений в мобильные приложения.

Метапредметные: способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные), делать выводы в процессе работы и по её окончании, корректировать намеченный план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи.

Личностные: эстетическое отношение к языкам программирования, осознание их выразительных возможностей, готовность и способность обучающихся к саморазвитию и личностному самоопределению; готовность и способность обучающихся к саморазвитию

и личностному самоопределению; сформированность их мотивации к обучению и целенаправленной познавательной деятельности.

Оборудование и материалы: компьютер, презентационное оборудование.

Распределение лабораторных работ:

Занятия 1, 2 — выполнение лабораторной работы 13.

8.Хранилища данных

Рекомендуемое количество часов на данную тему — 2 часа.

Планируемые результаты

Предметные: научиться организовывать хранение данных с помощью локальных хранилищ типа TinyDB.

Метапредметные: способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные), делать выводы в процессе работы и по её окончании, корректировать намеченный план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи.

Личностные: эстетическое отношение к языкам программирования, осознание их выразительных возможностей, готовность и способность обучающихся к саморазвитию и личностному самоопределению; сформированность их мотивации к обучению и целенаправленной познавательной деятельности.

Оборудование и материалы: компьютер, презентационное оборудование.

Распределение лабораторных работ:

Занятия 1, 2 — выполнение лабораторной работы 14

Дидактические материалы

Операционная система Андроид

Операционная система Андроид начала разрабатываться компанией Google с 2005 года. В 2007 году была представлена первая версия пакета разработчика и первый эмулятор Андроид. Сейчас последняя версия Андроид — Android X. Операционная система Андроид основана на ядре Linux и собственной реализацией виртуальной машины Java (Андроид не использует JVM по причине ограниченности ресурсов мобильных устройств для поддержки исходной JVM). Основными языками разработки Андроид-приложений являются Java и Kotlin. Язык Kotlin начал набирать популярность последние 2—3 года и, по сути, является ответом компании Google на претензии компании Oracle, связанные с использованием языка Java в Android.

Андроид обладает достаточно сложной структурой набора библиотек разработчика. С каждой версией Андроид появляются все новые и более изощренные компоненты и способы взаимодействия с ними. Кроме того, Андроид используется не только на мобильных устройствах типа смартфонов и планшетов, но и на множестве других «гаджетов»: очки, часы, элементы умного дома, нетбуки, телевизоры и т.д. Андроид активно развивается в духе концепции IoT (интернет-вещей).

В связи с достаточно высоким порогом вхождения в Андроид-программирование встал вопрос об упрощении разработки. Одним из решений было создание фреймворков, которые позволяли бы не переключаться на язык Java разработчикам .Net-приложе-

ний, например фреймворк Xamarin. В данное время большую популярность набирает разрабатываемый компанией Google фреймворк Flutter на базе языка Dart, который позволяет осуществлять быстрое прототипирование Андроид-приложений с высокой графической производительностью.

App Inventor

Одним из инструментов разработки мобильных приложений на базе платформы Андроид является среда визуальной разработки АИ, разрабатываемый и поддерживаемый MIT. АИ предоставляет возможность упрощённой разработки и идеально подходит для применения в сфере образования и обучения детей школьного возраста созданию мобильных приложений и основам программирования. Кроме того, за счёт простоты и возможности создания полноценных Андроид-приложений АИ подходит для быстрого создания и прототипирования приложений различного уровня сложности.

Редактор АИ представляет собой облачную среду визуальной разработки. Особенностью АИ является то, что среда, во-первых, не требует никакого специализированного ПО для разработки, а во-вторых, практически не требует навыков программирования для того, чтобы пользователь мог начать делать первые мобильные приложения на базе платформы Android.

Рис. 2. Логотип среды App Inventor

Для программирования в АИ используется графический интерфейс и Scratch-подобный визуальный язык программирования. Поэтому особый интерес АИ может представлять для учащихся, изучающих или изучивших Scratch.

АИ изначально разработан компанией Google в рамках подразделения Google Labs. Продукт разрабатывался на базе языка программирования Java и библиотеки Open Blocks. Библиотека Open Blocks разработана в MIT и представляет собой бесплатную библиотеку для разработки среды блочного программирования и также написана на языке Java. После решения о закрытии Google Labs компания Google прекратила поддержку АИ, при этом, сделав данное приложение открытым и передав его в MIT, который объявил об открытии нового центра мобильного обучения на базе АИ. Что примечательно, одним из сотрудников этого центра стал профессор Митчелл Резник, создатель языка Scratch. Первая публичная версия продукта была представлена в 2011 году. Официальный сайт продукта: http://appinventor.mit.edu

Компилятор, генерирующий байт-код Андроид на основе блоков, построенных в АИ, реализован на языке Kawa [1], представляющем собой диалект языка Scheme. Kawa является функциональным языком общего назначения и функционирует на базе Java-платформы (JVM). Среди всего прочего предоставляет возможность выполнения скриптов Scheme напрямую на Java.

Регистрация аккаунта Google

Для возможности работы в среде АИ необходим браузер (желательно Chrome) и аккаунт Google. Для создания аккаунта необходимо зайти на сайт Google https://www.google.ru

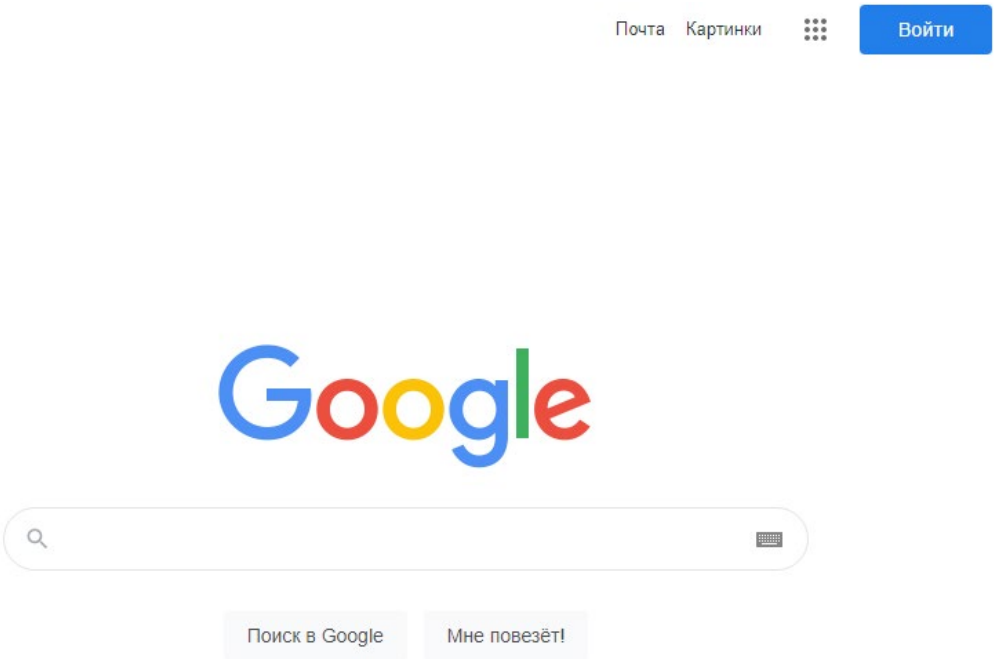


Рисунок 3. Сайт Google

Нажать кнопку «Войти» в правом верхнем углу, после чего система переходит в меню выбора или создания аккаунта:

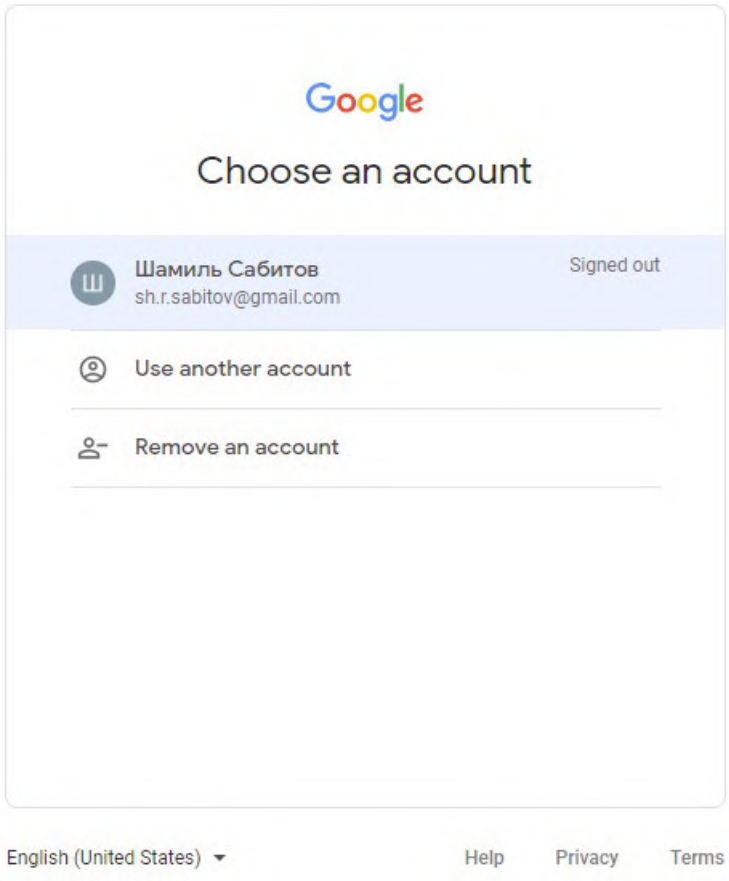


Рисунок 4. Выбор аккаунта Google

Если у пользователя уже есть аккаунт, то сайт предоставит возможность его выбора, либо сайт перекинет пользователя на окно входа в аккаунт:

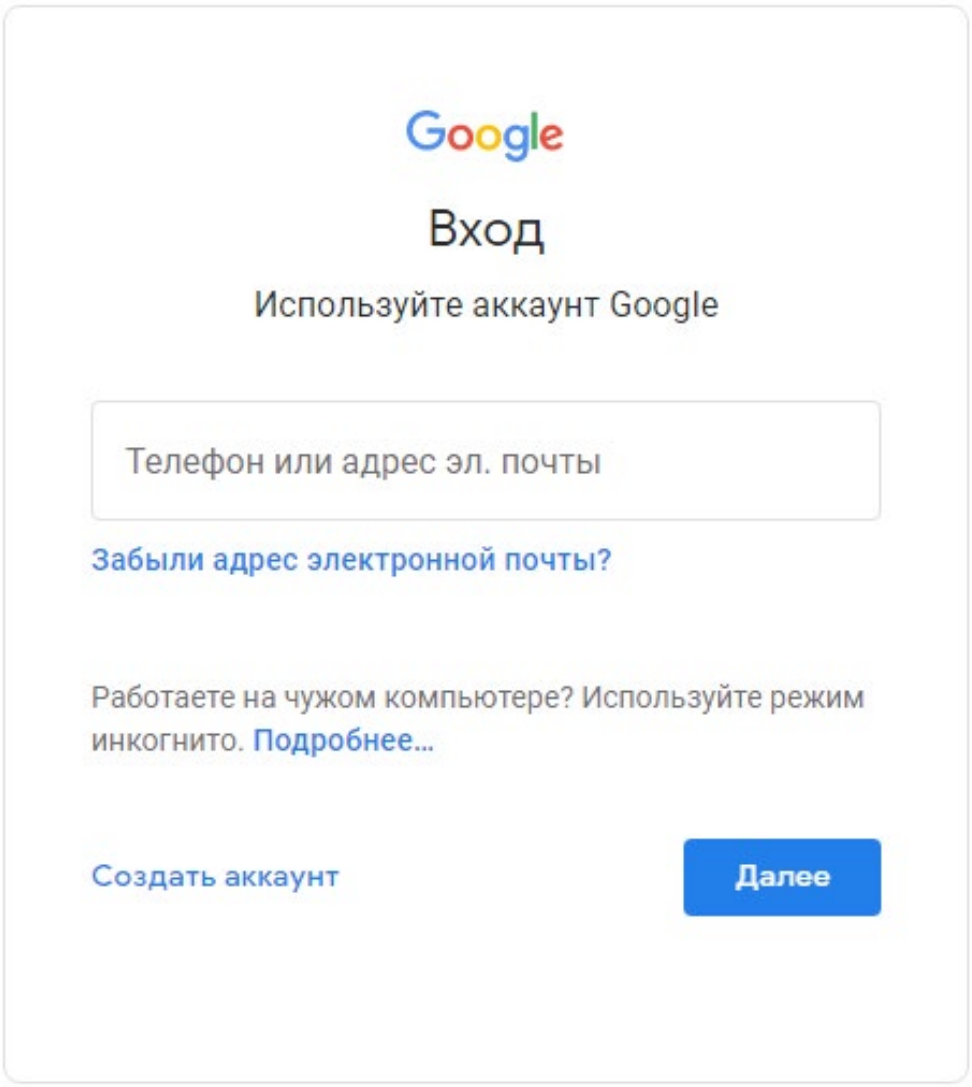


Рисунок 5. Вход в аккаунт Google

В меню входа в аккаунт вводится аккаунт Google (при наличии). В другом случае необходимо его создать, нажав кнопку «Создать аккаунт», расположенную в левом нижнем углу.

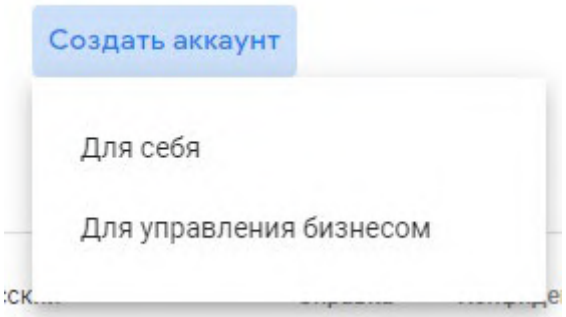


Рисунок 6. Кнопка «Создать аккаунт»

В открывшемся контекстном меню следует выбрать пункт «Для себя», после чего появляется окно заполнения личных данных, где пользователю предлагается создать новый аккаунт Google:

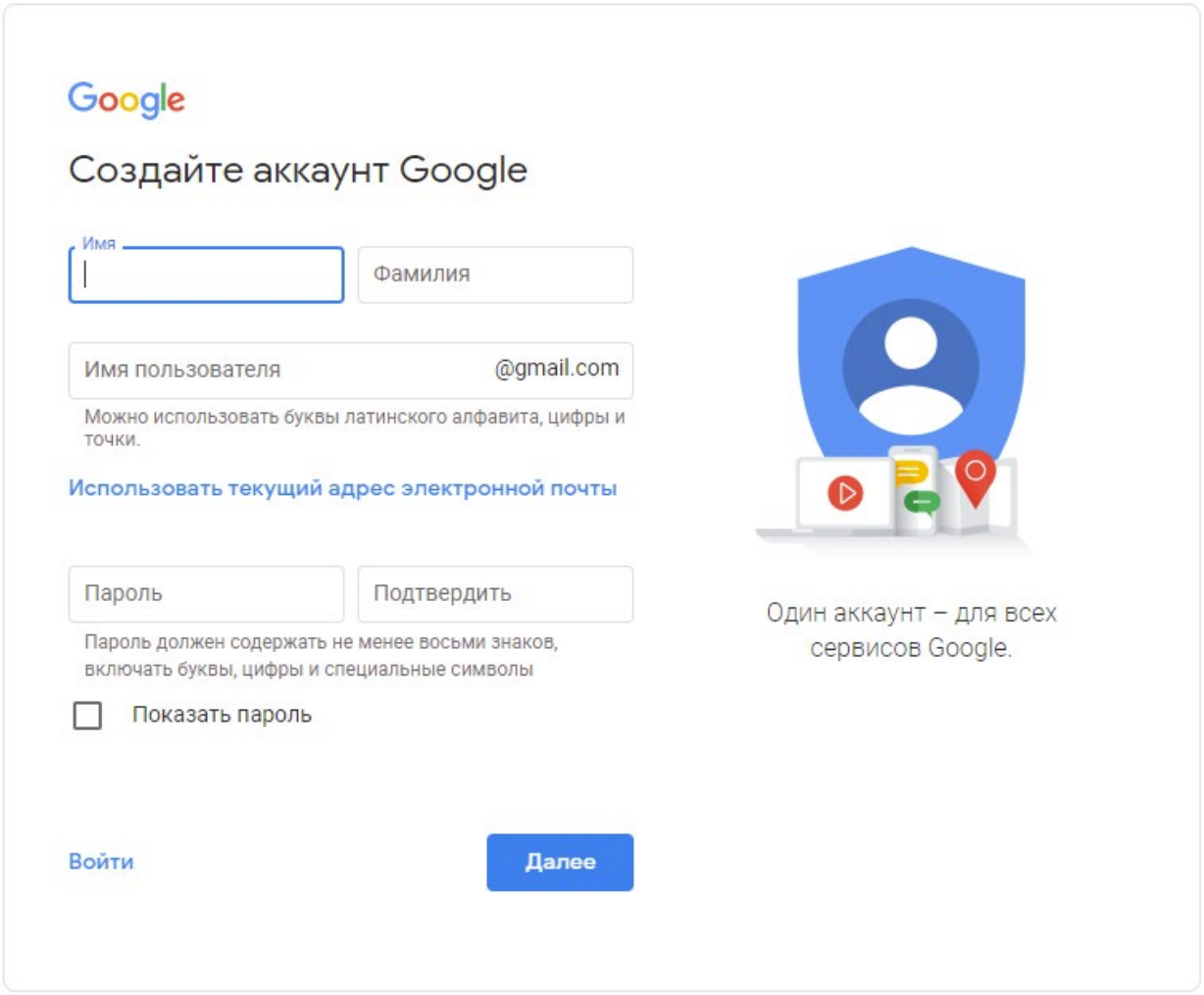


Рисунок 7. Заполнение личных данных

После заполнения личных данных необходимо нажать «Далее», после чего состоится переход на следующее окно с продолжением заполнения личных данных:

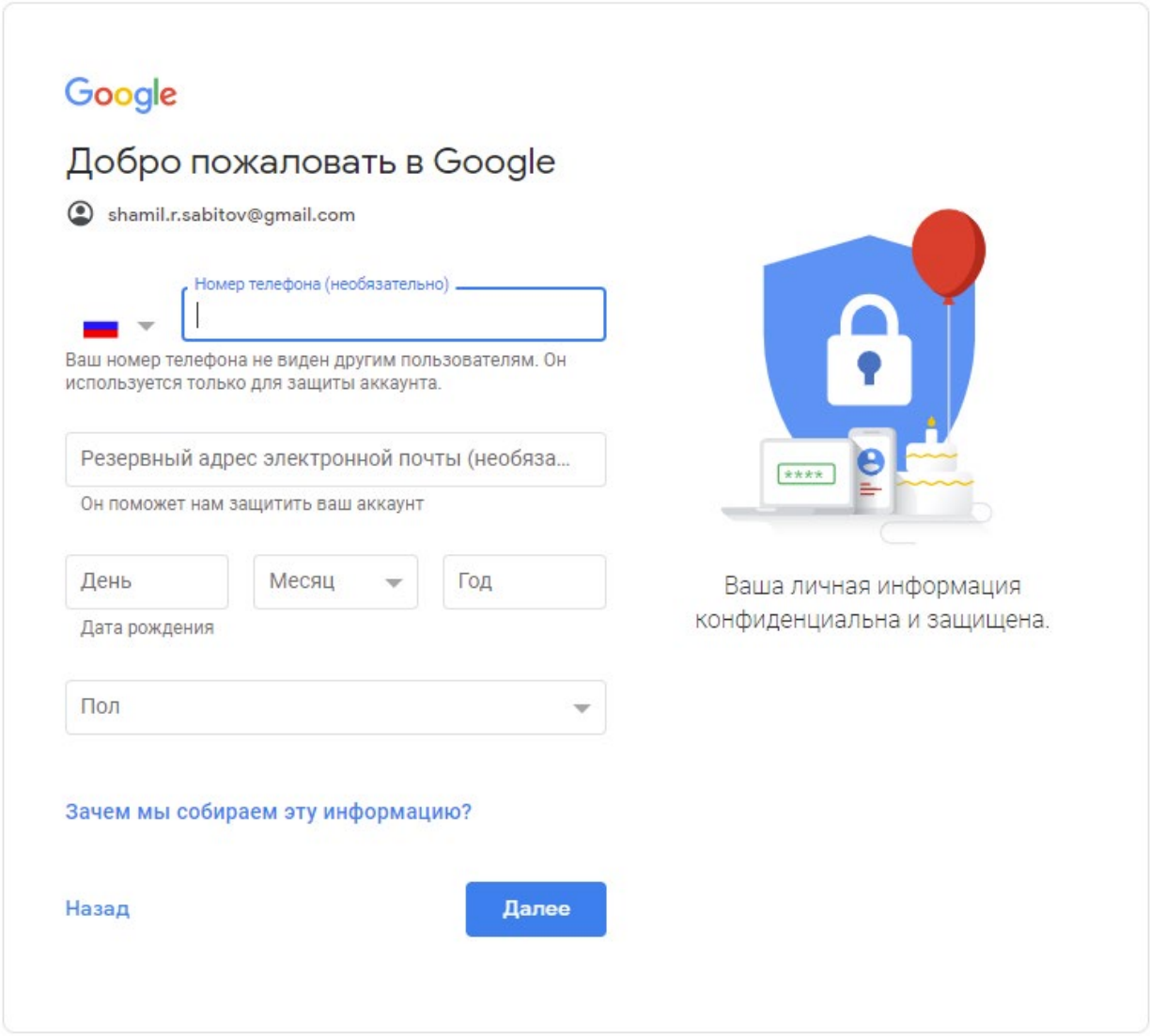


Рисунок 8. Второе окно заполнения личных данных

После заполнения данных также следует нажать «Далее» в правом нижнем углу, после чего в окне подтверждения номера необходимо подтвердить указанный номер телефона.

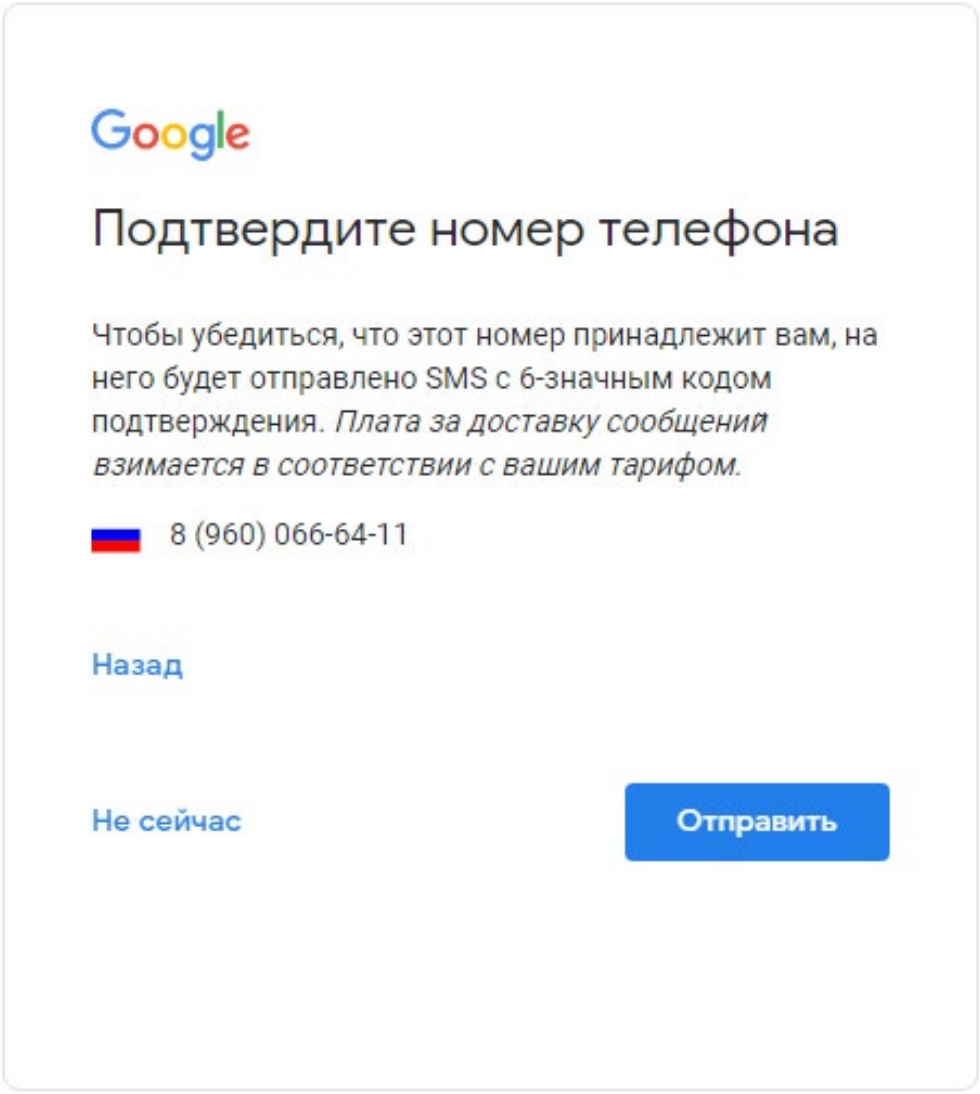


Рисунок 9. Подтверждение номера

Для начала операции подтверждения необходимо нажать кнопку «Отправить» и дождаться СМС-сообщения с проверочным кодом:

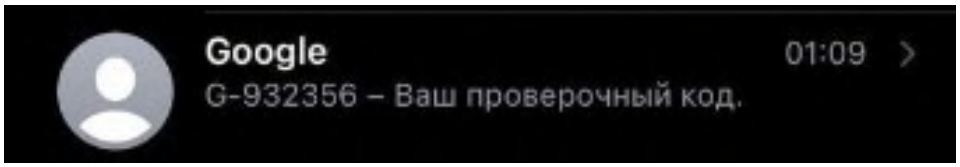


Рисунок 10. Проверочный код

После этого необходимо корректно ввести цифры после буквы «G-» в появившемся поле ввода кода и тем самым подтвердить номер телефона аккаунта Google.

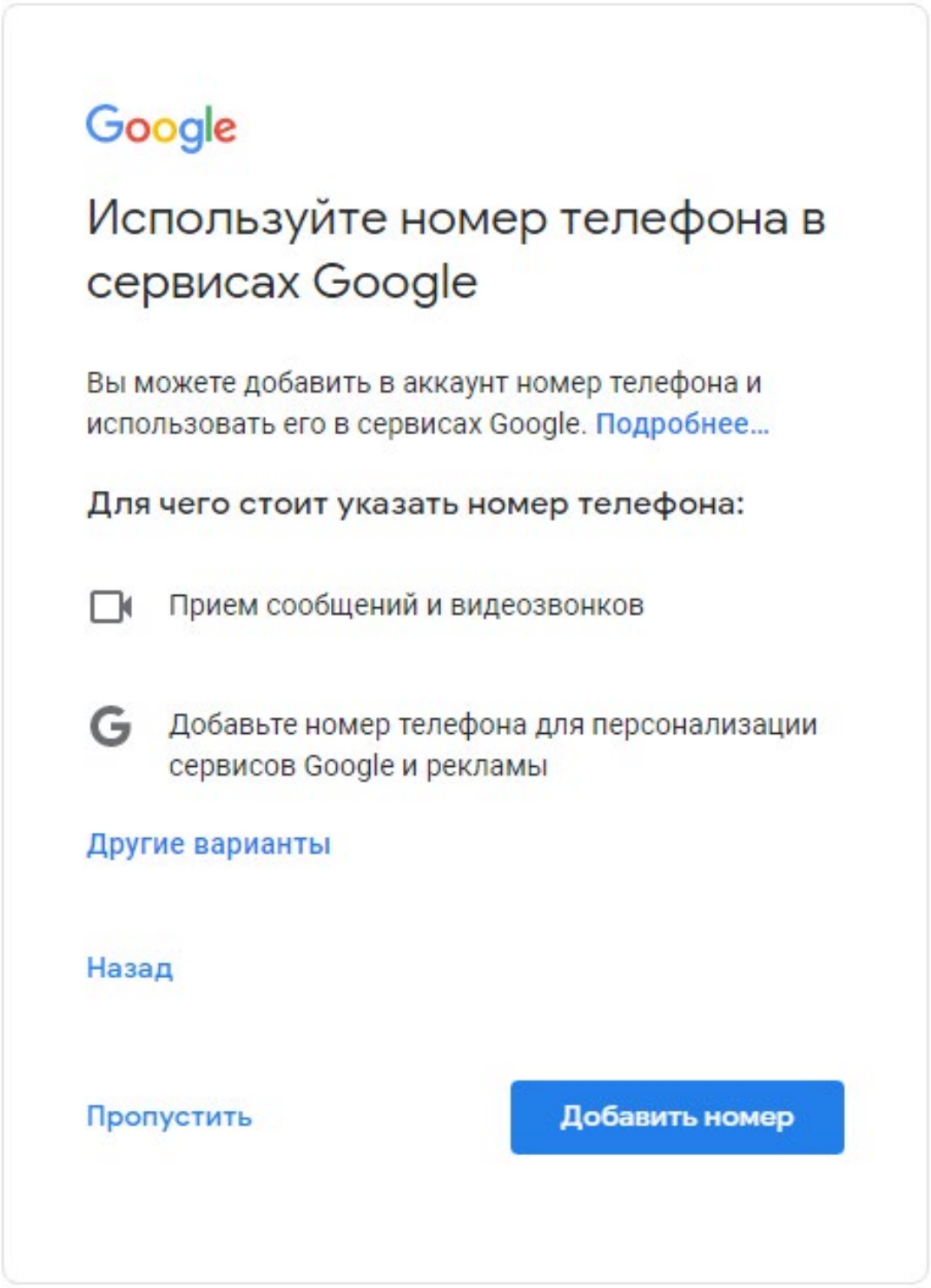


Рисунок 11. Подтверждение и добавление номера

В окне добавления номера для использования в сервисах Google выбрать «Пропустить». Далее в окне подтверждения договора необходимо пролистать текст договора до конца и нажать кнопку «Принимаю».

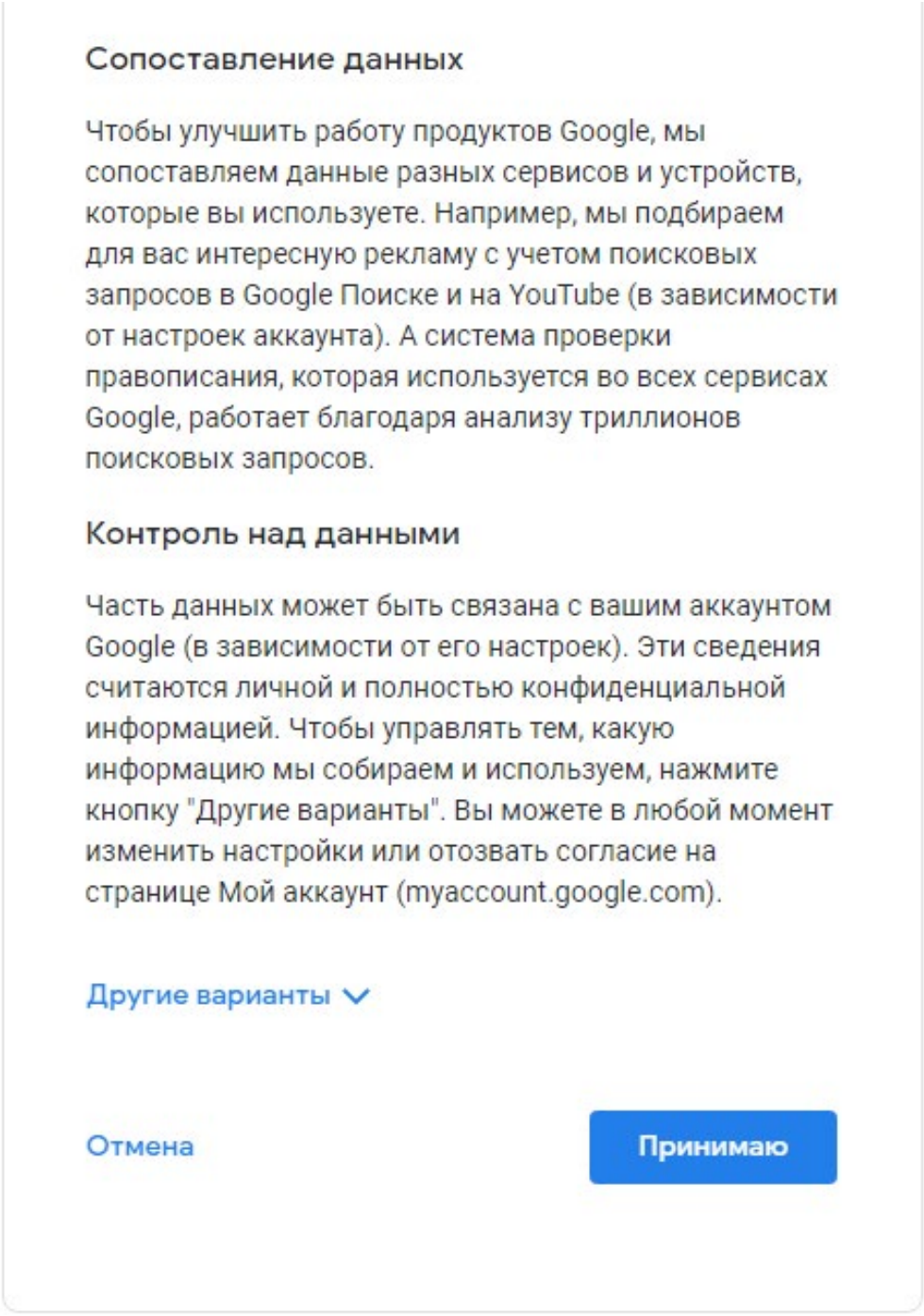


Рисунок 12. Подтверждение договора

Регистрация завершена. Можно приступать к использованию аккаунта Google.

Регистрация на сайте appinventor.mit.edu

Следующим шагом после создания аккаунта Google является переход на сайт АИ http://appinventor.mit.edu и навигация по нему.

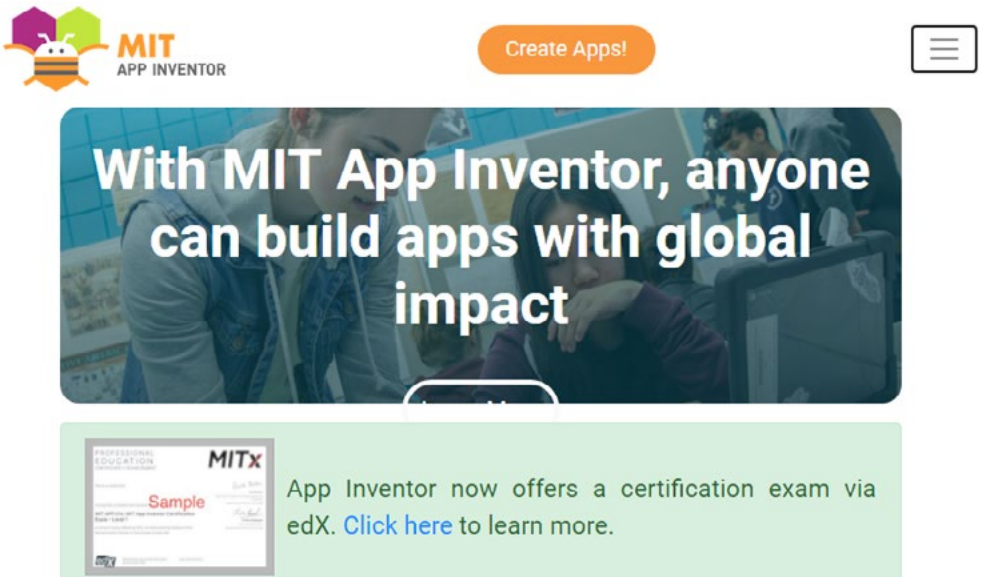


Рисунок 13.

После входа на сайт App Inventor необходимо нужно нажать на «Create Apps!2. Кнопка находится справа от логотипа App Inventor в левом верхнем углу сайта. После этого сайт перекидывает пользователя в окно входа и авторизации аккаунта Google:

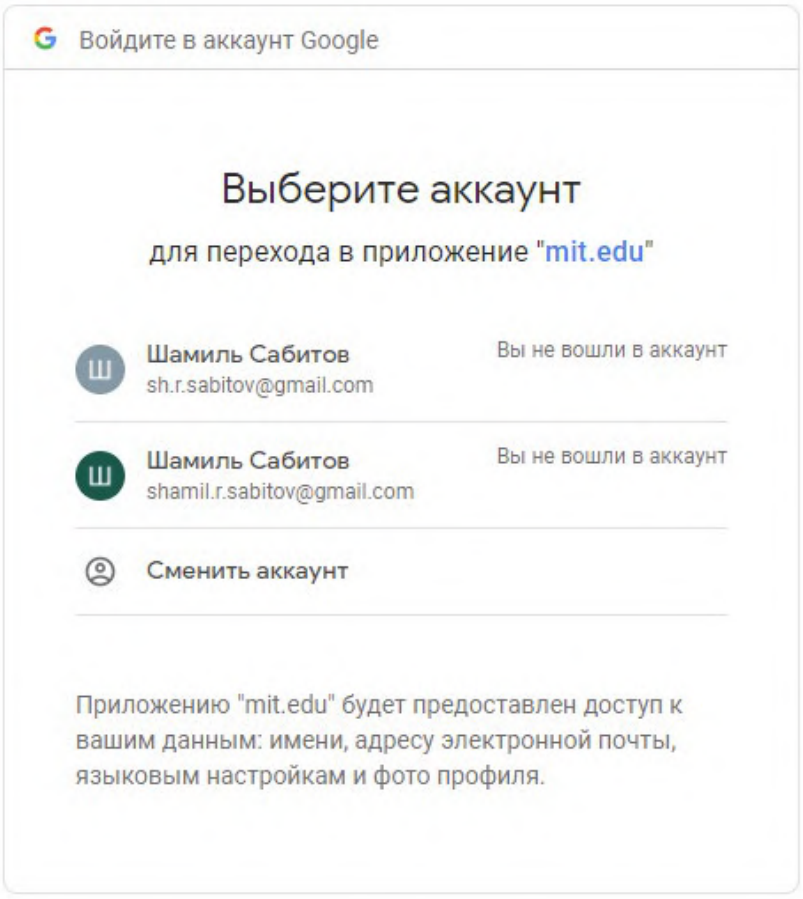


Рисунок 14. Вход в аккаунт Google

В данном окне необходимо выбрать созданный ранее аккаунт. Если он отсутствует, то выбрать кнопку «Сменить аккаунт» и перейти в окно авторизации для ввода логина аккаунта Google:

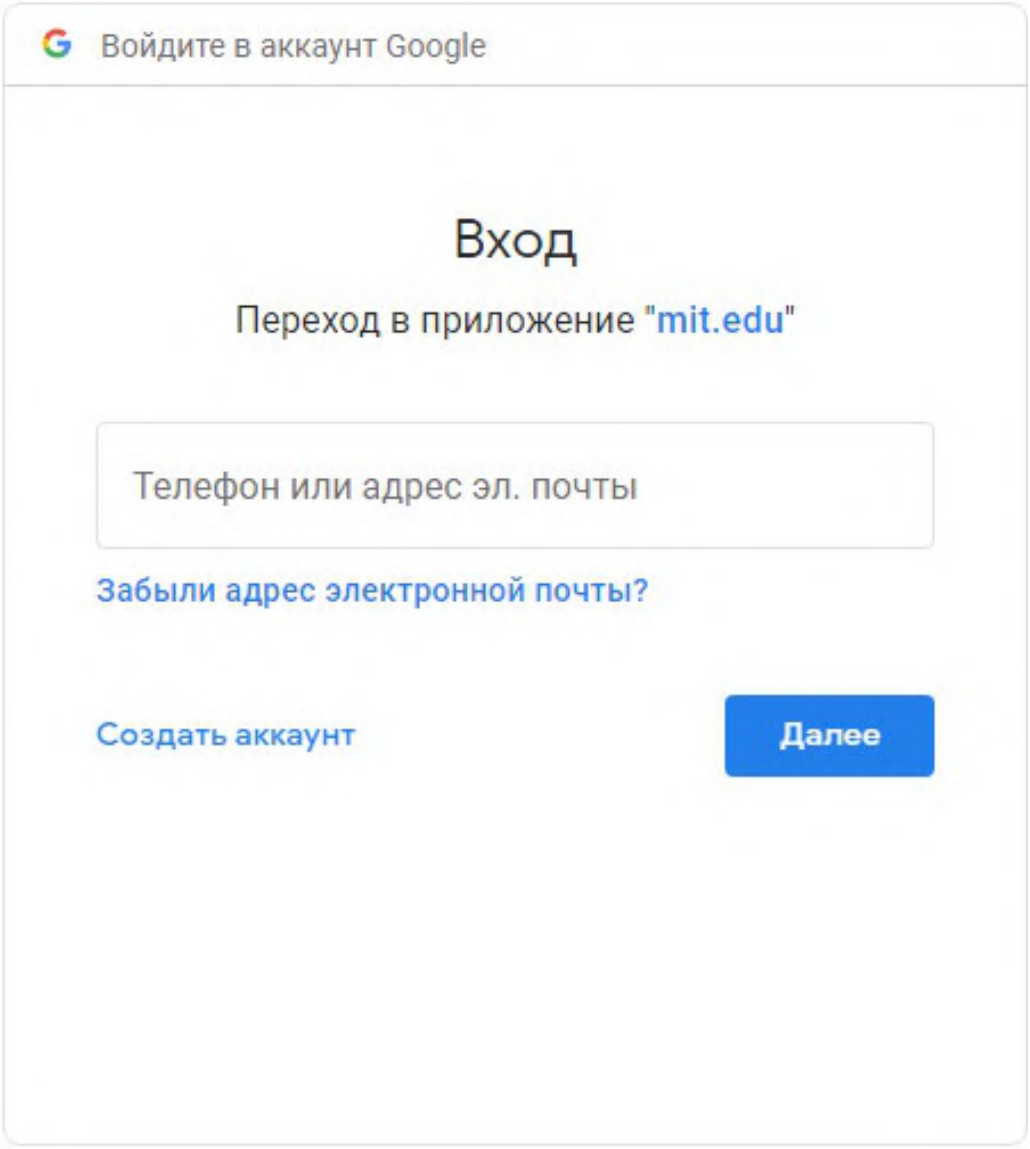


Рисунок 15. Ввод логина

В окне ввода логина требуется заполнить телефон или адрес электронной почты (адрес, указанный ранее как «Имя пользователя»), нажать кнопку «Далее», после чего откроется окно ввода пароля:

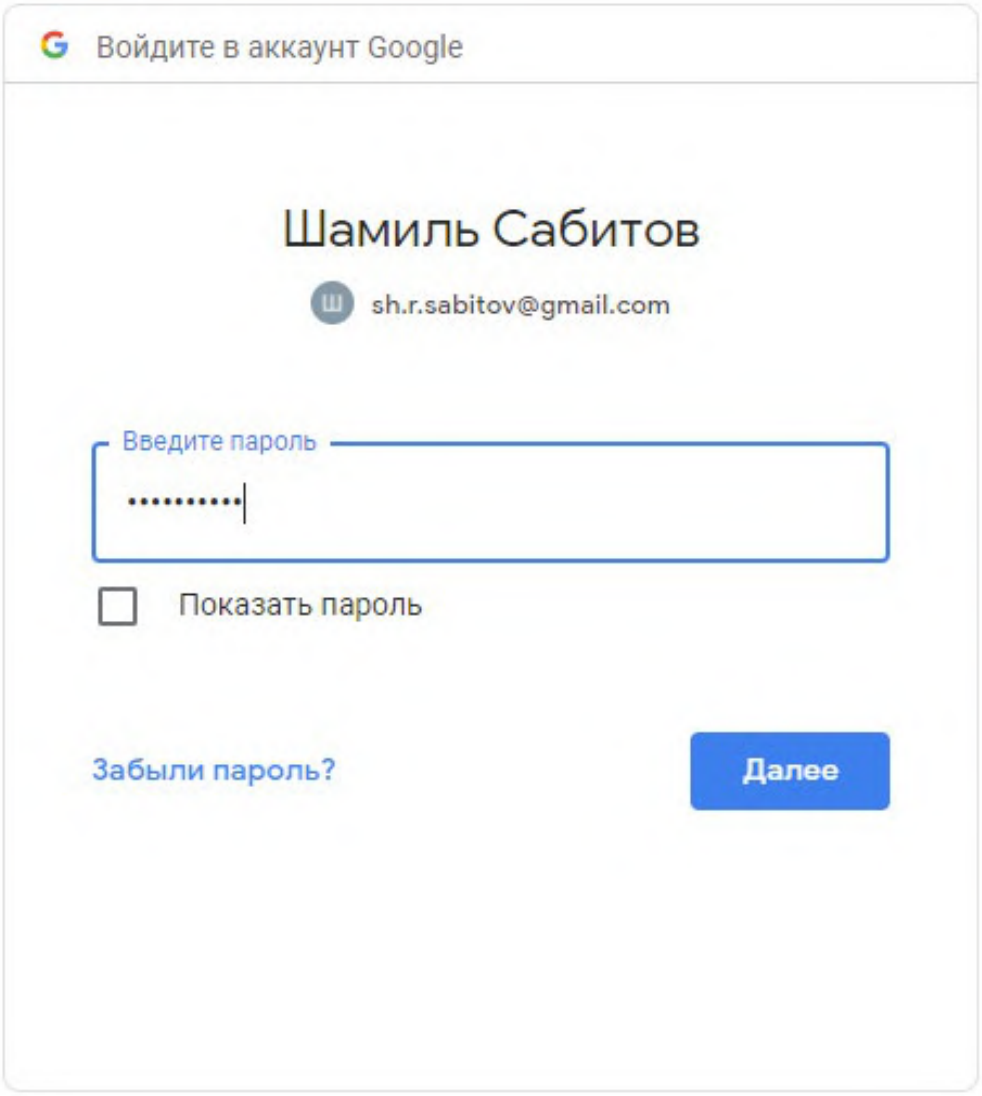


Рисунок 16. Окно ввода пароля

После ввода пароля следует нажать кнопку «Далее». При успешной авторизации система перейдёт на главную страницу сайта АИ, где необходимо нажать на кнопку «Create Apps!», после чего осуществится переход на окно подтверждения политики сайта, в котором необходимо нажать кнопку «I accept the terms of service!»:

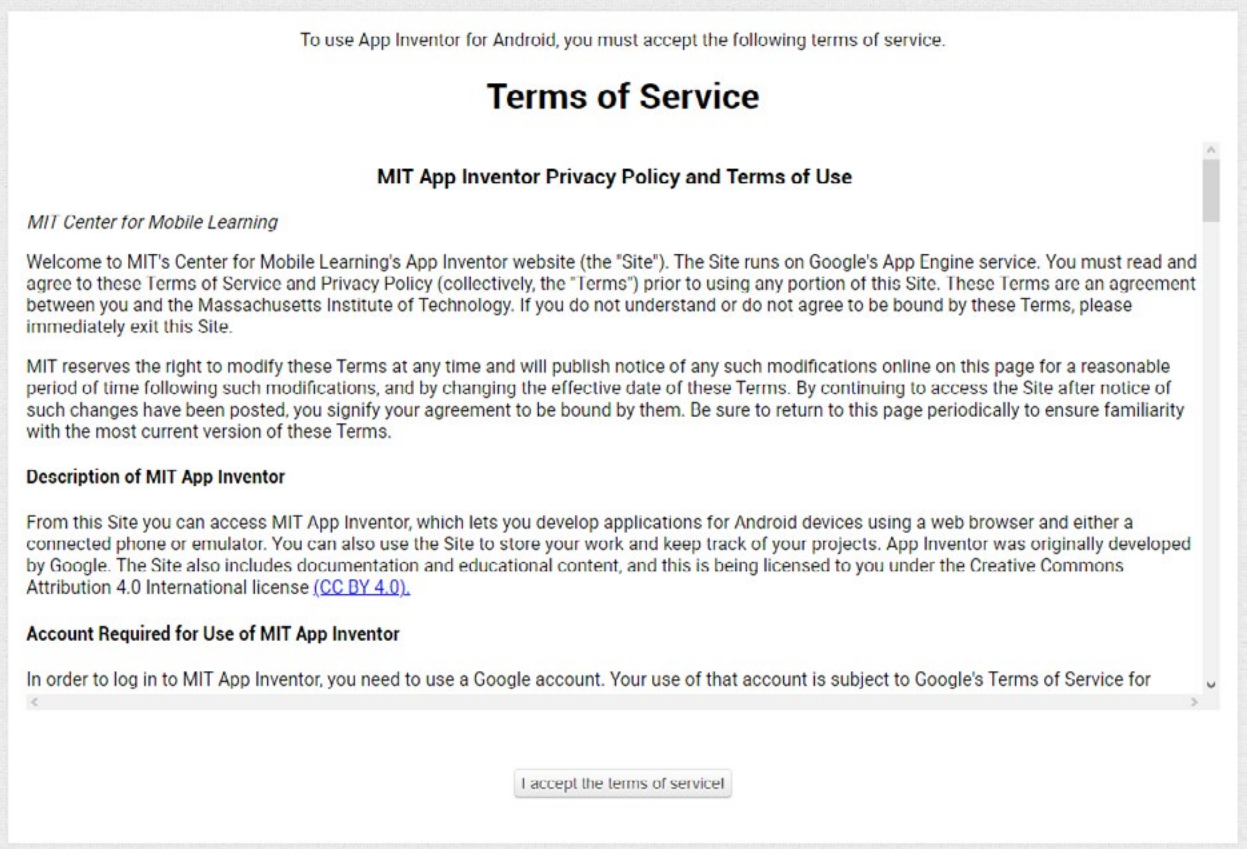


Рисунок 17. Политика подтверждения App Inventor.

Теперь можно наблюдать приветственное окно сайта «App Inventor»:

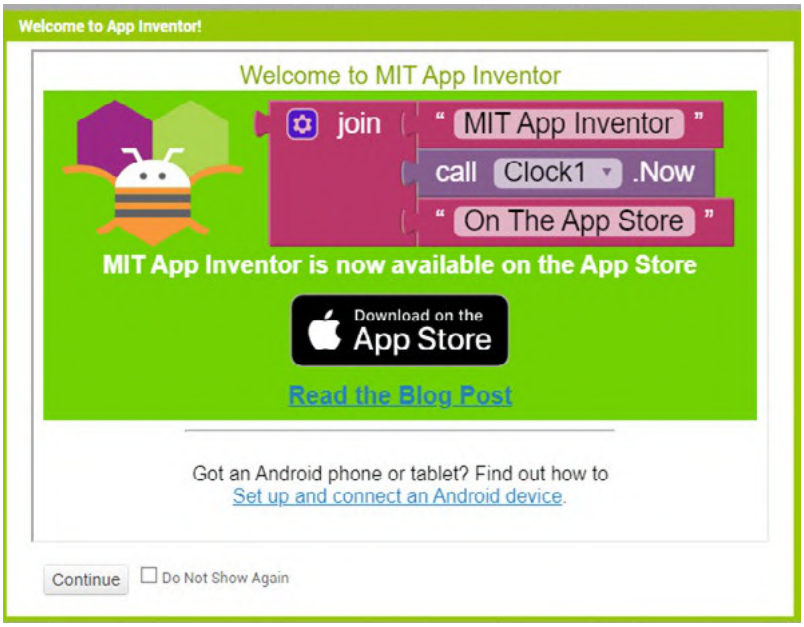


Рисунок 18. Приветственное окно сайта App Inventor

После нажатия кнопки «Continue» появится окно среды АИ на английском языке. Чтобы перейти на русский язык, требуется нажать на кнопку выбора языка (при первом входе это будет «English»), после чего в списке языков найти «Русский» и кликнуть на него ЛКМ.

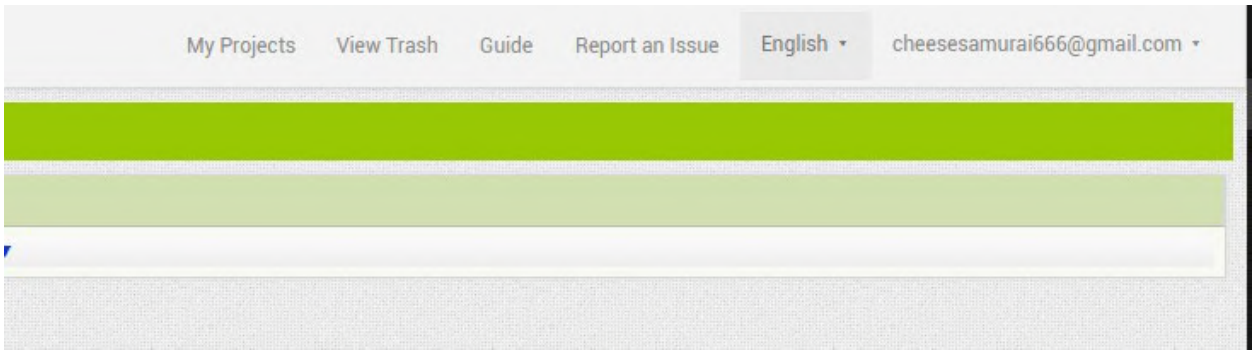


Рисунок 19. Выбор языка

Создание первого проекта

Кроме всего прочего, АИ привлекателен ещё тем, что для создания проектов не надо устанавливать никаких интегрированных сред разработки и SDK (комплектов библиотек для разработчика), каковые пришлось бы устанавливать и настраивать в случае использования, например, Android Studio. Для работы с АИ достаточно лишь операционной системы Win 10 и последней версии браузера (желательно Chrome).

При необходимости можно использовать дополнительное ПО: эмулятор. Эмулятор прост в установке и нужен лишь для работы в режиме отладки.

После выбора языка можно создавать проекты в среде АИ. Для этого нужно нажать кнопку «Проекты» в меню в левом верхнем углу, после чего выбрать в появившемся окне пункт «Начать новый проект».

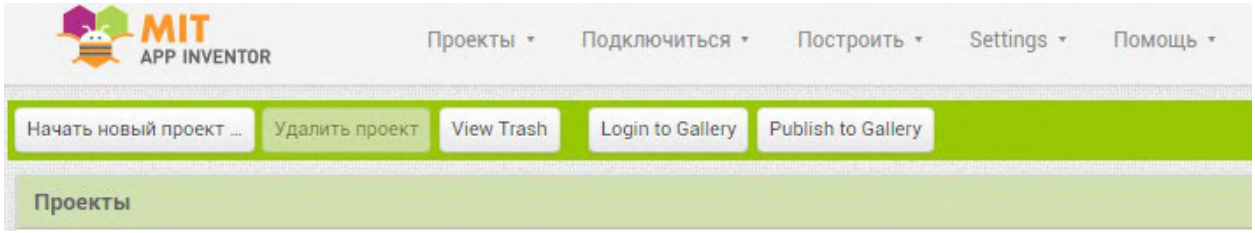


Рисунок 20. Проекты

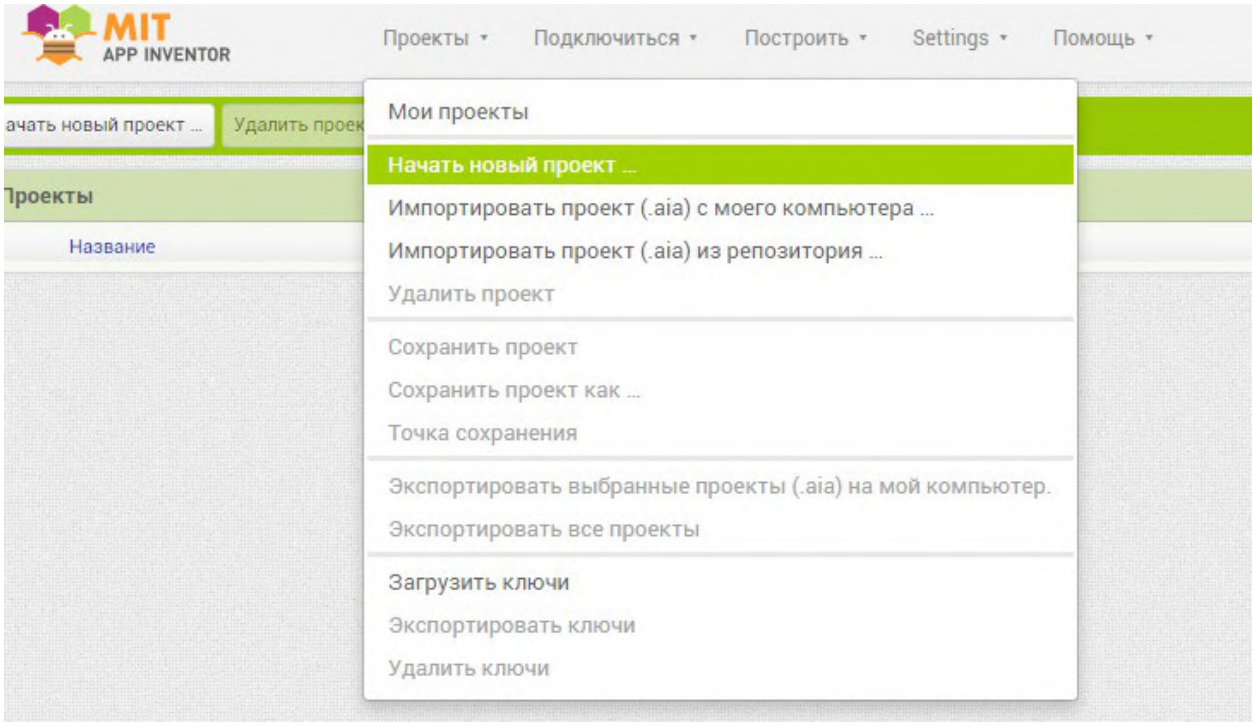


Рисунок 21. Начать новый проект

В новом окне нужно указать название проекта и нажать кнопку ОК:

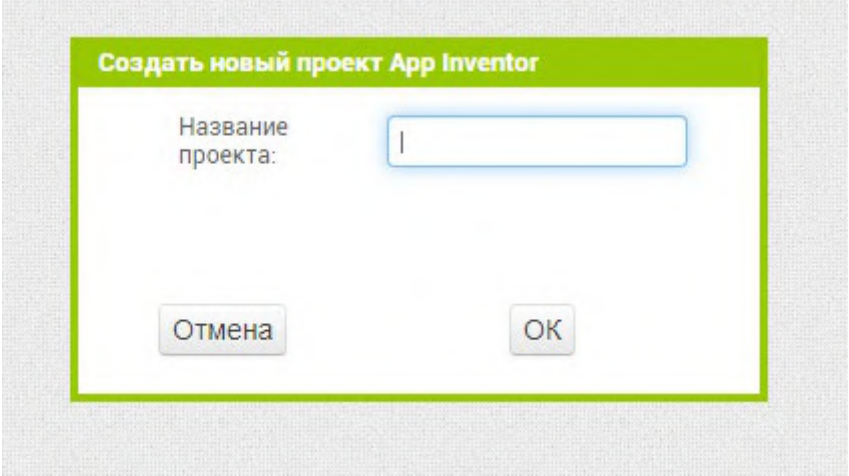


Рисунок 22. Название проекта

Далее состоится автоматический переход в рабочее пространство среды АИ.

Главное меню среды

В рабочем пространстве среды наверху представлено главное меню, состоящее из следующих пунктов (в виде кнопок):

Проекты, Подключиться, Построить, Settings, Помощь, Мои проекты, View Trash, Руководство, Сообщить о проблеме, Русский и **ссылка** на аккаунт пользователя.

Кнопка Проекты раскрывает следующие пункты меню:

Пункт меню «Мои проекты», где хранятся все проекты пользователя. При выборе этого пункта открывается окно «Проекты», где указаны все проекты пользователя с названием, датой создания и датой последнего редактирования.

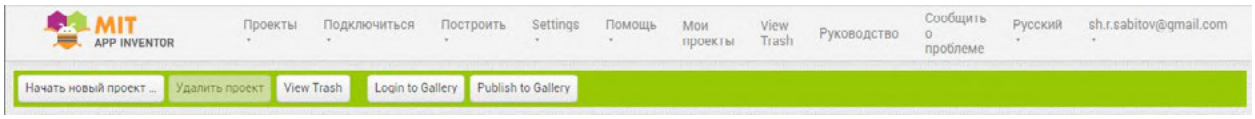


Рисунок 23. Проекты

Пункт меню «Новый проект» предназначен для создания нового проекта.

Пункт меню «Импортировать проект (.avi) с моего компьютера» предназначен для импорта шаблонов (только дизайн) АИ-проектов с компьютера пользователя.

Пункт меню «Импортировать проект (.avi) из репозитория» предназначен для импорта шаблонов готовых проектов (только дизайн) из репозитория АИ:

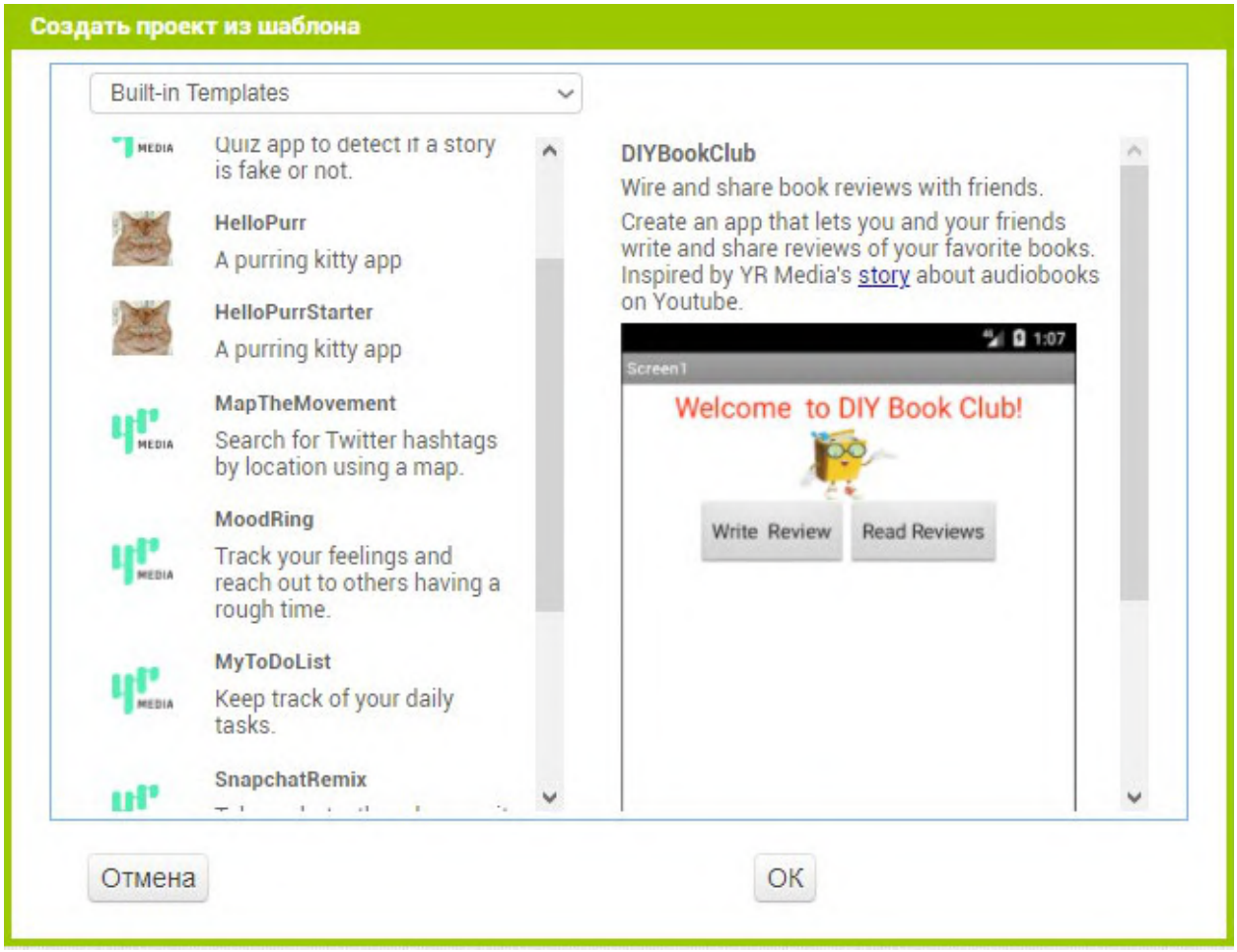


Рисунок 24. Импорт шаблонов проектов

Пункт меню «Удалить проект» предназначен для удаления выбранного проекта. Выбрать проект можно, пометив его галочкой (т.е. нажав на квадратик):

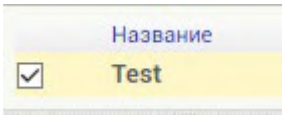


Рисунок 25. Выбор проекта

Пункт меню «Сохранить проект» предназначен для сохранения текущего проекта, а «Сохранить проект как...» даёт возможность сохранить текущий проект как новый проект (т.е. добавить его к уже имеющимся, создать копию):

Рисунок 26. Добавление проекта к уже имеющимся

Пункты меню «Загрузить ключи», «Удалить ключи» и «Экспортировать ключи» нужны в случае наличия у пользователя ключей от Google Play. Позволяют загружать созданные приложения в Google Play.

Кнопка «Подключиться» предназначена для отладки, т.е. для запуска приложения без его установки на устройство:

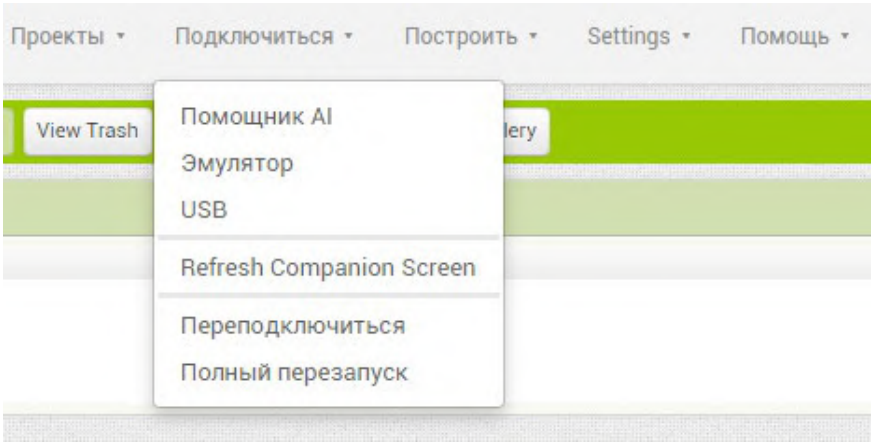


Рисунок 27. Меню «Подключиться»

Основными пунктами меню кнопки «Подключиться» являются пункты «Эмулятор» и «USB». Чтобы начать разработку приложений на базе АИ, необходимо устройство с операционной системой Android или же эмулятор устройства.

Пункт меню «Эмулятор» запускает виртуальный эмулятор мобильного устройства, работающего на базе платформы Андроид.

Пункт меню «USB» подключает мобильное устройство к ПК и запускает проекты АИ (т. е. созданные пользователем в АИ приложения Андроид) на физически существующем устройстве. Тем самым предоставляется возможность проверять работоспособность на реальных мобильных устройствах.

Пункты «Переподключиться» и «Полный перезапуск» отвечают за переподключение к мобильному устройству и перезапуск эмулятора.

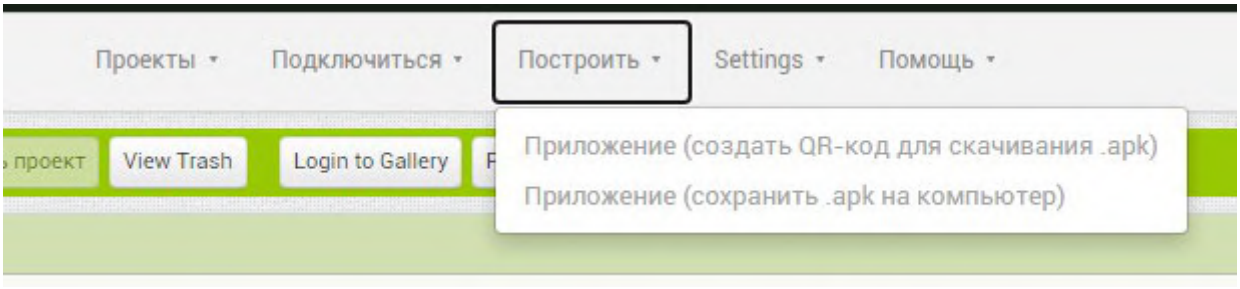


Рисунок 28. «Построить»

Функционал кнопки «Построить» состоит из двух пунктов:

Пункт меню «Приложение (создать QR-код для сканирования .apk)» предназначен для скана и загрузки созданного приложения через QR-код (скомпилированное приложение временно хранится на серверах разработчиков АИ).

Пункт меню «Приложение (сохранить .apk на компьютер)» позволяет сохранить приложение на ПК пользователя в формате «APK». После этого пользователь может самостоятельно загрузить его на устройство и установить (предварительно в меню настроек должно быть установлено разрешение на установку apk-файлов из посторонних источников. Местоположение этого пункта меню зависит от версии Андроид и может быть найдено через поиск в меню настроек мобильного устройства).

Кнопка «Settings» (Настройки) раскрывает меню с пунктами Disable Project Autoload и Enable OpenDyslexic:

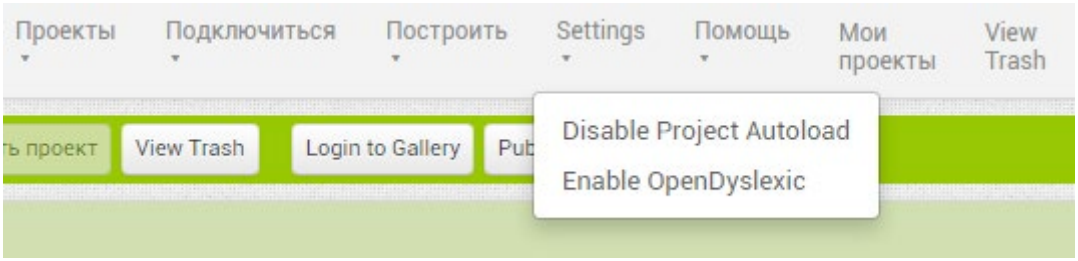


Рисунок 29. Функционал кнопки «Settings»

Функция «Disable Project Autoload» позволяет отключать автосохранение текущего проекта пользователя, что нежелательно.

Кнопка «Помощь» содержит типовой функционал меню **Помощь информационных систем** и предоставляет доступ к урокам, справке, форуму, общению со службой поддержки и т.д.

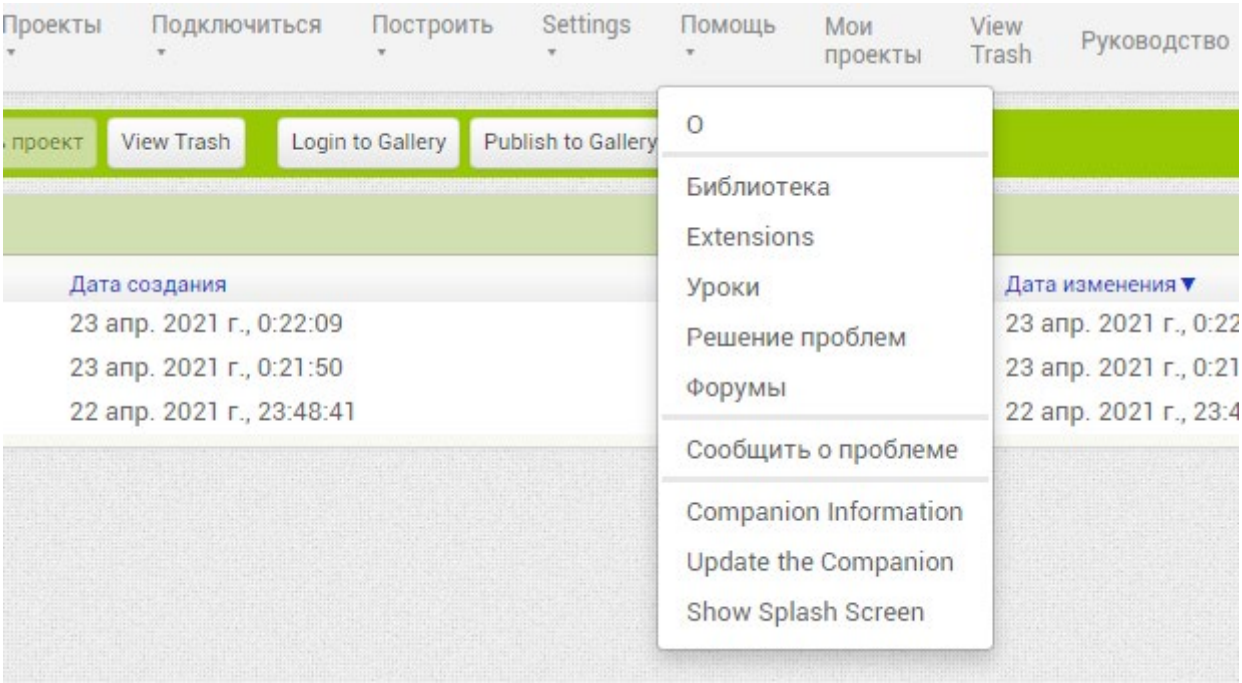


Рисунок 30. Функционал кнопки «Помощь»

Пункт меню «О» открывает окно о версиях совместимости будущего приложения с Android-системами, о текущей версии сборки.

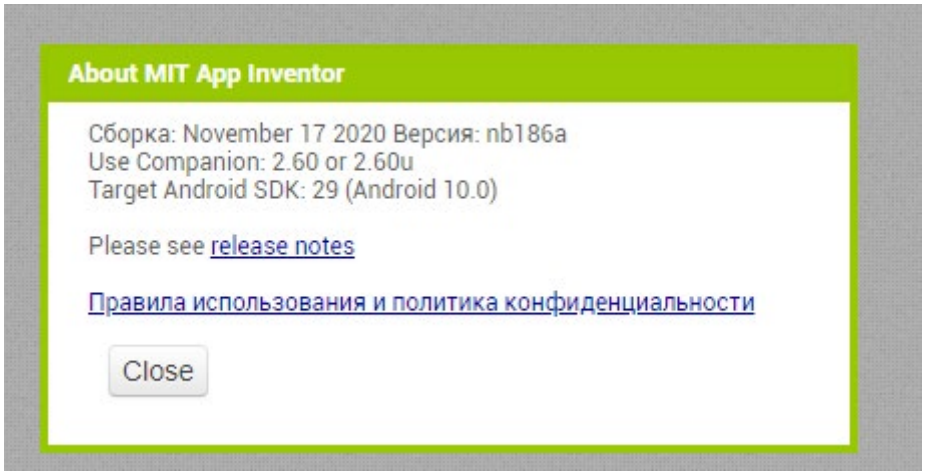


Рисунок 31. Функционал «О»

Функционал меню «Библиотека» содержит ссылки на обширную документацию (на английском языке).

Пункт меню «Extensions» является ссылкой на сайт, где расположены различные расширения, написанные энтузиастами для АИ. В данном курсе расширения АИ не рассматриваются.

Пункт меню «Уроки», является ссылкой на ресурс, где выставлены хорошие уроки по App Inventor от самих разработчиков (язык уроков — английский).

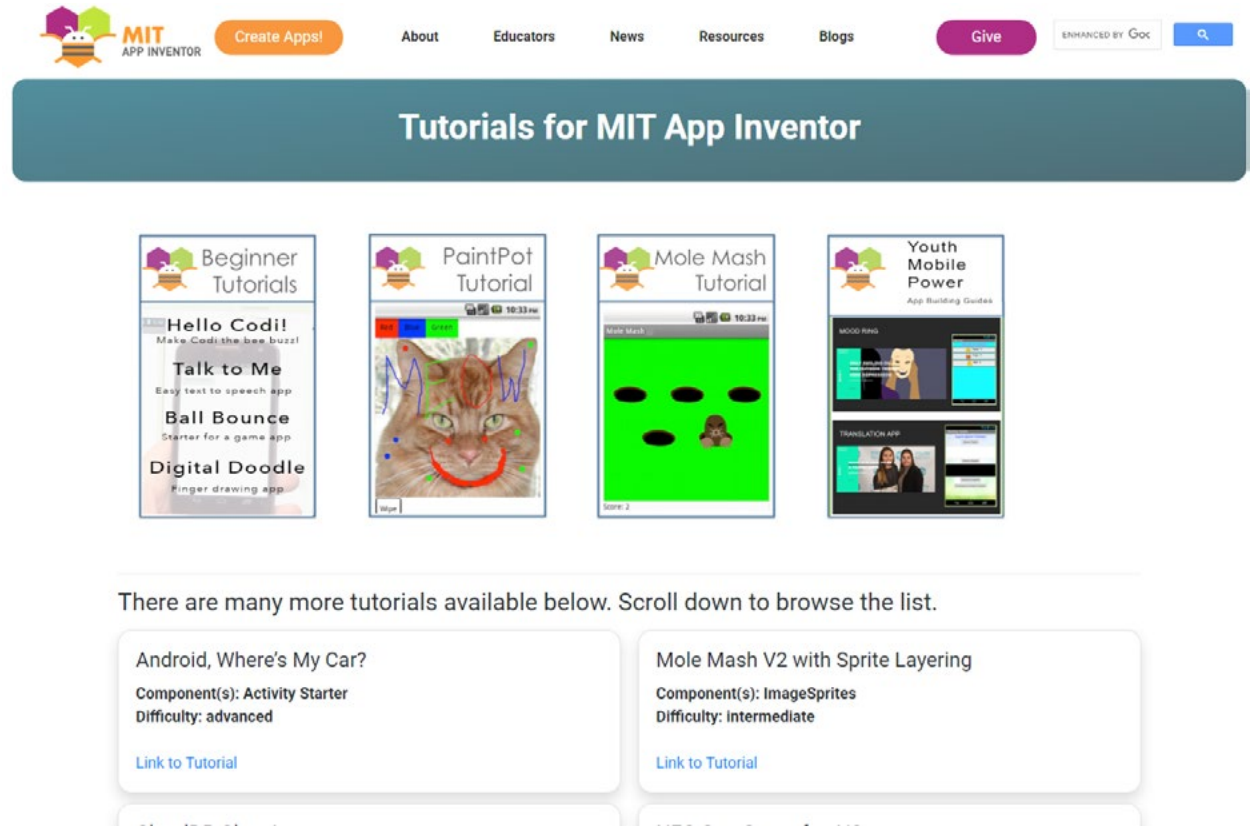


Рисунок 32. Функционал «Уроки»

Пункт меню «Решение проблемы» содержит ссылку на страницу, где находятся часто задаваемые вопросы (на английском языке).

Режимы работы среды

Имеются два основных режима работы среды разработки App Inventor: «Дизайнер», где разрабатывается внешний вид приложения (т.е. дизайн, элементы интерфейса пользователя (поля, кнопки и т.д.) и все косметические действия с приложением) и «Блоки» (где с помощью визуального конструктора разрабатывается функционал приложения). Переключение между режимами происходит путём нажатия ЛКМ на соответствующую кнопку:

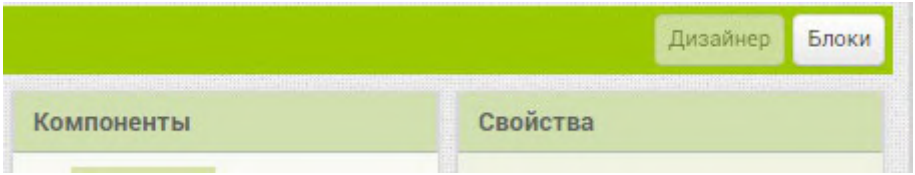


Рисунок 33. Переключение окон «Дизайнер» и «Блоки»

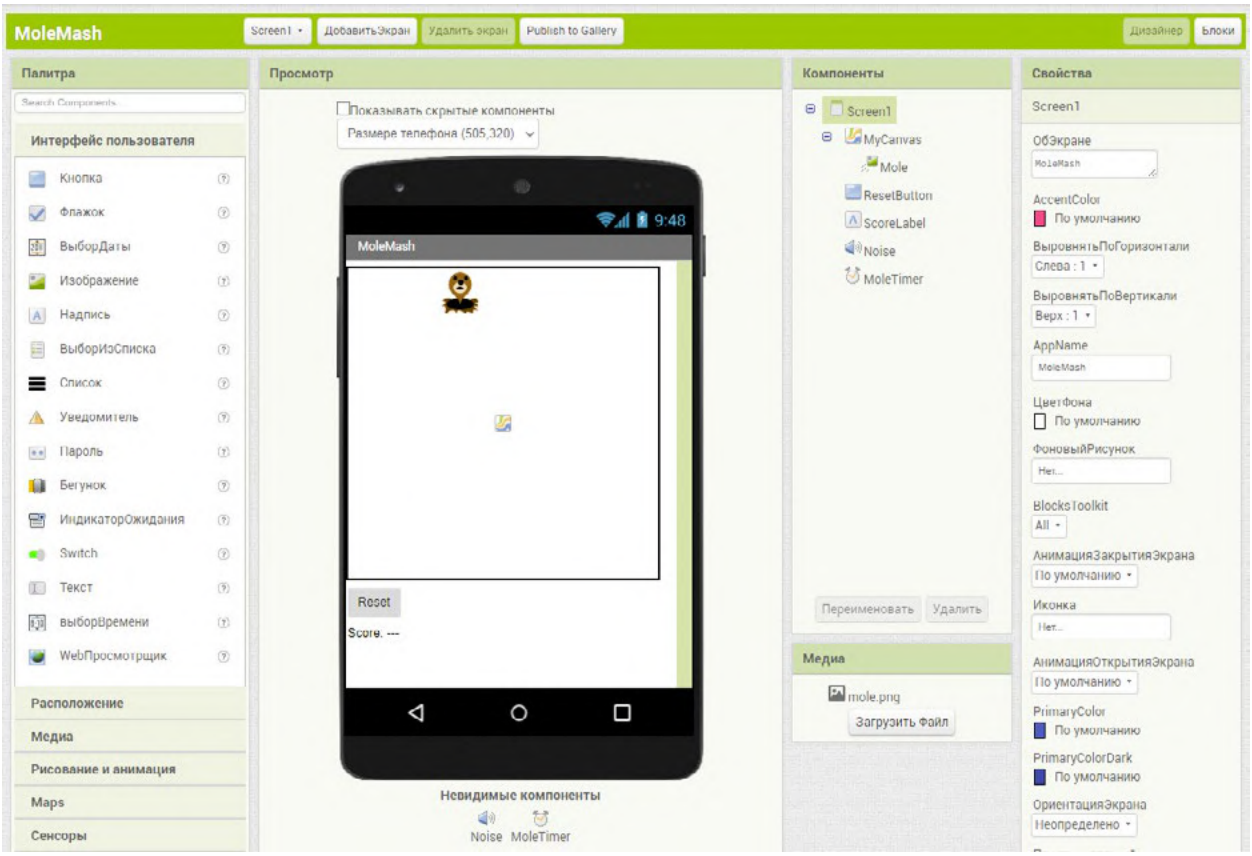


Рисунок 34. Окно режима «Дизайнер»

В режиме Дизайнер в меню сверху имеются кнопки для переключения между экранами и для создания экранов:

Рисунок 35. Выбор и создание экранов

Экран представляет собой то же, что и экран в реальном мобильном устройстве, — это компонент, посредством которого возможно наблюдать, как проявляется себя код и разработанный дизайн.

Режим Дизайнер представлен несколькими окнами (см. Рисунок 34 Окно режима «Дизайнер»):

- Окно палитры
- Окно просмотра
- Окно компонент
- Окно свойств
- Окно медиа

Окно палитры компонент содержит различные компоненты или объекты, которые пользователь может добавлять на экран. Сама палитра содержит несколько разделов. В каждом из разделов представлены компоненты определённого типа. Около каждой компоненты имеется знак вопроса. При нажатии на знак вопроса появляется всплывающее окно, в котором отображается информация по данному компоненту.

В первом разделе «Интерфейс пользователя» находятся кнопки, флажки, надписи, списки, поля и т.д., всё, что можно добавить для организации взаимодействия с пользователем; всё, посредством чего пользователь может передавать данные в систему.

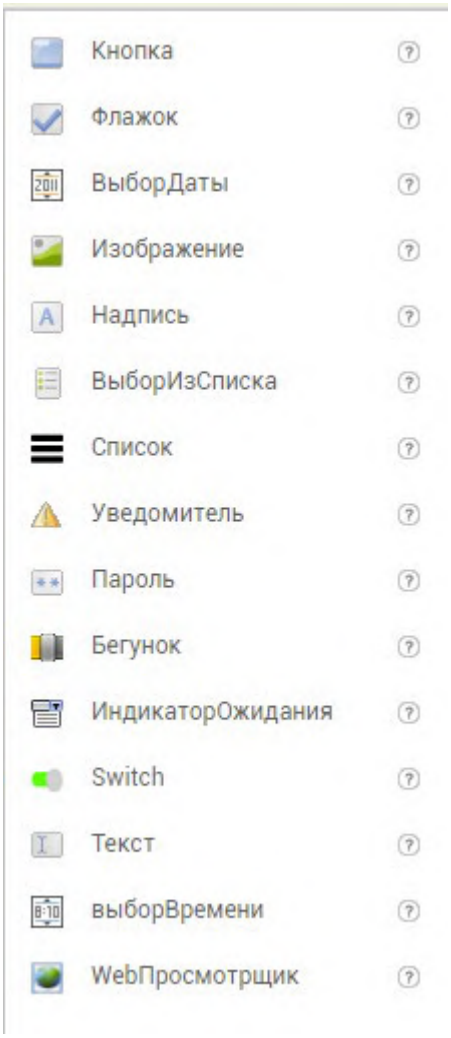


Рисунок 36. Раздел «Интерфейс пользователя»

Следующий раздел «Расположение» (специальные компоненты для группировки объектов по горизонтали или по вертикали). Используется для прокрутки экрана (скроллинга), например, когда на экране приложения имеется очень много объектов или элементов. В этом случае приходится прокручивать экран приложения, чтобы перейти от одной части экрана к другой.

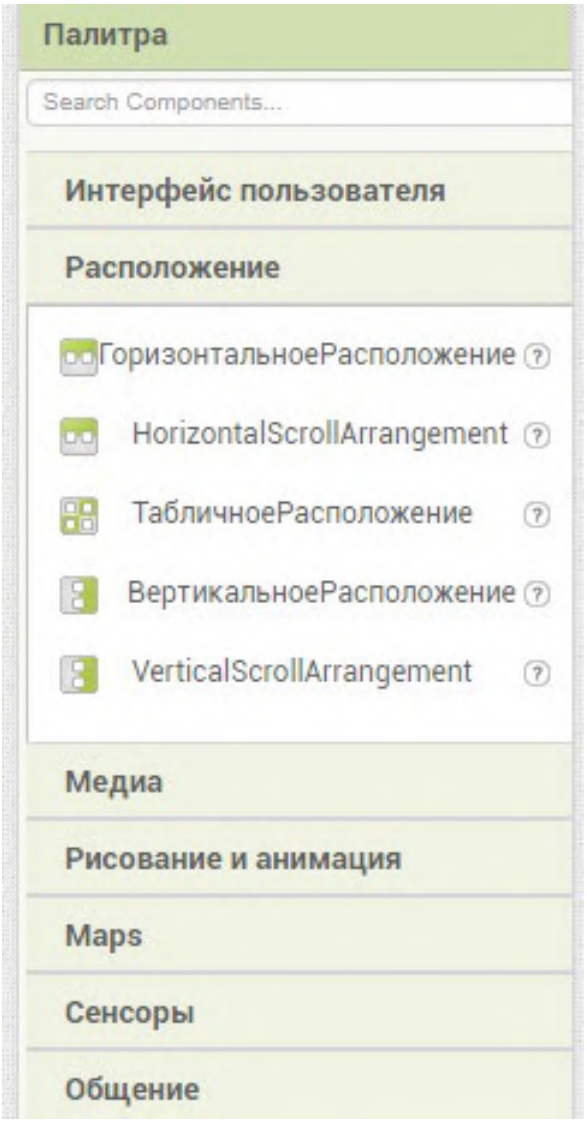


Рисунок 37. Функционал «Палитра» и все его компоненты

В разделе «Медиа» расположены компоненты для работы со звуком, видео и прочими мультимедиа.

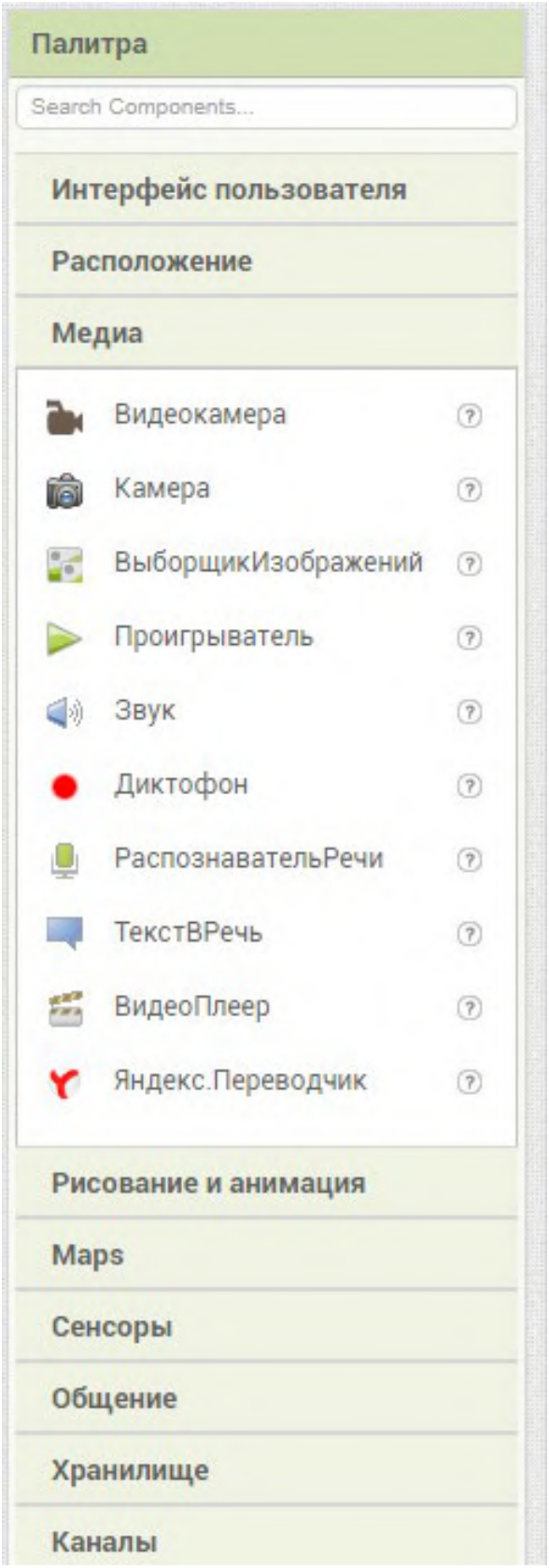


Рисунок 38. Функционал «Медиа»

Далее следует раздел «Рисование и анимация», содержащий средства для работы с графикой:

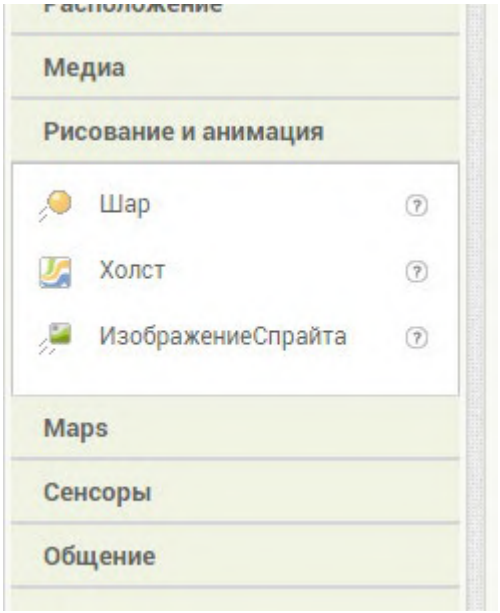


Рисунок 39. Функционал «Рисование и анимация»

Раздел «Карты» содержит компоненты, которые позволяют добавлять карты, выстраивать маршруты, навигацию, устанавливать маркеры и т.д.

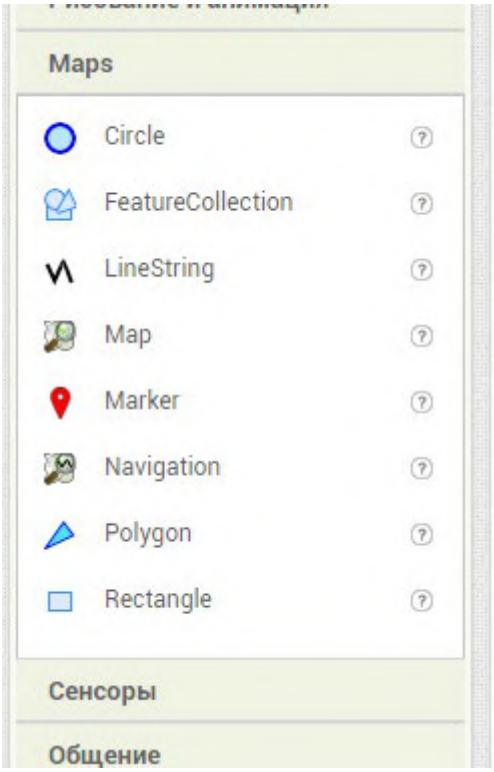


Рисунок 40. Функционал «Maps» или «Карты»

Следующий раздел — «Сенсоры» с компонентами Часы, Барометр, Шагомер и т.п.

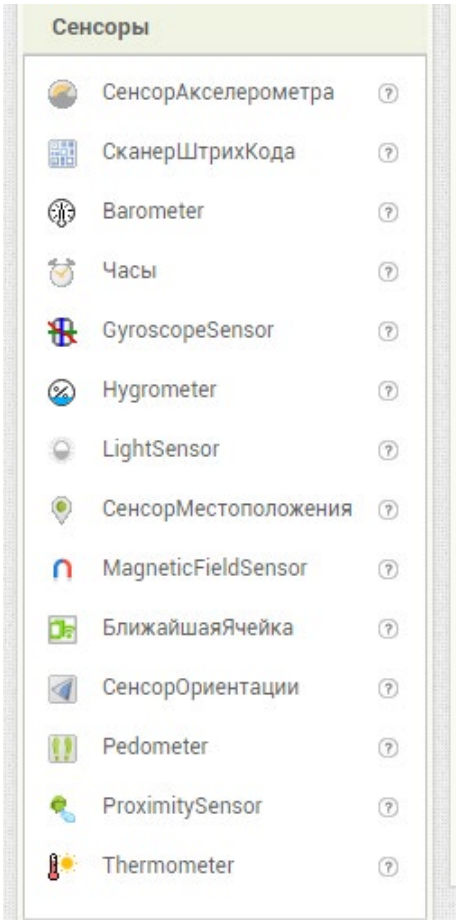


Рисунок 41. Функционал «Сенсоры»

Дальше идёт раздел «Общение», который позволяет реализовать общение в системе (набор номера, сборщик контактов, публикации и т.п.).

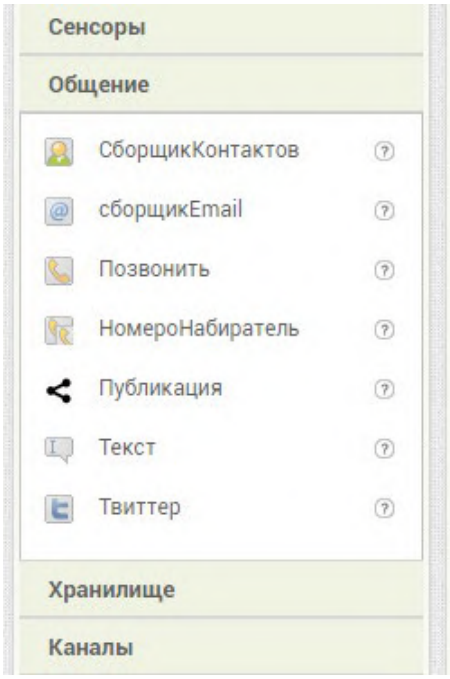


Рисунок 42. Функционал «Общение»

Далее следует раздел «Хранилище» с компонентами, представляющими различные базы данных, файлы и службы для хранения и управления информацией.

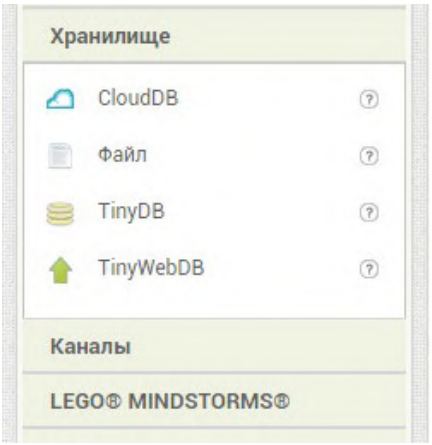


Рисунок 43. Функционал «Хранилище»

Раздел «Каналы» содержит компоненты, отвечающие за подключение к различным беспроводным устройствам.

Раздел «LEGO MINDSTORMS» содержит компоненты, которые отвечают за программирование роботов «LEGO». У компании LEGO есть отдельная линейка роботов, которых можно запрограммировать с помощью App Inventor.

Следующее окно «Компоненты» содержит все компоненты, которые используются на экранах в текущем проекте. Компоненты представлены в упорядоченной иерархической форме. Каждый компонент можно переименовать или удалить.

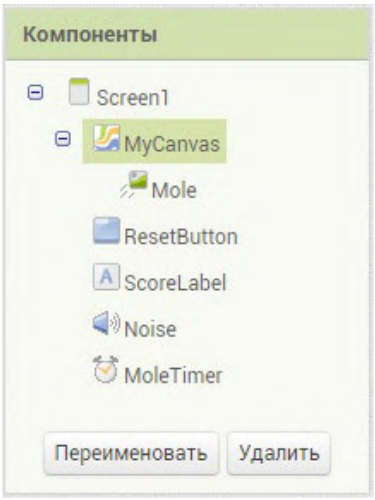


Рисунок 44. Функционал «Компоненты»

Окно «Медиа» используется для загрузки картинок, звуков, видео. Имеет простейший интерфейс загрузки элементов. При нажатии ЛКМ на мультимедиаэлементы их можно просмотреть, удалить или загрузить на компьютер.

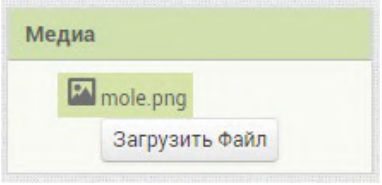


Рисунок 45. Функционал «Медиа»

Окно «Свойства» позволяет точечно настраивать свойства для каждого из компонент. Очевидно, что данное окно будет иметь различный вид для различных компонент. У каждого из компонент палитры обычно имеется свой уникальный набор свойств, хотя некоторые компоненты со сходным функционалом могут иметь почти одинаковый набор свойств.

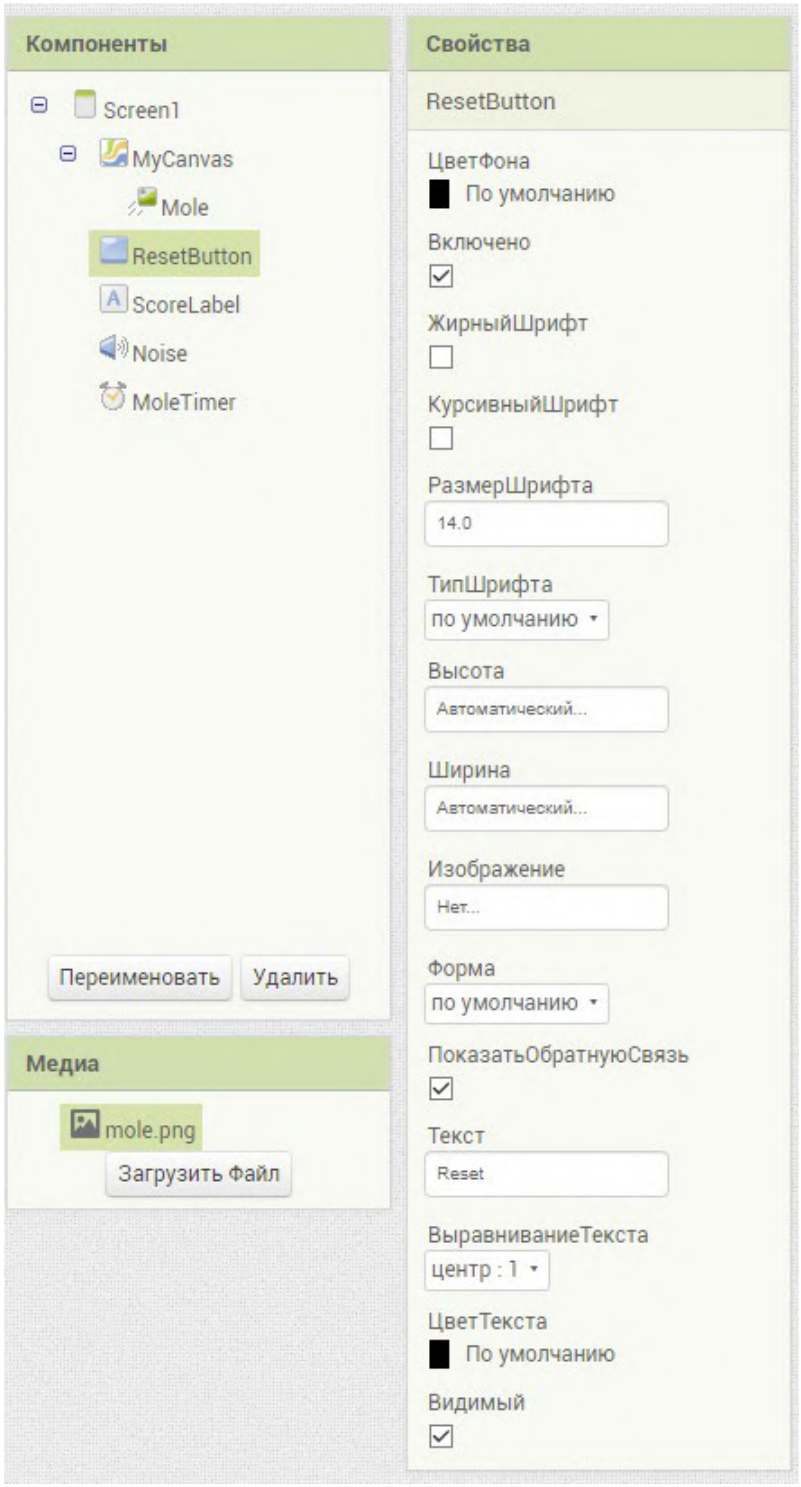


Рисунок 46. Функционал «Свойства»

Например, у кнопки можно настроить форму кнопки, цвет текста, цвет фона, изображение на кнопке, выравнивание текста, видима или нет кнопка, режимы ширины и высоты, настройки шрифта, текст кнопки.

Режим блоков

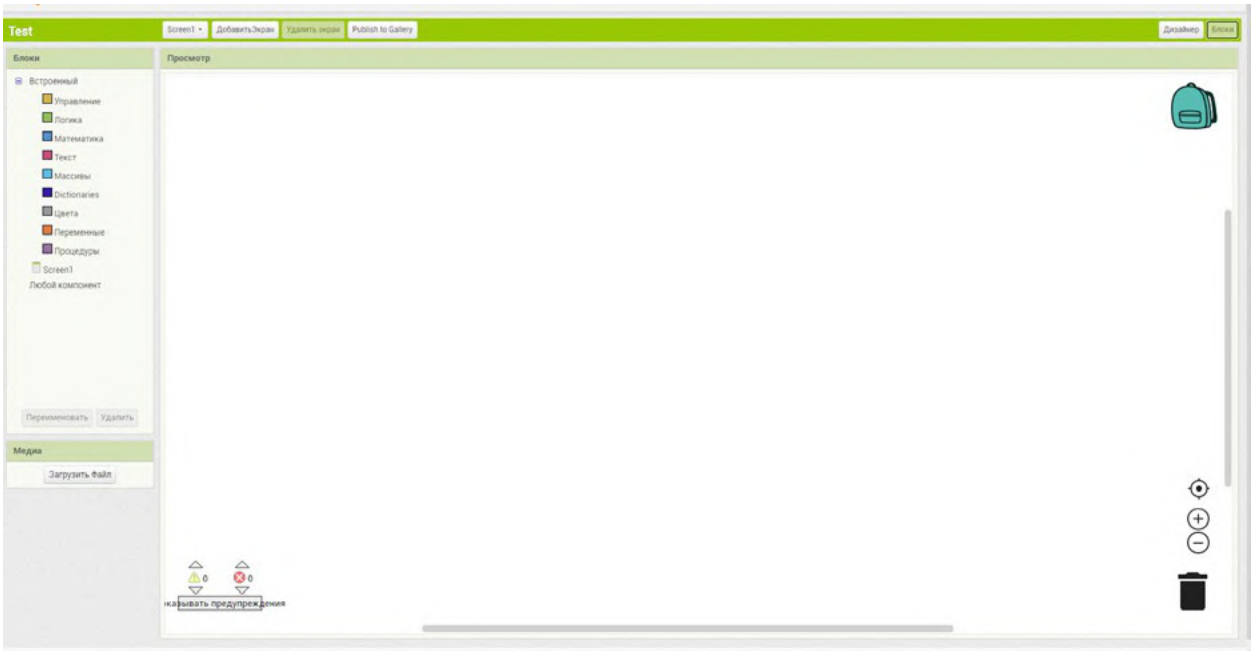


Рисунок 47. Окно «Блоков»

Каждый элемент блока представляет из себя некий пазл, который нужно соединять с другими пазлами, чтобы создавать готовые программы. Манипуляции с блоками очень просты, как в детской игре «пазлы». Необходимо соединять блоки между собой, исходя из логики работы и взаимодействия компонент. Таким образом, режим блоков отвечает за «внутренности» программы, т.е. за то, что должно происходить, когда должно происходить и каким образом.

Например, если рассмотреть раздел «Управление» из окна Блоки, то он представлен управляющими структурами:

Рисунок 48. Раздел «Управление» из окна «Блоки»

Первый блок «если ... то» представляет собой условный оператор:

if(условие){ делать } .

Второй блок «если ... то ... иначе» представляет собой условный оператор:

if(условие){ делать } else { делать }

Третий блок «если ... то ... иначе если ... то ...иначе» есть не что иное, как условный оператор вида:

if(условие){ делать } else if(условие){ делать } else { делать }

Четвертый блок является циклом типа for (со счётчиком):

for(счетчик от 1; до 5; шаг 1){ делать }

Пятый блок является циклом типа foreach:

foreach(Элемент в списке){ делать }

Шестой блок итерирует в словаре по записям типа «ключ-значение».

Седьмой блок является циклом типа while (до тех пор, пока выполняется условие):
while(условие){ делать }
И т. д.
Имеются следующие разделы встроенных блоков в окне Блоки:
— Управление (отвечает за некое управление в приложении)
— Логика (отвечает за логику приложения)
— Математика (отвечает за логические вычисления)
— Текст (отвечает за текст)
— Массивы (отвечает за массивы или списки)
— Dictionaries (отвечает за работу со списками типа «ключ-значение»)
— Цвета (отвечает за цветовую гамму чего-либо)
— Переменные (отвечает за некие переменные, которые мы дадим)
— Процедуры (отвечает за процедуры)

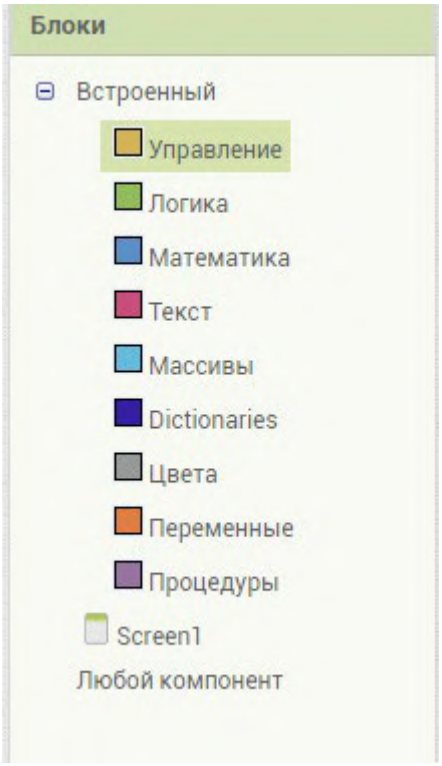


Рисунок 49. Встроенные типы блоков

В разделах встроенных блоков содержится огромное множество блоков разного назначения. Все эти блоки и тем более их комбинации рассмотреть в рамках данного курса не представляется возможным. Более подробно работа с конкретными блоками объясняется в ходе лабораторных работ.

Установка эмулятора

Чтобы протестировать или просто посмотреть, как работает созданное приложение, необходимо либо подсоединить мобильное устройство, либо установить эмулятор. Подробные инструкции по установке эмулятора см. в [2], [3]. Процесс установки очень прост. Достаточно скачать установочную программу **MIT_Appinventor_Tools_2.3.0** размером около 80 Мб на жёсткий диск компьютера:

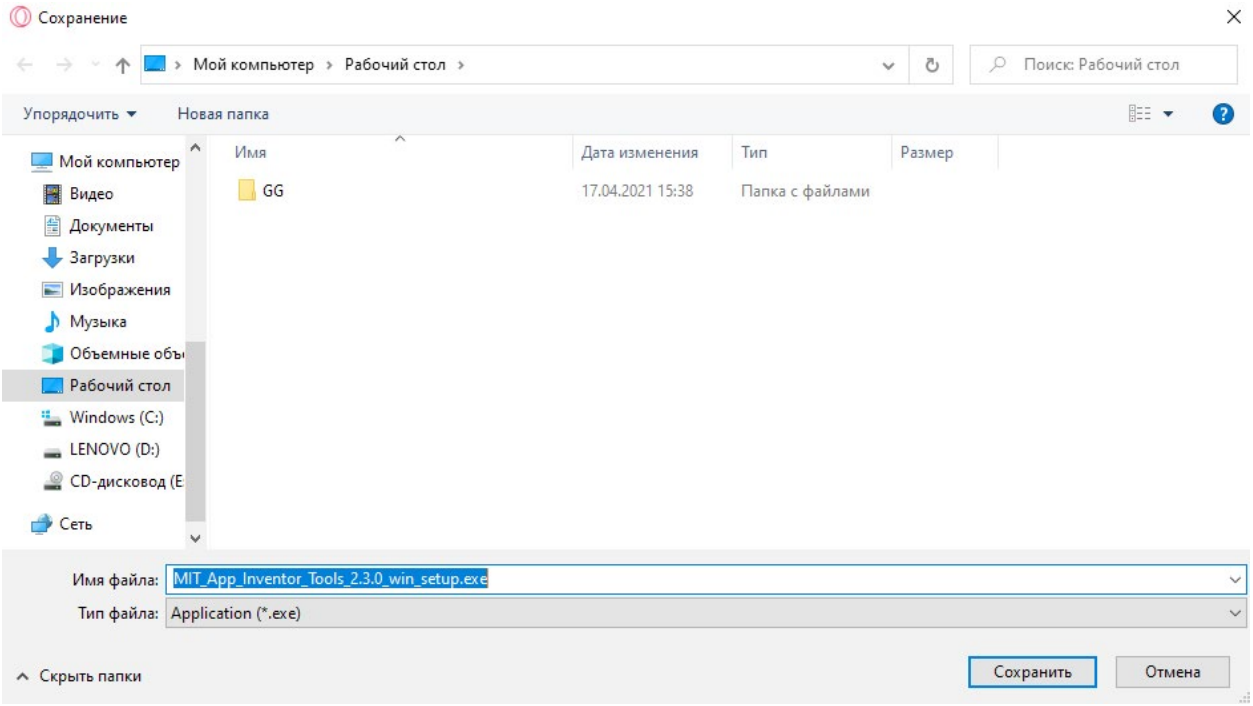


Рисунок 50. Выбор расположения файла **MIT_Appinventor_Tools_2.3.0.exe**

Затем нужно запустить её и выполнить простейший типовой процесс установки.

После этого в меню Пуск появится приложение aiStarter в папке MIT App Inventor Tools:

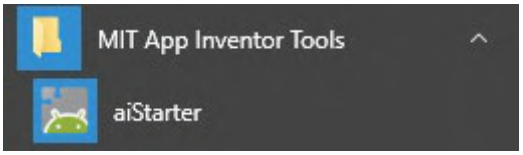


Рисунок 51. Значок приложения aiStarter в меню Пуск

Далее, для того чтобы эмулятор работал, необходимо вначале запустить aiStarter.

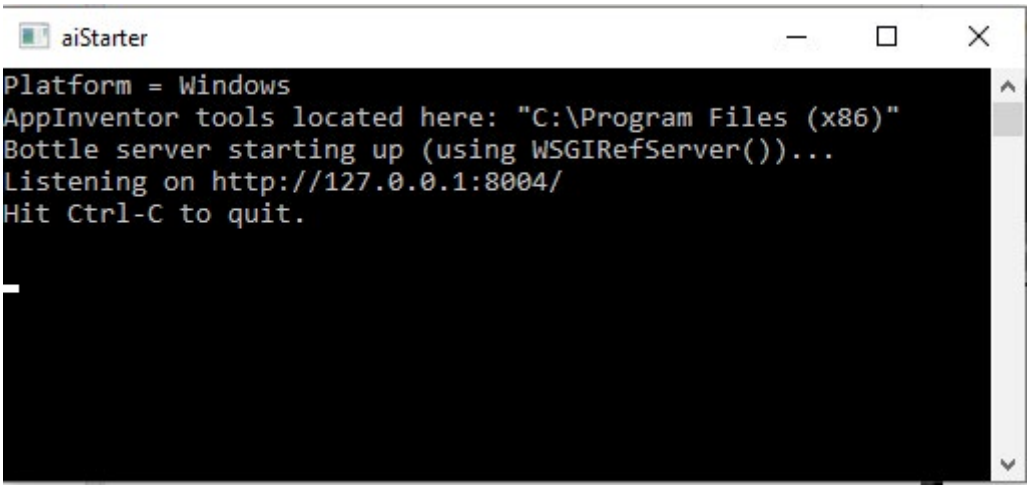


Рисунок 52. Окно приложения aiStarter

И только после этого нужно вызвать эмулятор через кнопку Подключиться в главном меню среды App Inventor.

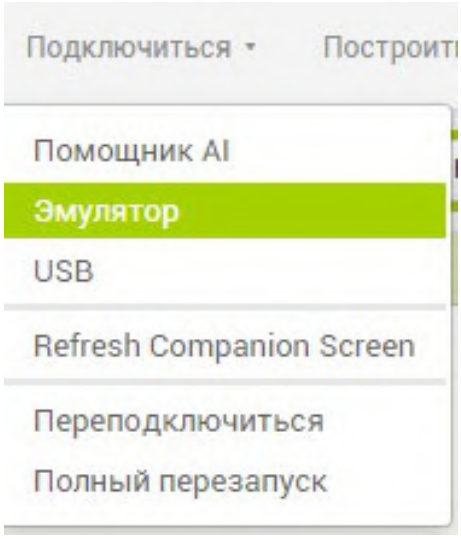


Рисунок 53. Пункт эмулятор в меню Подключиться

После установки приложения aiStarter и вызова эмулятора в меню Подключиться требуется некоторое время для запуска эмулятора мобильного устройства.

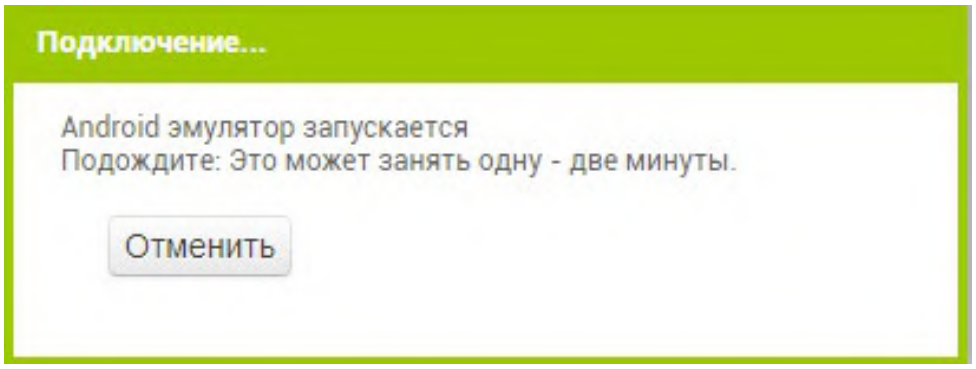


Рисунок 54. Ожидание запуска эмулятора.

Если всё пройдёт успешно, то эмулятор запустится и на отображаемом мобильном устройстве можно протестировать работу приложения.

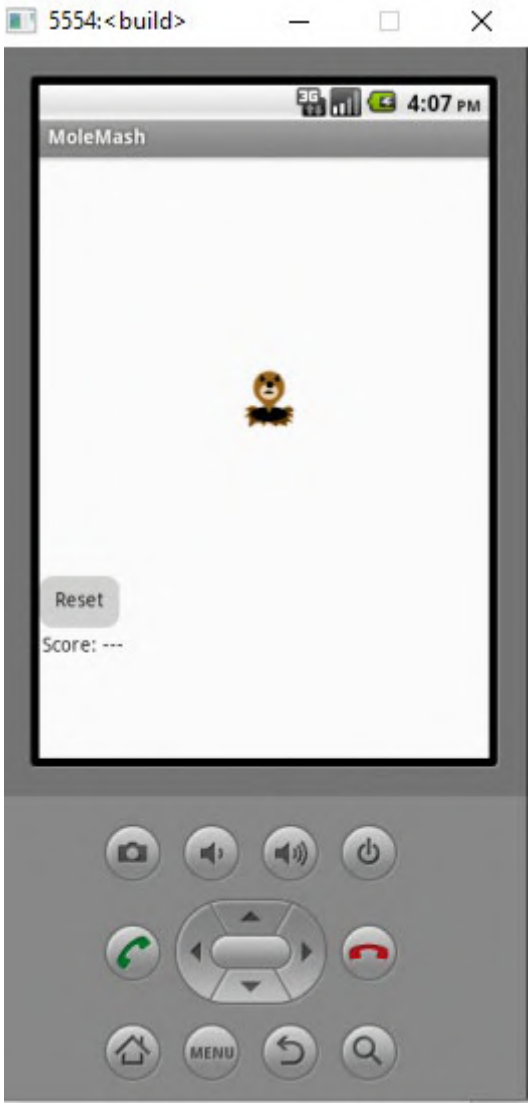


Рисунок 55. Работа приложения в эмуляторе

Далее с эмулятором можно работать так же, как и с обычным мобильным устройством.

Особенностью работы эмулятора является возможность вносить изменения в приложение без перезагрузки эмулятора. Все изменения практически немедленно отображаются на экране эмулятора, например, если требуется поменять цвет компоненты, или добавить новую, или внести изменения в блоки. То же самое касается и соединения по USB в режиме отладки. В некоторых случаях возможны критические ошибки, тогда может появиться необходимость перезапуска эмулятора.

Иногда эмулятор предлагает обновить помощника Ai2 на эмулируемом смартфоне. После обновления необходимо нажимать кнопку Done, а не Open.

Установка USB-соединения

Отладка и загрузка приложения в режиме USB происходят намного быстрее, чем в режиме эмулятора. При установке данного вида соединения требуется мобильное устройство с системой Андроид.

Необходимо скачать приложение AI Companion с помощью службы Google Play (Play Market) и установить его на устройстве.

Далее необходимо включить Режим Разработчика в настройках устройства. Для этого требуется зайти в настройки устройства и найти пункт «Система» -> «О телефоне» (или «О планшете») -> «Номер сборки». После чего надо 7 раз быстро нажать на пункт «Номер сборки», и пункт «Параметры разработчика» становится доступным в настройках устройства.

После этого остаётся найти в настройках данный пункт, зайти в него и включить опцию «Отладка по USB».

Подробные инструкции по установке доступны по ссылке: [9].

Основные приёмы работы с блоками App Inventor

Если режим Дизайнер АИ отвечает за создание различных компонент приложения (кнопки, звук, карты и пр.) и настройку их свойств по умолчанию, то режим блоков отвечает за поведение этих компонент, за их взаимодействие с пользователем, друг с другом и с системой. Если придерживаться терминов многослойной архитектуры, можно отметить, что режим Дизайнер отвечает за слой отображения (View), а режим блоков отвечает за слой контроллера и модели (Model, Controller).

Здесь описываются приёмы работы с блоками в целом. Конкретные примеры и случаи применения представлены в цикле лабораторных работ данного курса.

Здесь и далее при описании названия блока в зависимости от контекста используется либо дословное название «когда Кнопка1.Щелчок», либо описание вида: «обработчик события нажатия на Кнопку1», либо сокращённое описание «обработчик нажатия на кнопку»/«обработчик кнопки»:

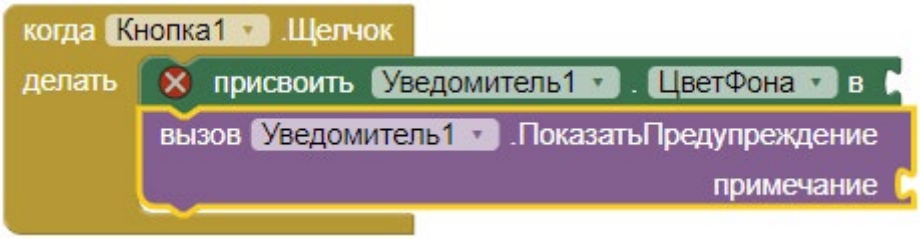


Рисунок 56. Обработчик события нажатия на Кнопку1

Основные типы блоков

Режим блоков представлен тремя основными типами блоков:

Встроенные блоки (разделы Управление, Логика, Математика, Текст, Массивы и т.д.). Здесь представлены блоки для построения типовых алгоритмов, управляющих структур, вызова процедур, создания массивов и списков, работы со строками, цветовые и логические константы. Встроенные блоки могут использоваться в привязке с компонентами приложения.

Блоки компонент приложения (раздел, соответствующий названиям экранов приложения, по умолчанию Screen1). Здесь представлены блоки, связанные с конкретными компонентами приложения, представленными в окне Палитра в режиме Дизайн: всё, что касается интерфейса пользователя (кнопки, текстовые окна, надписи, уведомители и т.д.), медиакомпоненты (звук, видео, камера и т.д.) и всех прочих групп компонент (карты, сенсоры, контакты, СМС, Интернет, рисование и анимация и пр.).

Блоки компонент, соответствующих типам компонент приложения (раздел Любой компонент). Здесь представлены блоки, позволяющие упростить работу с множеством идентичных компонент [4].

Основные типы блоков компонент приложения

Блоки компонент бывают следующих видов:
Обработчик (слушатель) события компоненты (коричневый цвет)
Вызов процедур или команды для компоненты (синий цвет)
Геттер, получатель свойства компоненты (зелёный цвет)
Сеттер, установщик свойства компоненты (зелёный цвет)
Геттеры и сеттеры для переменных (оранжевый цвет)
У встроенных блоков свои отдельные цвета (математические, массивы и словари — оттенки синего, логика — зелёный, текст — бордовый, управление — коричневый).

Соединение блоков

Работа с блоками очень проста. Они представляют собой пазлы, которые вкладываются друг в друга. Вкладывание осуществляется нажатием на ЛКМ. Необходимо перетащить один блок в другой, удерживая нажатой ЛКМ. Например, если требуется в блок слушатели событий Кнопки1 добавить команду присвоения Фону Надписи1 Розового цвета, можно выполнить следующую последовательность действий:

Рисунок 57. Меню блоков компоненты Кнопка1

Нажать на Кнопка1 в разделе Screen1:

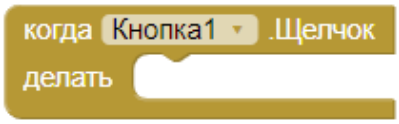


Рисунок 58. Обработчик нажатия на кнопку

Перетащить ЛКМ блок слушателя Кнопки1 «когда Кнопка1.Щелчок»:

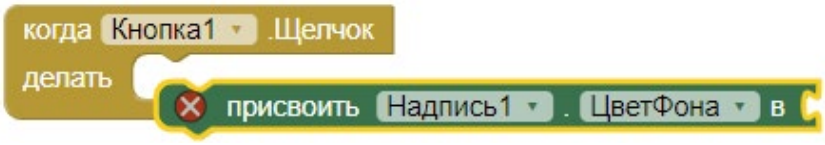


Рисунок 59. Формирование блоков внутри обработчика

Нажать на Надпись1 в разделе Screen1, выбрать блок и перетащить блок сеттер Надписи1 «присвоить в» внутрь блока слушателя Кнопки1. Необходимо совместить выемку одного блока и выступающий крепёж другого блока.

Рисунок 60. Незавершенная комбинация с сеттером (блоком присвоения)

Выбрать из раздела Цвета встроенных блоков нужный цвет:

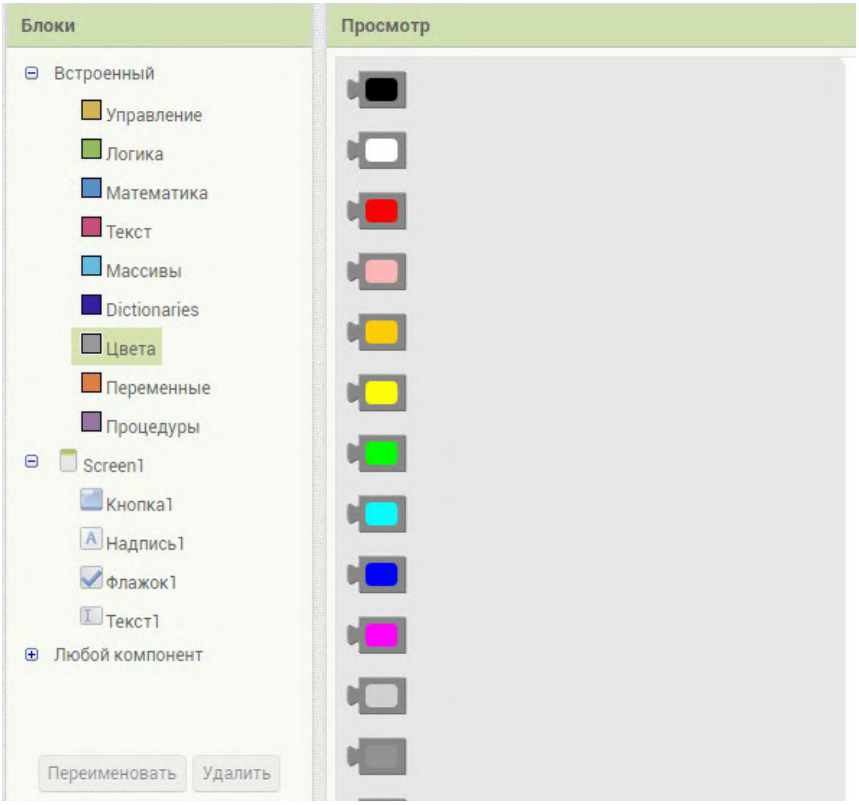


Рисунок 61. Выбор цвета

Перетащив соответствующий блок в окно просмотра, подцепить его справа к блоку сеттера Надписи1:

Рисунок 62. Присвоение цвета фону компоненты Надпись1

Затем необходимо совместить выемку одного блока и выступающую часть другого блока.

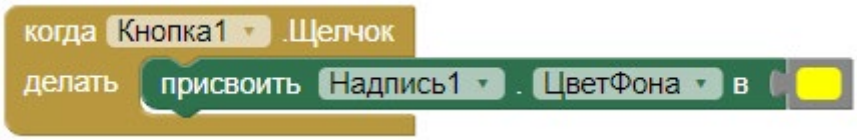


Рисунок 63. Завершение присвоения цвета фону компоненты Надпись1

Все подобные операции многократно прорабатываются при выполнении лабораторных работ.

Способы крепления

По способу крепления блоки делятся на 4 большие группы (на самом деле первично место блока в организации программного потока, а вторичным проявлением этого выступает способ крепления, но для удобства можно отталкиваться от вторичного, более наглядного проявления).

Блоки 1-го уровня — независимые блоки, которые не требуют наличия других блоков, чтобы находиться в окне просмотра. Обычно это блоки про-цедур и слушателей, инициализаторы переменных. Выступающий крепёж блока или выемка блока находится внутри блока. Например:

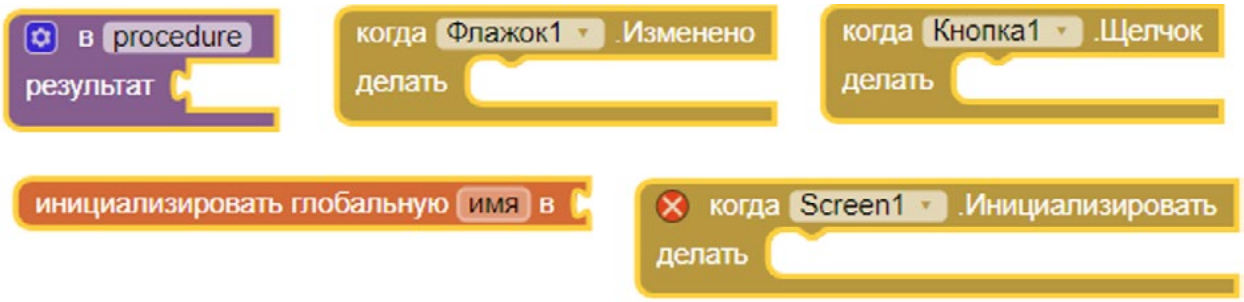


Рисунок 64. Блоки 1-го уровня

Блоки 2-го уровня первого типа, вставляемые в независимые блоки, т.е. которые требуют наличия независимых блоков. Обычно это блоки сеттеров и команд. Выемка находится сверху. Выступающий крепёж — снизу. Часто к этой выступающей части прикрепляются другие блоки 2-го уровня. Справа находится выемка для блоков 3-го или 4-го уровня. Например:

Рисунок 65. Блоки 2-го уровня 1 типа

Блоки 2-го уровня второго типа также вставляются в независимые блоки, но возвращают результат. Обычно это блоки геттеров или команд, проверок, возвращающие результат. Выступающий крепёж находится слева, а выемка справа. **Выемки справа можно назвать входами, а выступающие части слева — выходами.** Например:

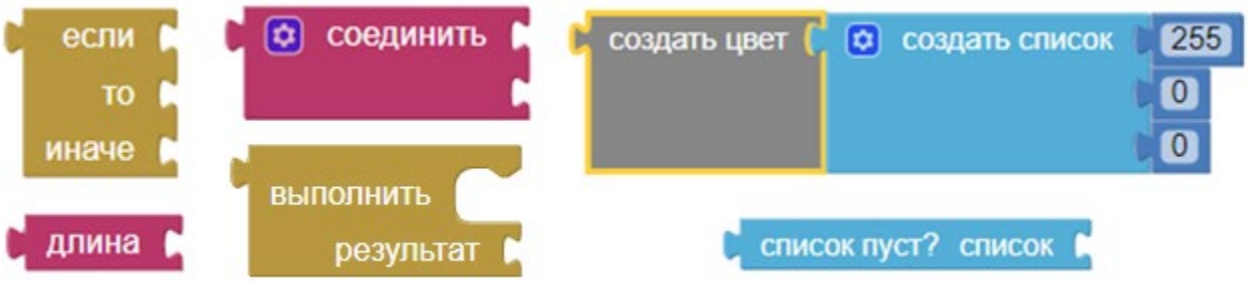


Рисунок 66. Блоки 2-го уровня 2-го типа

Блоки 3-го уровня терминальные или завершающие комбинацию блоки. Блоки со значениями чисел, цветов, логических переменных, геттеры. Выемка блока находится слева. Справа крепежи отсутствуют.

Рисунок 67. Блоки 3-го уровня

Переменные

В среде АИ применяются глобальные (уровня всего экрана) и локальные (уровня конкретного блока) переменные.

Для работы с переменными необходимо перейти в режим Блоки. Блоки для работы с переменными находятся в разделе Переменные встроенных блоков.

Для инициализации глобальной переменной необходимо перетащить блок «инициализировать глобальную «имя» в»:

Рисунок 68. Инициализация глобальной переменной

Задать значение глобальной переменной в блоках можно следующим образом:

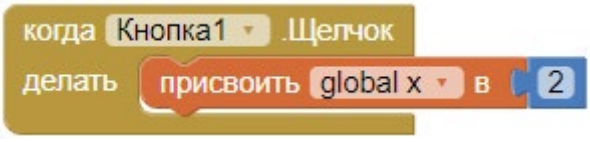


Рисунок 69. Присвоение значения глобальной переменной в блоке

Для получения значения глобальной переменной можно использовать геттер:

Рисунок 70. Геттер глобальной переменной

Рисунок 71. Присвоение тексту кнопки значения глобальной переменной x

С помощью следующего блока можно проинициализировать локальную переменную в блоке:

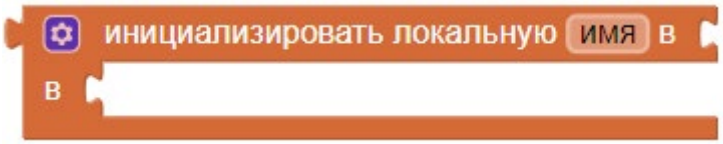


Рисунок 72.

В рамках данного курса рассматриваются только глобальные переменные.

Не следует путать переменные с параметрами компонент режима Дизайн. Например, у компоненты Камера имеется параметр «изображение»:

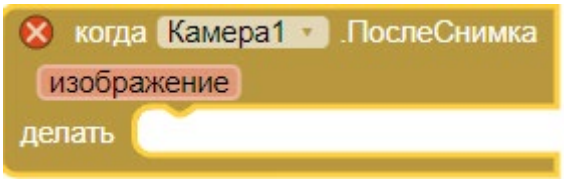


Рисунок 73. Обработчик события завершения съёмки

Параметр предоставляет доступ к только что сделанному снимку.

Мутаторы

Мутаторы вызываются через знак **шестерёнки** и позволяют изменить количество входов или выходов на блоках. Например, при работе с текстовым блоком, склеивающим текст, можно тем самым увеличить количество соединяемых строк, которое изначально равно 2.

1.	2.
	

Продолжение

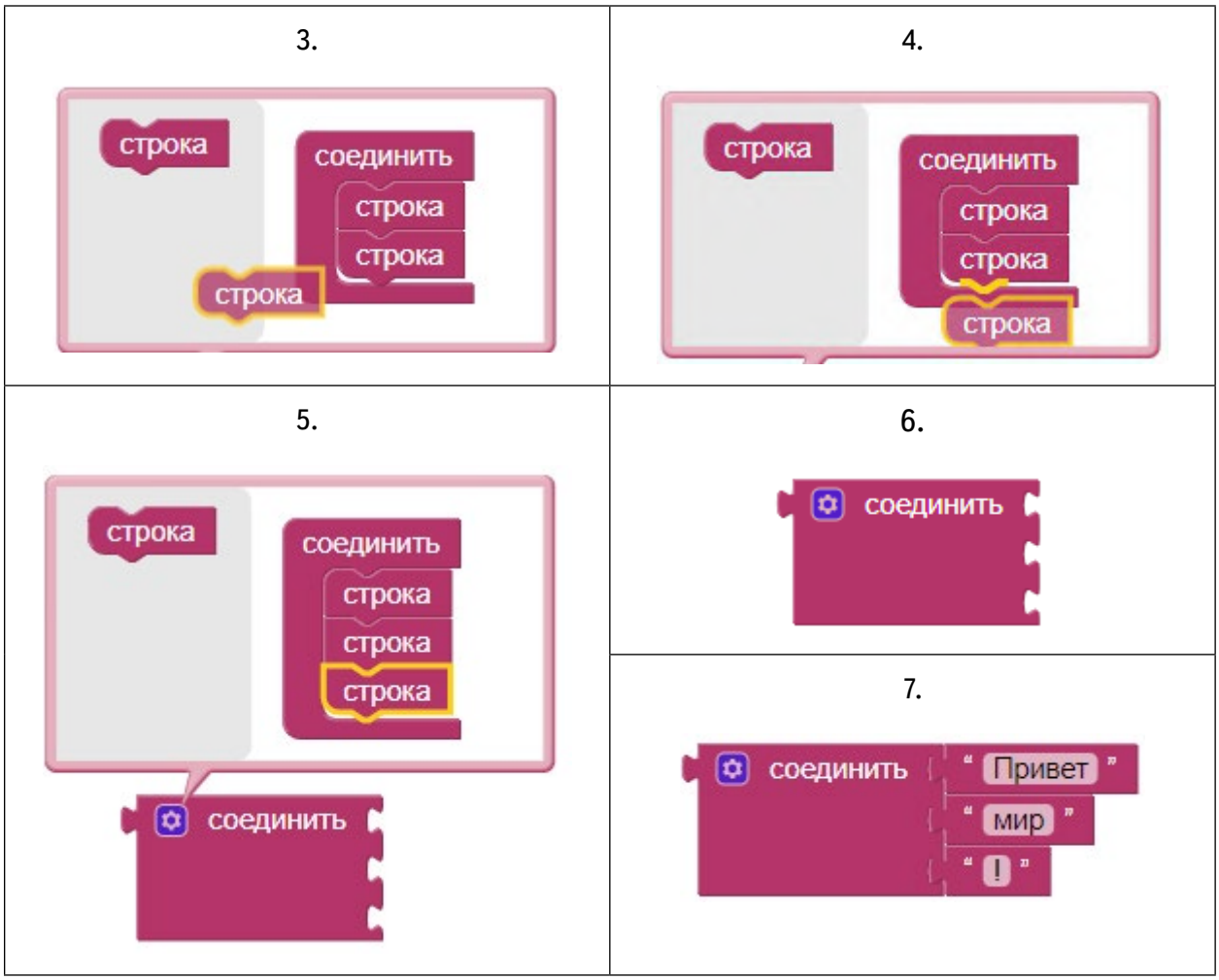
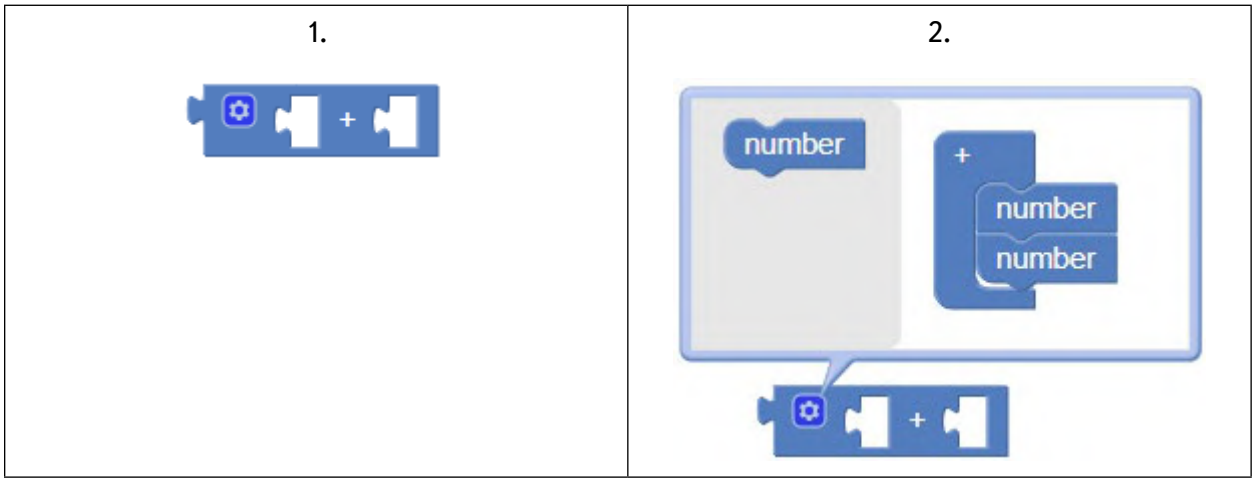


Рисунок 74. Последовательность работы с мутатором блока «соединить»

Мутаторы имеются на многих блоках. Широко применяются в блоках раздела Математика, например, при изменении количества слагаемых в операции суммирования и пр.



Продолжение

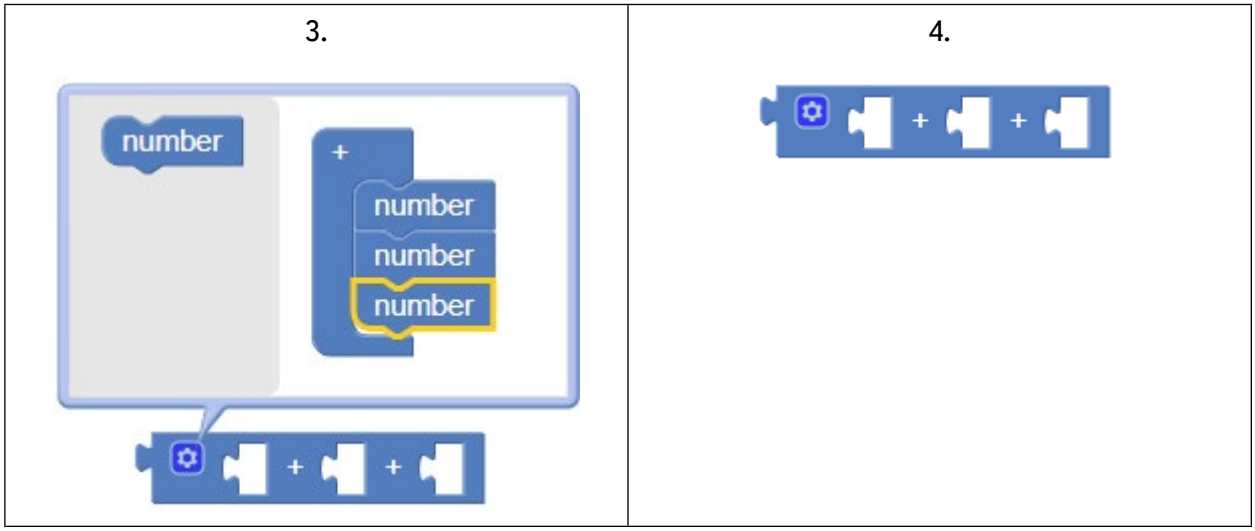


Рисунок 75. Последовательность работы с мутатором блока суммирования

Процедуры

Особое место занимают процедуры из раздела Процедуры. С помощью этих блоков создаются команды, которые затем вызываются по мере необходимости. Например, когда происходит какое-то событие, связанное с Кнопкой, или срабатывает таймер на Часах. В исходной справке АИ [5] приводится пример с использованием процедуры, задающей координаты для случайного появления спрайта в игре MoleMash:

Рисунок 76. Пример процедуры из игры MoleMash[5]

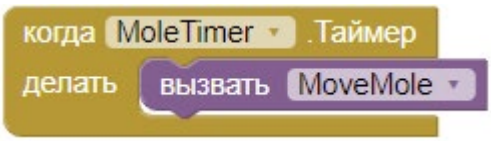


Рисунок 77. Вызов процедуры в обработчике таймера

В АИ, как и во многих языках программирования, существует два типа процедур:

процедуры, выполняющие команды, как на рисунке с примером из игры MoleMash;

процедуры, возвращающие значение (очень удобны при многократном повторении определённых вычислений или просто для того, чтобы логику вычисления, держать в отдельном месте):

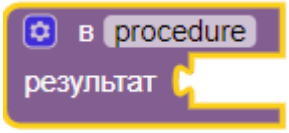


Рисунок 78. Блок процедуры, возвращающей результат

Процедуры можно переименовывать (в этом примере — в proc1).

Пример

Для умножения чисел a и b необходимо из раздела Математика добавить блок умножения, затем через мутатор добавить в процедуру аргументы a и b:

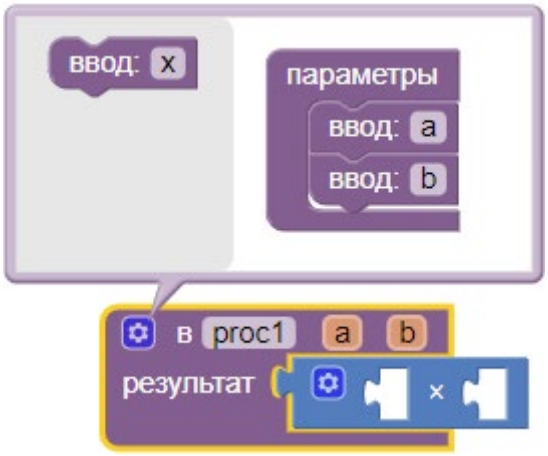


Рисунок 79. Применение мутатора

После этого следует установить зависимость результата от аргументов (параметров) процедуры от параметров.

Рисунок 80. Работа с параметрами процедуры

Для этого необходимо подвести курсор мыши к значку параметра «a» на процедуре. В появившемся контекстном окне выбрать блок «получить a» и перетащить его в пустой пазл синего математического блока умножения. То же самое — с параметром «b»:

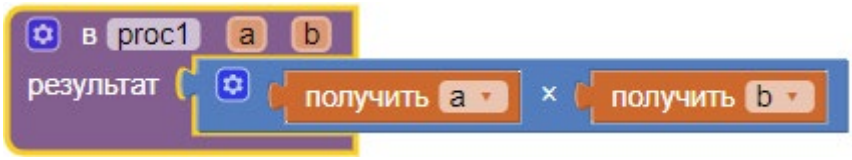


Рисунок 81. Процедура, возвращающая результат умножения a на b

Далее такую процедуру можно использовать при помощи блоков вызова процедур (они появляются в разделе Процедуры после создания процедур), например:

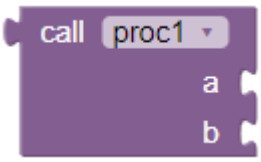


Рисунок 82. Блок вызова пользовательской процедуры

Данный блок на вход справа получает значения параметров a и b, а на выход слева отправляет результат их умножения и, например, может использоваться следующим образом:

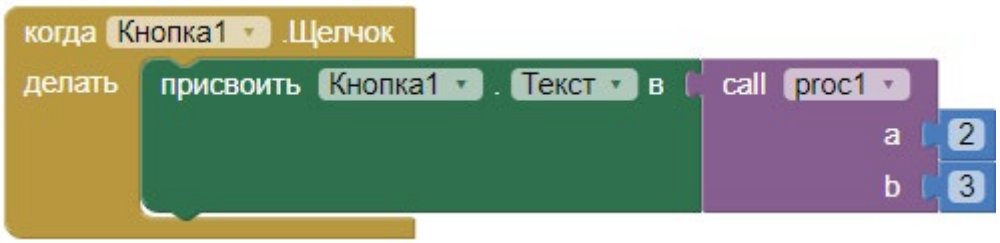


Рисунок 83. Пример использования блока вызова процедуры

Лабораторные работы

Когда упоминается тот или иной режим (Дизайн/Блоки), подразумевается, что пользователь должен переключиться в тот или иной режим через кнопки Дизайн и Блоки среды АИ.

Лабораторная работа 1. Знакомство со средой

Теоретическая часть

В данной работе в том числе используется теоретический материал из дидактических материалов.

Компонента Уведомитель представляет собой всплывающее сообщение с информацией для пользователей. Существуют различные процедуры вызова Уведомителя:

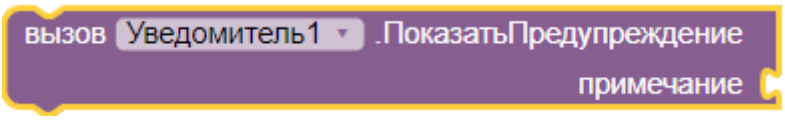


Рисунок 84. Вызов Уведомителя

Самый простой вариант Уведомителя, когда выводится текстовое сообщение. Пример использования вместе с кнопкой:

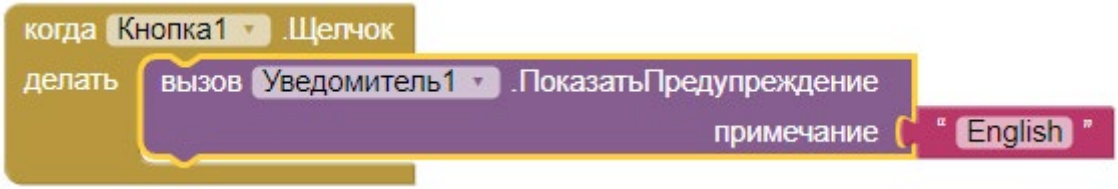


Рисунок 85. Вызов Уведомителя при нажатии на кнопку

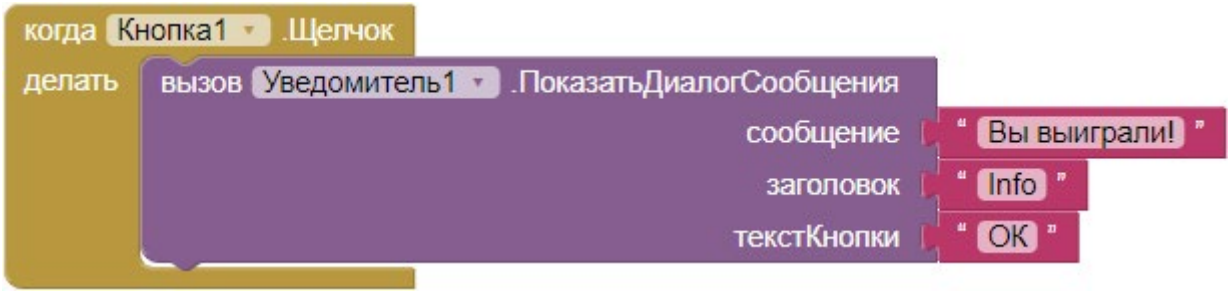


Рисунок 86. Уведомитель с заголовком, сообщением и кнопкой

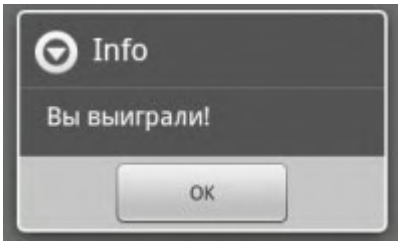


Рисунок 87. Пример окна уведомителя

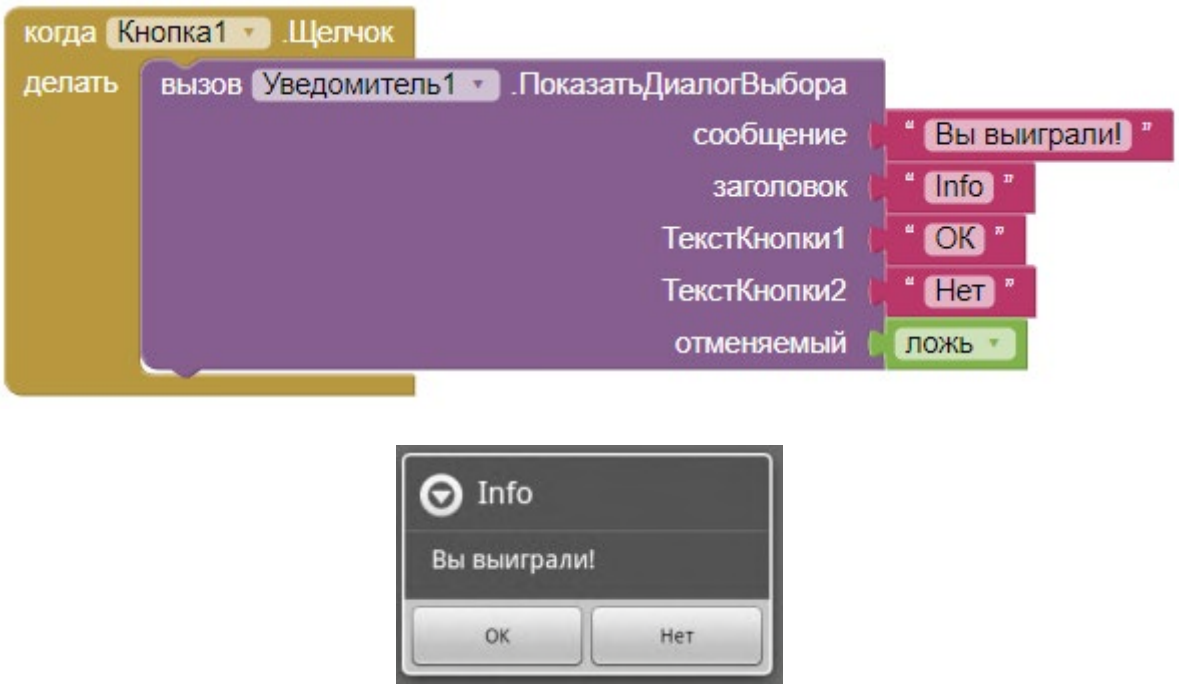


Рисунок 88. Пример блоков и окна уведомителя с двумя кнопками

Для обработки ответа пользователя (кнопки 1 или кнопки 2) можно воспользоваться следующим блоком (если пользователь выбрал ОК, то в Надписи1 появляется текст «ОК»):

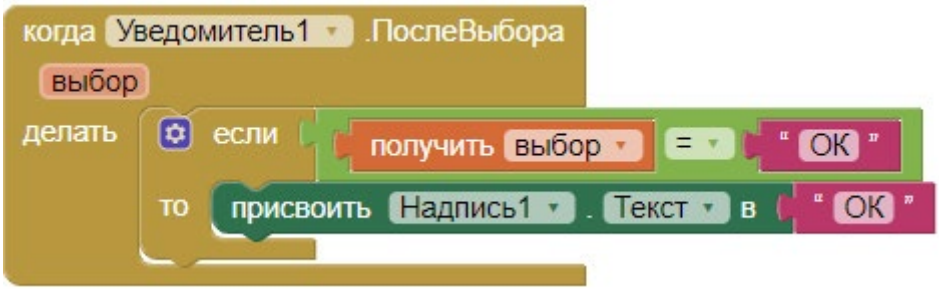


Рисунок 89. Пример обработки ответа пользователя

Практическая часть

Цель работы: зарегистрироваться для работы с АИ, ознакомиться со средой редактора АИ, создать первый проект, создать приложение, выводящее сообщение «Hello, world!» при нажатии на кнопку.

Ход работы

1. Скачать установщик **MIT_Appinventor_Tools.**
2. Установить дистрибутив и запустить приложение aiStarter.
3. Перейти по адресу http://ai2.appinventor.mit.edu/ и запустить среду АИ (авторизоваться на сайте «App Inventor» при необходимости).
4. Через пункт главного меню «Проекты» -> «Начать новый проект ...» создать новый проект под названием FirstProject.
5. Перенести компоненты из раздела **Интерфейс пользователя** на экран приложения (Screen1) согласно следующей схеме:

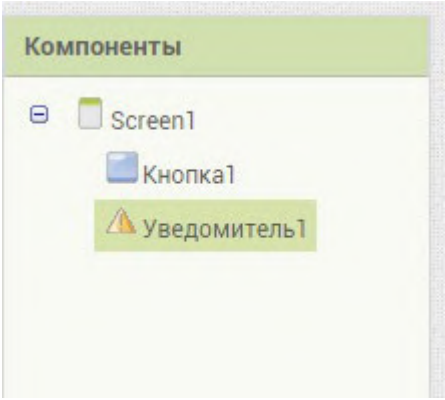


Рисунок 90. Схема компонент

6. Для этого в режиме Дизайнер перетащить ЛКМ на главный экран приложения компоненты Кнопка1 и Уведомитель1.
7. Для главного экрана Screen1 установить свойства:
   Заголовок — FirstProject
   Theme — Device Default
8. Для кнопки установить свойства:
   Высота, Ширина — Наполнить родительский.
   Текст — «Привет!»

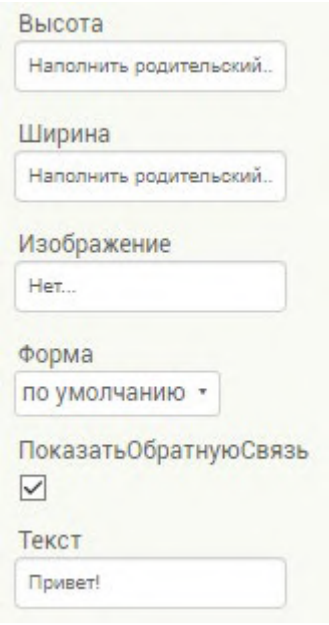


Рисунок 91. Свойства кнопки

9. Проверить, что финальная схема компонент соответствует следующему дизайну:

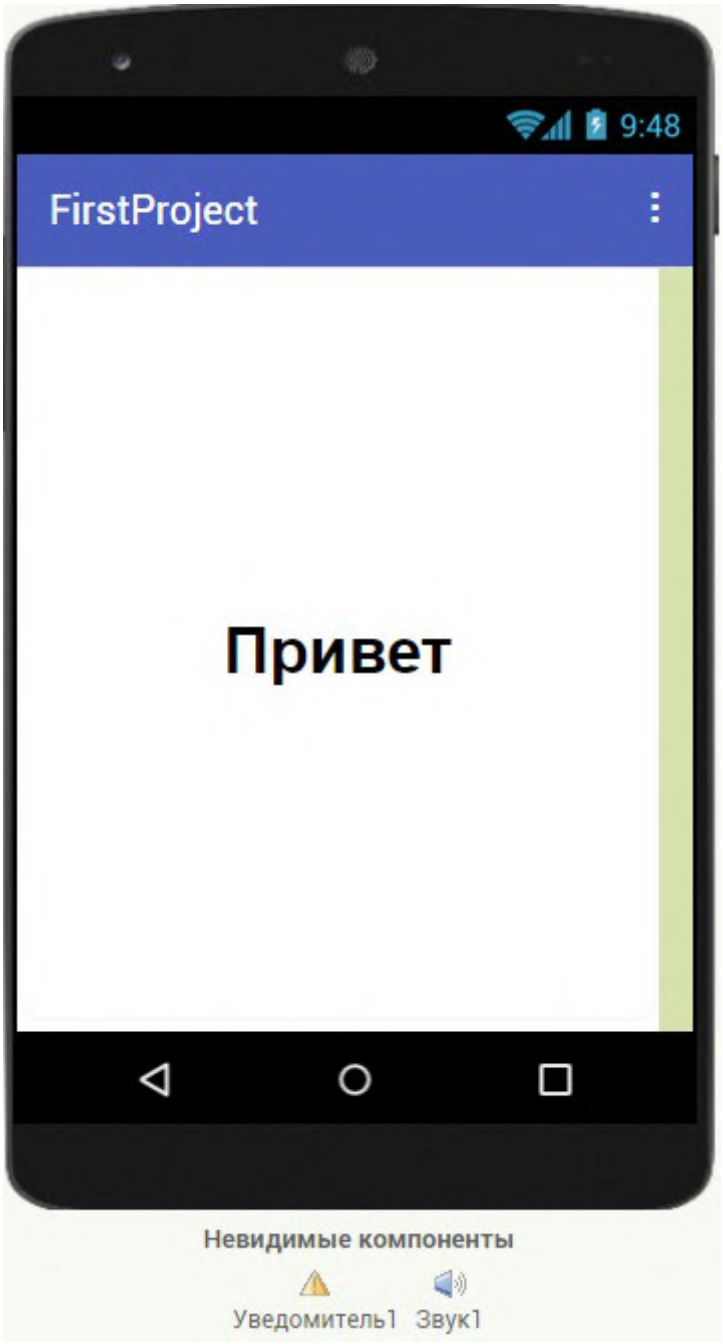


Рисунок 92. Дизайн приложения

10. Перейти в режим Блоки и перетащить обработчик нажатия на Кнопка 1 в окно Просмотр:

Рисунок 93. Обработчик события нажатия на Кнопку1

11. Вставить в слушатель блоки Уведомителя1 «присвоить» и вызов процедуры «вызов.ПоказатьПредупреждение»:

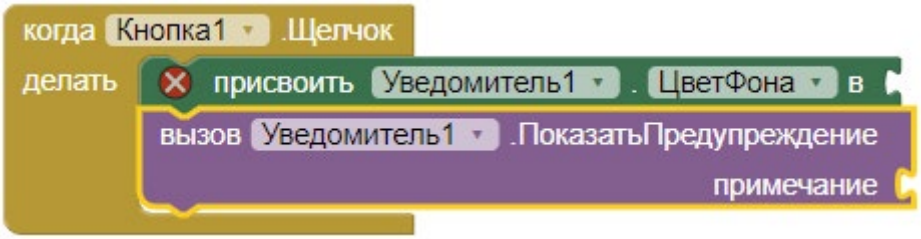


Рисунок 94. Промежуточный вид блоков обработчика

12. Установить для свойства ЦветФона зелёный цвет из блока Цвета, а для примечания текстовую строку «Hello, world!» из блока Текст.

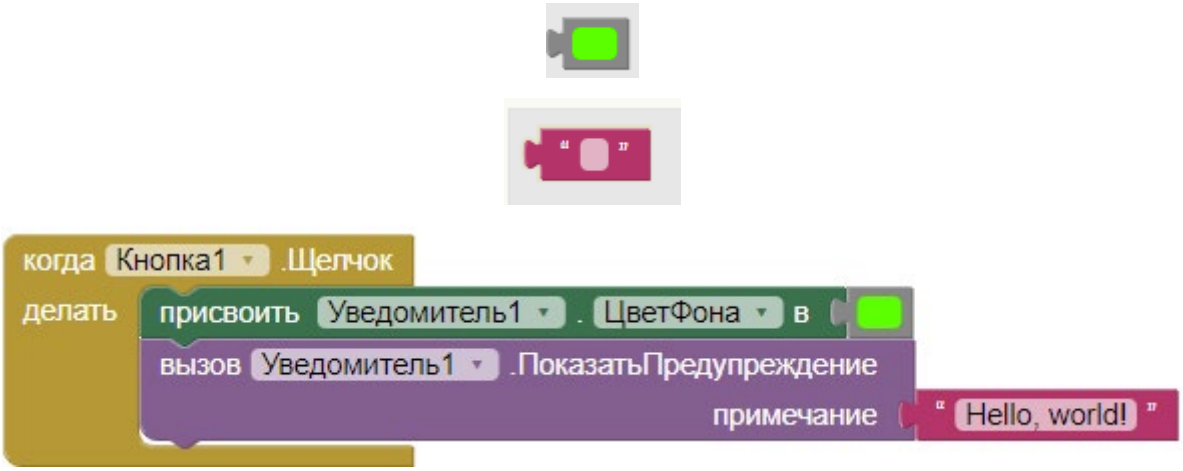


Рисунок 95. Итоговый вид блоков обработчика

13. Поскольку это мобильное приложение, то можно усилить сообщение эффектом вибрации, вернуться в режим Дизайн и добавить компоненту Звук1.

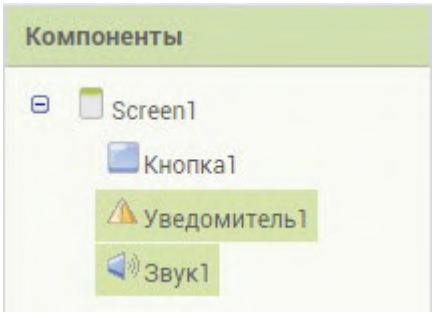


Рисунок 96. Новый вид схемы

14. Опять переключиться в режим Блоки и выбрать в окне Блоки у компонента Звук1 процедуру «вызов.Вибрировать». Из блока Математика перетащить компоненту «задаёт число» и установить её значение равным 500.

Рисунок 97.

15. Финальная комбинация блоков приобретёт следующий вид:

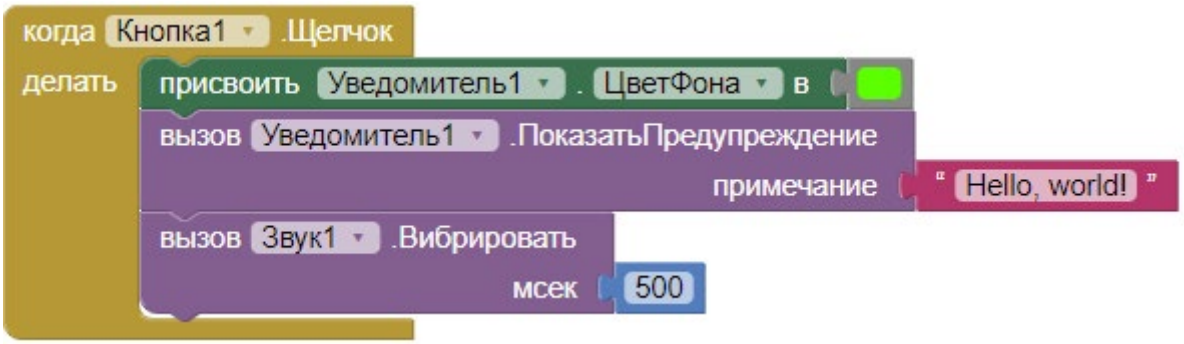


Рисунок 98. Новый вид блоков обработчика

16. Запустить эмулятор через кнопку меню «Подключиться»->«Эмулятор». Дождаться запуска эмулятора и запустить приложение.

Рисунок 99. Приложение в эмуляторе

17. Поэкспериментировать с цветами, шрифтами, фоном кнопки и главного экрана.

18. Изменить значения выводимого текста Уведомления и значение «Вибрировать м/сек» установить равным 250.

19. Проверить работу приложения в эмуляторе.

Выводы: в данной лабораторной работе производится ознакомление со средой АИ и разрабатывается приложение вида «Hello, world».

Контрольные вопросы:

1) Что такое App Inventor?

2) Что такое Эмулятор и для чего он нужен?

3) Из каких окон состоит среда разработки App Inventor?

Лабораторная работа 2. HelloWorld с посимвольным выводом

Теоретическая часть

В данной работе в том числе используется теоретический материал из дидактических материалов.

Компоненту экран очень удобно использовать для инициализации переменных. Например:

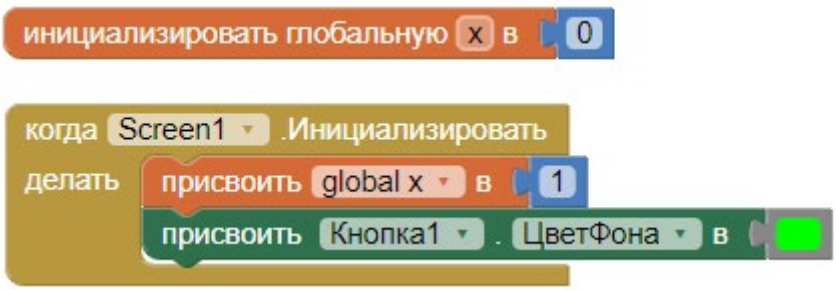


Рисунок 100. Инициализация переменных и свойств при запуске экрана приложения

Также может быть удобно использовать обработчик события поворота экрана:

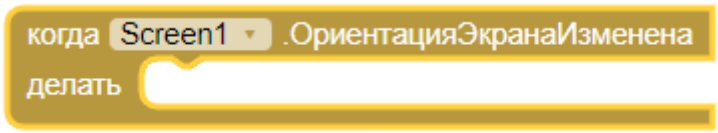


Рисунок 101. Обработчик поворота экрана

Практическая часть

Цель работы: создать приложение, состоящее из кнопки и надписи. При нажатии на кнопку выводить надпись «Привет, IT-куб!» на экран с интервалом в 1 секунду посимвольно.

Ход работы

1. Перейти по адресу http://ai2.appinventor.mit.edu/ и запустить среду АИ (при необходимости авторизоваться на сайте «App Inventor»).
2. Через пункт главного меню «Проекты» -> «Начать новый проект ...» создать новый проект под названием HelloITCube.
3. Перенести компоненты из раздела **интерфейс пользователя** на экран приложения (Screen1) согласно следующей схеме:

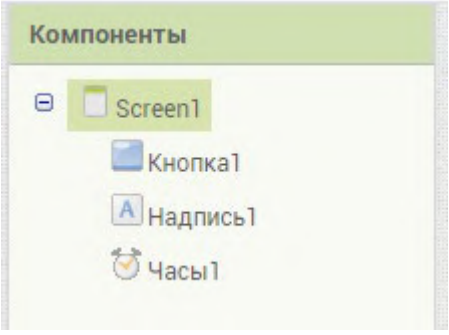


Рисунок 102. Схема компонент экрана

4. Для главного экрана Screen1 установить свойство:
   a. Заголовок — HelloITCube
   b. ВыровнятьПоГоризонтали — Центр
5. Для компоненты Кнопка1 установить свойства:
   a. Текст — «Hello, cube!»
   b. РазмерШрифта — 22
   c. ЖирныйШрифт — отметить галочкой
6. Для главного экрана Надпись1 установить свойство:
   a. РазмерШрифта — 22
   b. Текст — «Привет, IT-куб!»
7. Для компоненты Часы1 установить свойства:
   a. ТаймерВсегдаВключён — отмечено галочкой
   b. ИнтервалТаймера — 1000
8. Убедиться, что финальная схема компонент соответствует следующему виду:

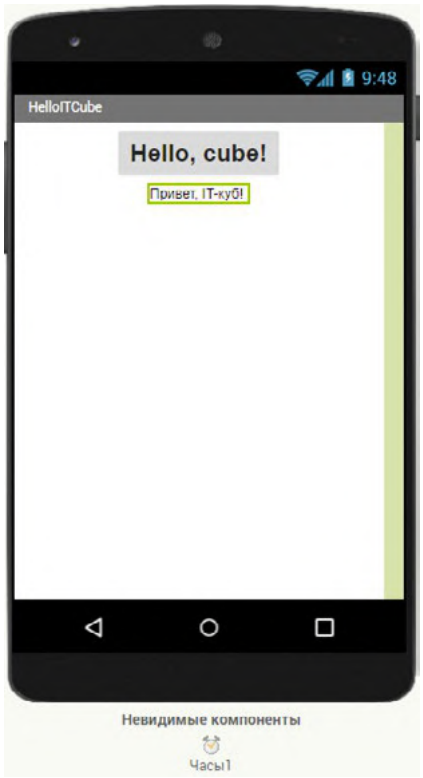


Рисунок 103. Дизайн приложения

9. Перейти в режим Блоки.
10. Протестировать взаимодействие кнопки и надписи следующим образом:

Нажать на нажать на Кнопку1 в окне Блоки (Screen1). Выбрать коричневый обработчик события нажатия на кнопку «когда Кнопка1.Щелчок» и перетащить его в окно Просмотр.

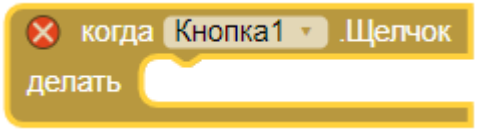


Рисунок 104

11. Выделить в разделе Screen1 компоненту Надпись1.
12. Из меню блоков надписи перетащить внутрь обработчика Кнопки1 сеттер «присвоить Надпись1.текст в»:

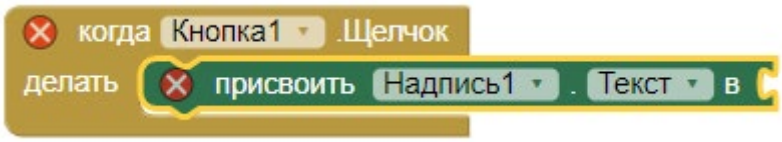


Рисунок 105

13. Перетащить из раздела Текст самый верхний блок и вставить его справа в блок-сеттер текста для Надписи1. Ввести в блок текст «Привет, IT-куб!».
14. Финальная комбинация блоков приобретёт следующий вид:

Рисунок 106. Итоговая схема блоков в обработчике нажатия на кнопку

15. Далее запустить эмулятор и проверить работу приложения.
16. Убедиться, что при нажатии на компоненту Кнопка1 в Надписи1 появляется текст «Привет, IT-куб!».
17. Приступить к реализации основного алгоритма, выводящего приветствие посимвольно в Надпись1 согласно интервалам таймера компоненты Часы1.

Работа программы представлена тремя основными группами блоков:

a. Блоки инициализации переменных.
b. Блоки обработки события нажатия на кнопку.
c. Блоки для организации работы таймера.

18. Необходимо инициализировать 3 глобальные переменные:

start — логическая переменная для контроля запуска таймера (изначально принимает значение Ложь);
text — содержит текст **«Привет, IT-куб!»;**
i — счётчик символов в тексте **«Привет, IT-куб!».**

Для этого нужно использовать блок «инициализировать глобальную» из раздела Переменные в окне Блоки. Вписать названия переменных в окошко сразу после слова «глобальную».

19. Для переменной start перетащить второй блок из раздела Логика. В данном случае Ложь означает запрет для включения таймера, пока эта переменная не станет равной значению Истина.
20. Для переменной text перетащить первый блок из раздела Текст и вписать значение «Привет, IT-куб!». Это сообщение и будет выводиться посимвольно в Надписи1.
21. Переменная i будет использоваться как счётчик и при каждом такте таймера увеличиваться на 1, отхватывая следующий символ из переменной text. Для инициализации i перетащить первый блок из раздела Математика.
22. Финальная комбинация блоков инициализации приобретёт следующий вид:

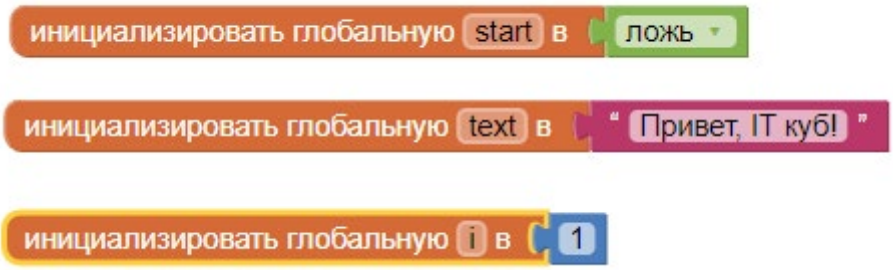


Рисунок 107. Блоки инициализации

23. Поменять содержимое обработчика Кнопки1. Для этого можно с помощью ЛКМ перетащить в изображение корзины и утилизировать:

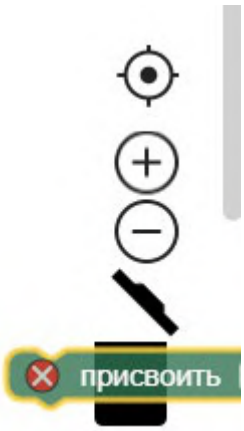


Рисунок 108. Корзина для удаления блоков

Либо временно перенести в пустое место окна Просмотр, так как этот блок пригодится в дальнейшем.

24. В пустой обработчик Кнопки1 добавить из раздела Переменные сеттер для переменной start и установить значение Истина.

Рисунок 109. Инициализация переменной в обработчике кнопки

Это означает, что при нажатии на Кнопку1 глобальная переменная start принимает значение Истина (что далее влечёт запуск Таймера).

25. Добавить событие запуска Таймера у компоненты Часы1. В разделе Screen1 нажать ЛКМ на компоненте Часы1 и из меню перетащить в окно Просмотр блок события «когда Часы1.Таймер»:

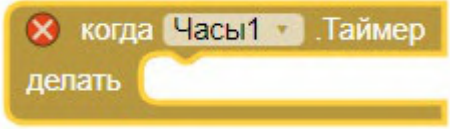


Рисунок 110. Обработчик интервалов таймера Часов

26. Добавить проверку глобальной переменной start. Если переменная start Истина, значит, согласно интервалу Таймера (т.е. с тактом в 1 секунду), будет выполняться содержимое условного блока «если ... то». В данном случае это команда на присвоение свойству Текст компоненты Надпись1 некоторого значения:

Рисунок 111. Начальная схема блоков обработчика таймера

27. В качестве текста должны использоваться символы из выражения «Привет, IT-куб!», хранящегося в переменной text. Перетащить из раздела Текст блок «сегмент текста».

28. Установить для первого параметра (аргумента) блока значение переменной text. Для этого из раздела Переменные перетащить блок-геттер «получить». Вставить как первый параметр в блок «сегмент текста» и выбрать в раскрывающемся списке переменную text.

Рисунок 112. Блок получения сегментов текста

29. Установить в качестве начала текста вторым параметром блока индекс (порядковый номер) символа. Каждую секунду (так как интервал таймера установлен равным 1 секунде) индекс будет увеличиваться на единицу за счёт увеличения переменной i. За счёт этого каждый раз будет выбираться последующий символ из текста.

30. Установить в качестве параметра длина первый блок из раздела Математика. В блоке указать цифру 1 (поскольку требуется вывод ровно одного символа в Надпись1).

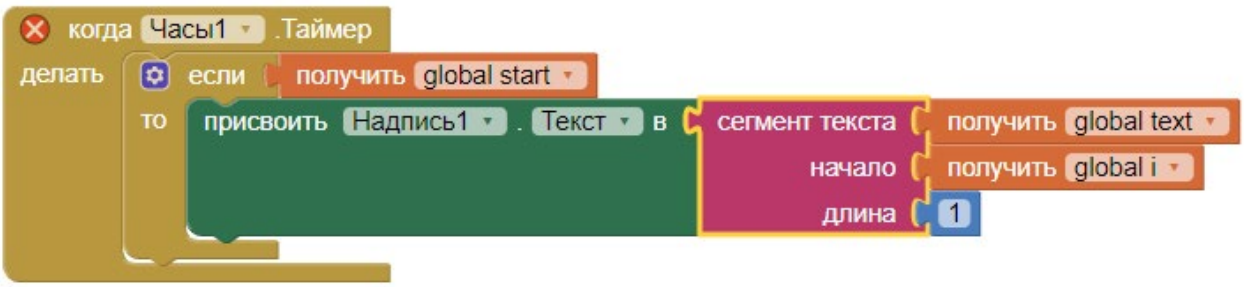


Рисунок 113. Промежуточный вид 1 схемы блоков обработчика таймера

31. Добавить сеттер для переменной i, увеличивающий её на единицу (чтобы менялся порядковый номер извлекаемого символа). Для этого из раздела Переменные перетащить сеттер «присвоить», указать в нём переменную i и прикрепить его снизу к зелёному сеттеру «присвоить» Надписи1.

32. Чтобы он заработал, дополнительно необходимо применить математическую операцию суммирования «+» (перетащить из раздела Математика блок «+» и прикрепить справа к сеттеру переменной i). В качестве параметров блока суммирования указать саму же переменную i и цифру 1:

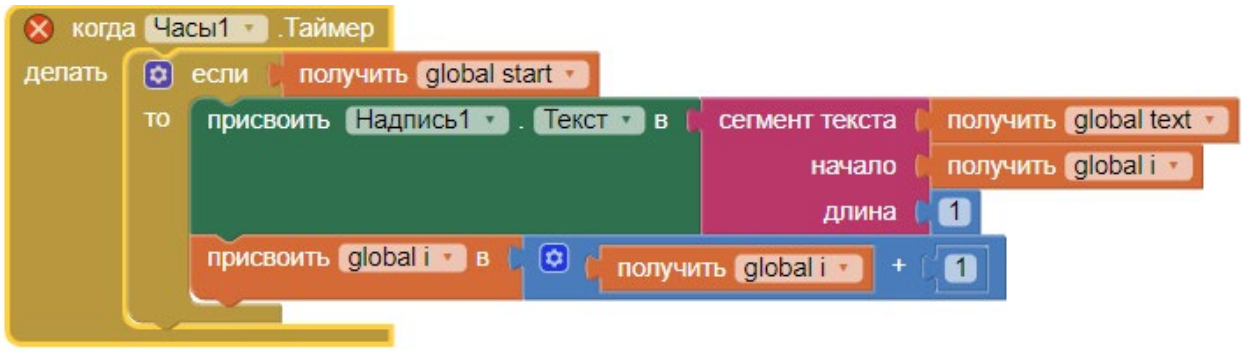


Рисунок 114. Промежуточный вид 2 схемы блоков обработчика таймера

33. Учесть, что текущий индекс i для извлекаемого из текста символа не может превышать длины текста «Привет, IT-куб!». Иначе произойдёт сбой в работе приложения. Значит, должны соблюдаться два условия для работы таймера:

Переменная start должна принимать значение Истина.

Переменная i не должна принимать значение больше длины текста «Привет, IT-куб!».

Тогда вместо прежнего условия, что глобальная переменная start принимает значение Истина, появляются два условия, объединённые блоком «логическое И».

Рисунок 115.

34. Для этого временно перетащить ЛКМ оранжевый геттер «получить global start» в пустое место окна Просмотр и вместо него из блока Логика добавить блок условия «Логическое И».

35. В качестве первого параметра блока «Логическое И» установить временно отложенный в сторону геттер переменной start.

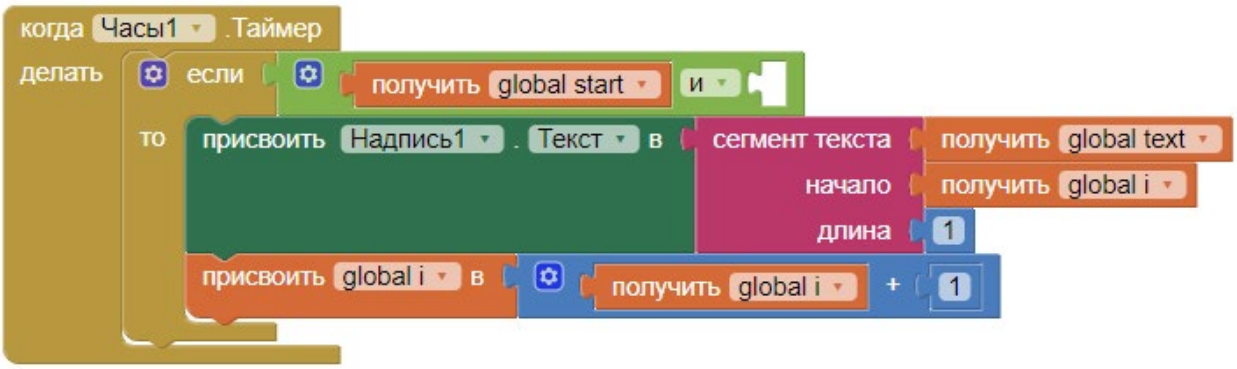


Рисунок 116. Промежуточный вид 3 схемы блоков обработчика таймера

36. В качестве второго параметра блока «Логическое И» вставить из раздела Математик выражение «меньше или равно»:

Рисунок 117.

37. Далее в данное математическое выражение необходимо вставить проверку, что переменная i меньше или равна длине текста text. Для этого в левую часть блока «меньше или равно» перетащить геттер переменной i из раздела «Переменные».

38. В правую часть блока вставить значение длины текста «Привет, IT-куб!». Для этого перетащить блок «Длина» из раздела Текст и прикрепить к нему справа геттер переменной text.

Рисунок 118.

39. Финальная комбинация блоков работы Таймера приобретёт следующий вид:

Рисунок 119. Финальная схема блоков обработчика таймера

40. Запустить эмулятор и проверить работу приложения. Текст выводится в Надпись1 при каждом такте Таймера ТОЛЬКО по одному символу друг за другом. После вывода последнего символа обработка событий Таймера прекращается и в надписи остаётся последний символ — «!».

Дополнительные задания: *необходимо, чтобы в Надпись1 выводилось не по одному символу, а при каждом такте Таймера добавлялся дополнительный символ справа, например:*

Первый такт: ‘П’
Второй такт: ‘Пр’
Третий такт: ‘При’
Четвертый такт: ‘Прив’
Пятый такт: ‘Привет’
И т. д.

Подсказка: для этого достаточно использовать код приложения из лабораторной работы и заменить в блоке

аргументы (параметры) «начало» на «длина» и наоборот:

В этом случае текст будет выбираться каждый раз (при каждом такте таймера) начиная именно с буквы «П» первого символа выражения «Привет, IT-куб!». Длина выводимого текста каждый раз будет увеличиваться на 1, так как значение глобальной переменной i после каждого такта увеличивается на 1.

41. Проверить работу нового варианта приложения в эмуляторе.

42. В данной реализации весь текст выводится посимвольно только один раз. При следующем нажатии на Кнопку1 ничего не происходит. Чтобы текст начал выводиться заново, добавить в обработчик Кнопки1 сеттер, сбрасывающий значение переменной i до 1.

Рисунок 120. Схема блоков сброса значения переменной i до 1

43. Проверить работу нового варианта приложения в эмуляторе.

Выводы: в данной работе рассматривалась работа с компонентами интерфейса пользователя совместно с базовыми математическими и логическими блоками.

Контрольные вопросы:

Для чего нужен таймер компоненты Часы?

Как увеличить переменную на 1?

Что делает обработчик события нажатия на Кнопку?

Лабораторная работа 3. Калькулятор

Теоретическая часть

В данной работе в том числе используется теоретический материал из дидактических материалов.

Практическая часть

Цель работы: создать простой калькулятор. Ввод цифр осуществлять в текстовых полях.

Ход работы

1. Перейти по адресу http://ai2.appinventor.mit.edu/ и запустить среду АИ (при необходимости авторизоваться на сайте «App Inventor»).

2. Через пункт главного меню «Проекты» -> «Начать новый проект» создать новый проект под названием Calculator. Перенести компоненты из раздела **интерфейс пользователя** на экран приложения (Screen1) согласно следующей схеме, где a, b, c — текстовые поля, a plus, minus, mult, div — кнопки:

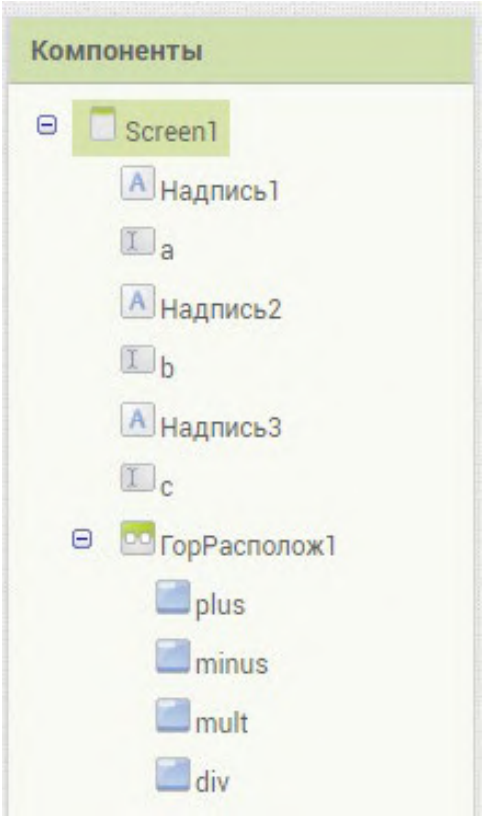


Рисунок 121. Схема компонент

3. У главного экрана Screen1 установить свойства:
   Заголовок — Calculator
4. У компоненты ГорРасполож1 установить свойство:
   Ширина — Наполнить родительский
5. У всех кнопок в горизонтальном расположении установить свойство:
   Ширина — Наполнить родительский
6. У компоненты Надпись1 установить свойство:
   Текст — «Введите первое число»
7. У компоненты Надпись2 установить свойство:
   Текст — «Введите второе число»
8. У компоненты Надпись3 установить свойство:
   Текст — «Результат»
9. У всех надписей и текстовых полей установить свойство:
   a. РазмерШрифта — 22
   b. У всех текстовых полей установить свойство:
   c. ТолькоЦифры — отметить галочкой
10. Убедиться, что финальная схема компонент соответствует следующему виду:

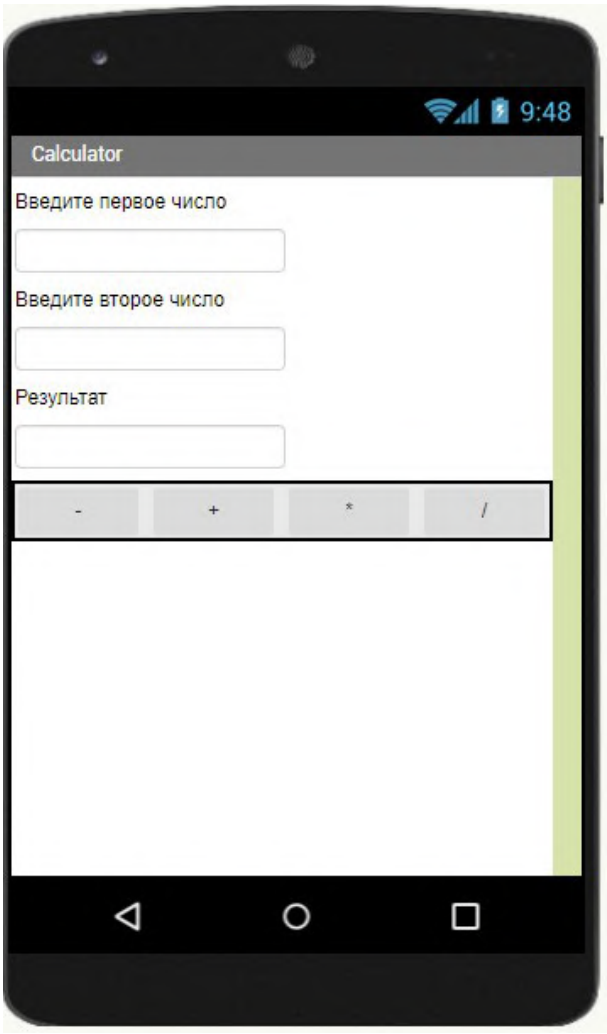


Рисунок 122. Примерный дизайн приложения Калькулятор

11. Перейти в раздел «Блоки».

В режиме Блоки логика работы приложения должна быть представлена двумя основными группами блоков:

блоки инициализации переменных (в которых содержатся значения введённых в текстовые поля чисел);

блоки обработки событий нажатия на кнопки операций (сложение, вычитание и т. д.).

12. Добавить инициализацию трёх глобальных переменных: a, b. Для этого перетащить блок «инициализировать глобальную» из раздела Переменные и первый блок из раздела Математика. В блоки инициализации вписать имена переменных.

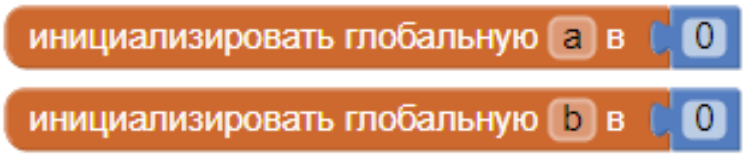


Рисунок 123. Инициализация переменных

13. Добавить обработчики события нажатия для каждой кнопки. Алгоритм обработки стандартный для всех кнопок: заполнить переменные данными из полей ввода a и b.

Рисунок 124. Инициализация переменных значениями текстовых полей

14. Затем с помощью блоков из раздела Математика добавить нужную математическую операцию над числами:

Рисунок 125. Блок суммирования

15. В конце алгоритма обработки полученный результат вывести в текстовое поле c:

Рисунок 126. Сложение переменных a и b

16. Повторить схему блоков для всех обработчиков кнопок с учётом типа математической операции. Для этого скопировать содержимое обработчика кнопки plus:

Рисунок 127. Итоговый вид блоков обработчика кнопки операции сложения

и вставить его в другие обработчики. Необходимо сменить блок суммирования на другие блоки из раздела Математика: вычитание, умножение и деление — и заново вставить в них в качестве параметров глобальные переменные.

17. Финальная схема блоков приложения приобретёт следующий вид:

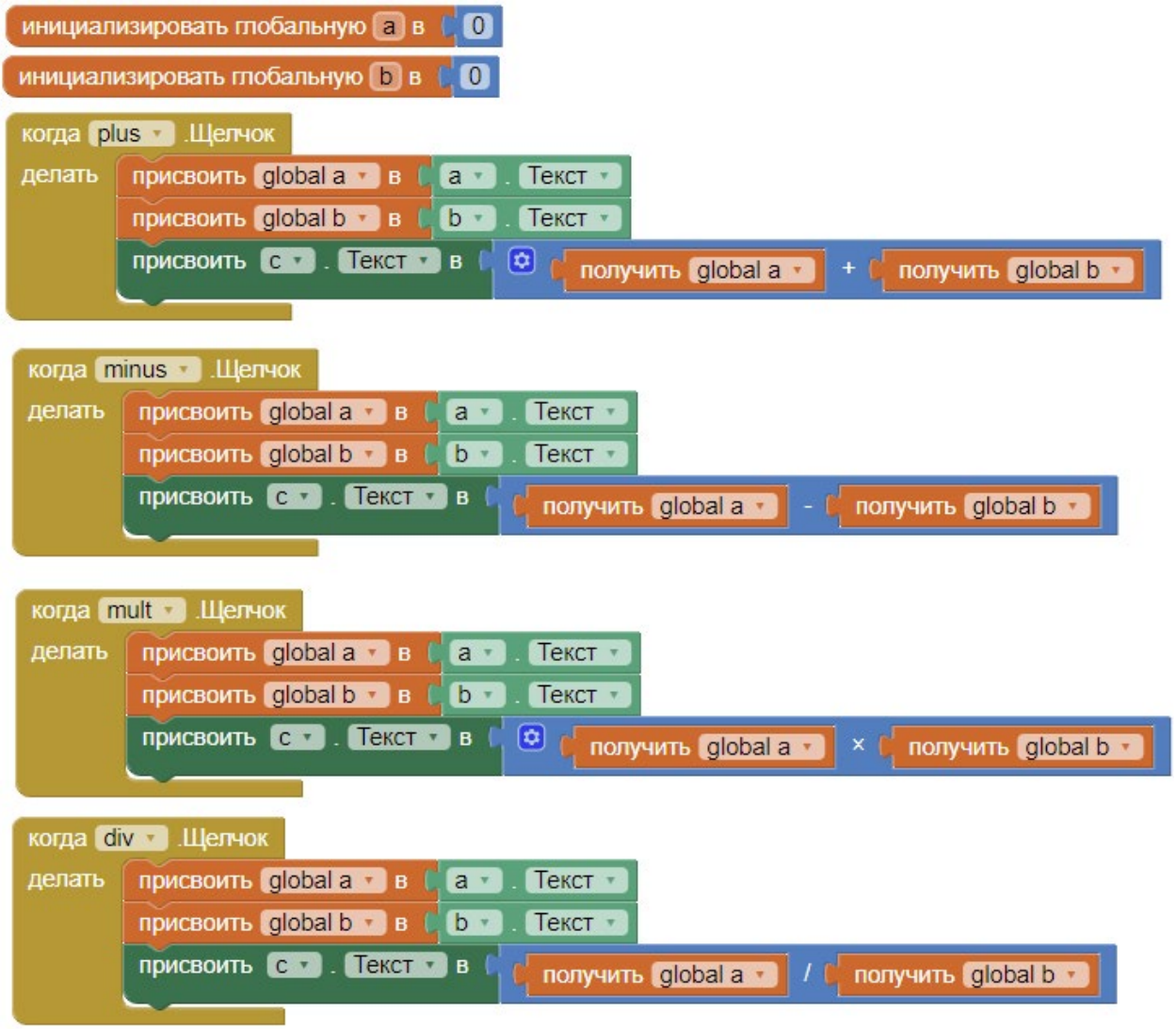


Рисунок 128. Финальная схема блоков приложения

18. Запустить эмулятор и проверить работу приложения.

Дополнительные задания:

1. Заменить текстовое поле  на надпись, чтобы результат выводился в надпись, а не в текстовое поле.
2. Добавить операции извлечения квадратного корня, sin, cos.
3. Перенести логику применения математических операций в отдельную процедуру. Для этого из раздела Процедуры перетащить блок «в procedure результат», так как необходима процедура, которая вычисляет именно результат, а не выполняет просто набор команд. В противном случае было бы достаточно перетащить блок «в процедура выполнить».

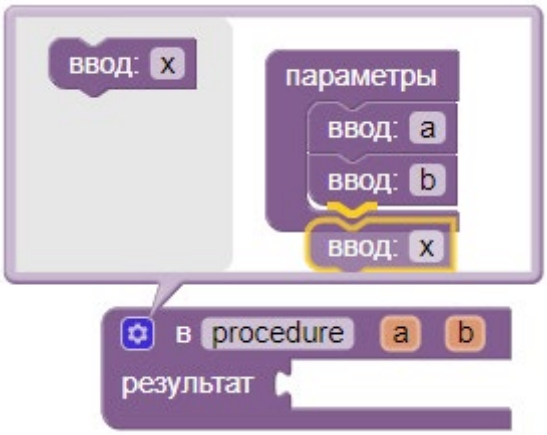


Рисунок 129. Работа с процедурой

4. Переименовать процедуру в Proc1.
5. С помощью мутатора добавить три параметра процедуры: a, b для чисел из полей ввода и operation для типа операции.
6. Создать следующую конструкцию из блоков:

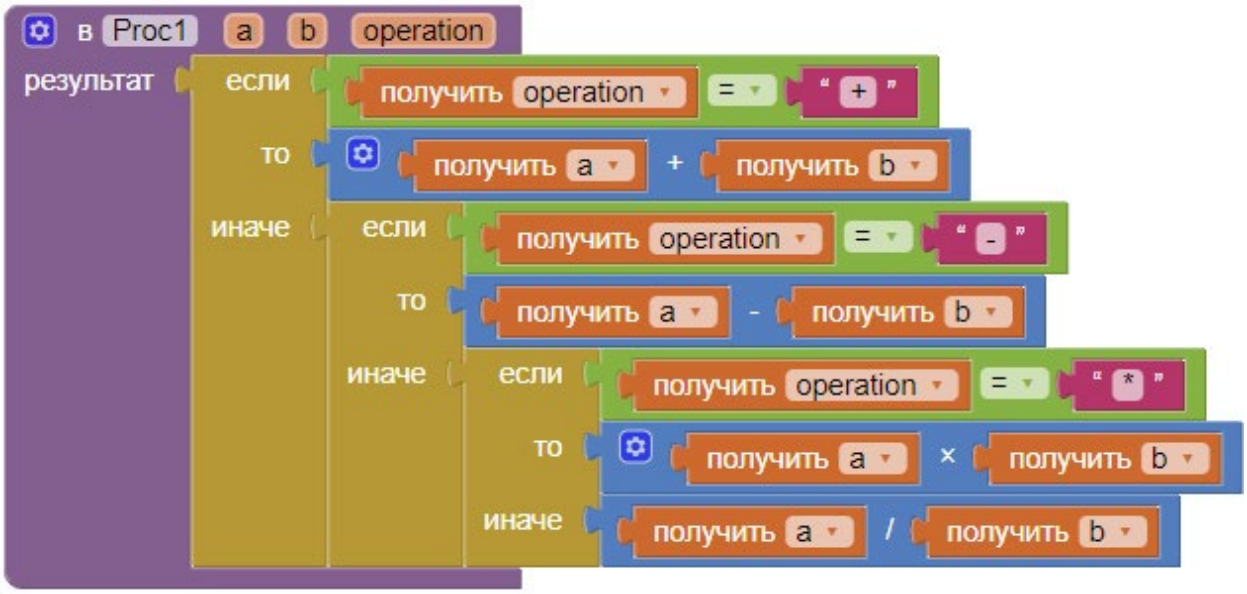


Рисунок 130. Представление операций калькулятора через процедуру

7. Модифицировать все обработчики кнопок подобным образом:

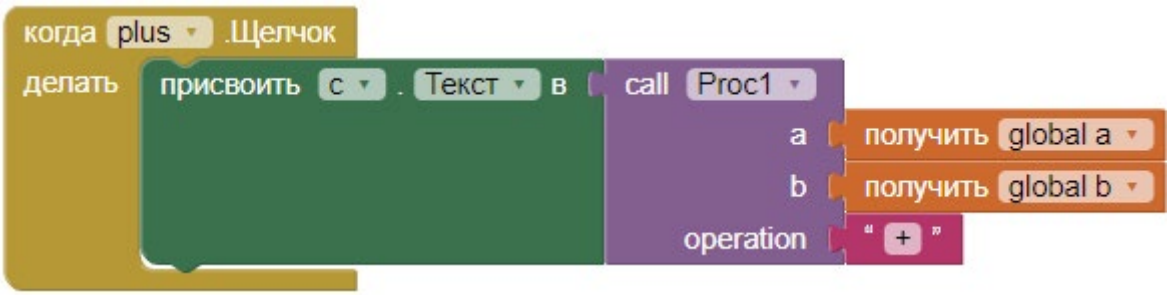


Рисунок 131. Замена прямого присвоения значения в поле «с» вызовом процедуры с указанием типа операции

Тем самым логика вычислений выносится из обработчиков в процедуру Proc1. Обработчик нажатия кнопки больше не содержит прямых указаний, как осуществлять операцию (не содержит логики осуществления операции), и стало намного проще копировать блоки для создания новых обработчиков. Теперь, чтобы произвести вычисление в других обработчиках, достаточно:

- скопировать все блоки внутри обработчика кнопки plus;
- вставить их в другой обработчик;
- сменить знак операции («+», «—», «*», «/») в параметре operation.

Выводы: с помощью процедур можно вынести логику работы программы в отдельные модули, что упрощает дальнейшие модификации кода.

Контрольные вопросы:

Что собой представляют переменные?

Какие базовые арифметические операции доступны в AppInventor?

Какие блоки используются для присваивания значений переменным?

Лабораторная работа 4. Кнопочный калькулятор

Теоретическая часть

В данной работе в том числе используется теоретический материал из дидактических материалов.

Блоки из раздела «Любой компонент» очень удобны для обработки событий множества одинаковых компонент. Например, если в приложении имеются 10 кнопок и процедура обработки события нажатия на кнопки одинаковая. При наличии ещё 5 кнопок, которые должны обрабатываться по-другому, это можно учесть в обработчике «любого компонента» посредством блока «если... то... иначе... то...».

Например, следующая комбинация блоков создаёт обработчик для нажатий на ЛЮБУЮ кнопку экрана. При нажатии на любую кнопку экрана вызывается компонента Уведомитель1 и выводится сообщение с текстом именно нажатой Кнопки:

Рисунок 132. Обработчик нажатия на любую кнопку экрана

Для этого нужен Зелёный блок-геттер для свойства Текст кнопки, который находится там же, в разделе «Любой компонент». Чтобы выводился текст именно ЛЮБОЙ нажатой кнопки, необходим оранжевый геттер «получить компонент» для переменной «компонент» из блока «when any Кнопка.Щелчок».

Оранжевый геттер переменной «получить компонент» добывается наведением ЛКМ на оранжевое слово «компонент» под словом «when». Далее следует перетащить ЛКМ геттер в нужное место.

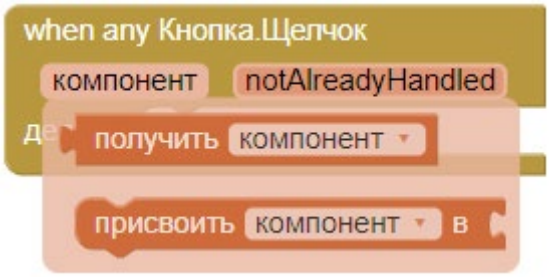


Рисунок 133. Получение геттера для внутреннего параметра блока

Вообще, зелёный геттер может получать в качестве параметра и конкретные компоненты тоже, например конкретную Кнопку1. В этом случае при нажатии любой кнопки будет выводиться только текст Кнопки1.

Блоки работают одинаково для всех кнопок приложения. Тем самым отпадает необходимость для каждой кнопки писать одинаковый код. В данной работе блоки применяются для считывания значения цифры из кнопки.

Также в данной лабораторной работе в режиме Блоки происходит знакомство с такой структурой данных, как списки. **Не путать с компонентой Список в режиме Дизайнер.** Структура данных типа список представляет собой обычный массив, поэтому в дальнейшем они будут обозначаться в том числе как массивы. Списки находятся в разделе Массивы встроенных блоков.

В данном курсе используются следующие блоки для работы с массивами:

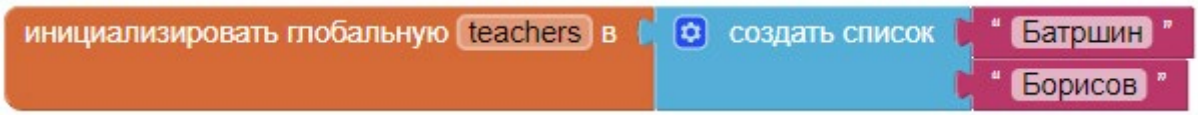


Рисунок 134. Блок для создания списка с уже определёнными значениями

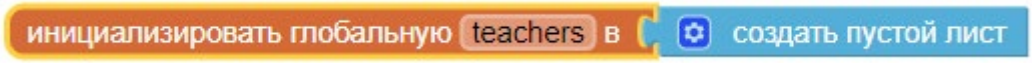


Рисунок 135. Блок для создания пустого списка

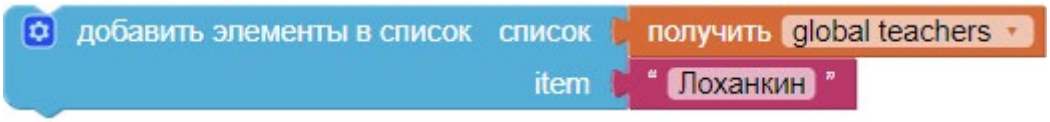


Рисунок 136. Блок для добавления нового элемента в список

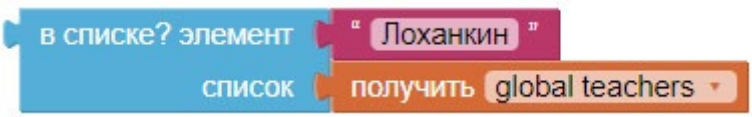


Рисунок 137. Блок, проверяющий, содержится ли элемент в списке

Блок для проверки *содержится ли элемент в списке* возвращает значение типа логической переменной и может использоваться в логических выражениях, например:

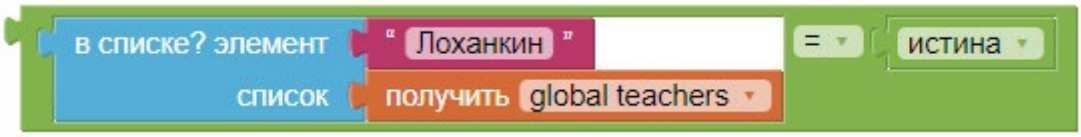


Рисунок 138. Логическое выражение с блоком проверки

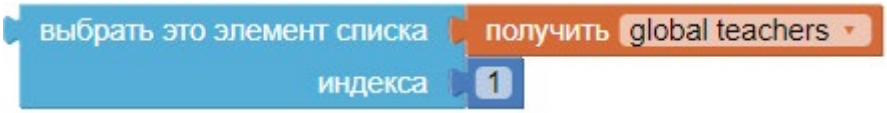


Рисунок 139. Блок для выбора элемента из списка по индексу

Полезными могут оказаться также следующие блоки:

Рисунок 140. Блок для получения количества элементов в списке

Рисунок 141. Блок для выбора случайного элемента из списка

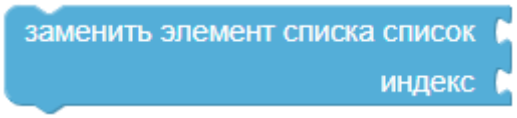


Рисунок 142. Блок для удаления элемента из списка по индексу

Практическая часть

Цель работы: изучить работу с компонентами (контейнерами) раздела Расположение; познакомиться с разделом «Любой компонент» на примере целочисленного кнопочного калькулятора с операциями сложения и вычитания.

Ход работы

1. Перейти по адресу http://ai2.appinventor.mit.edu/ и запустить среду АИ (при необходимости авторизоваться на сайте «App Inventor»).
2. Через пункт главного меню «Проекты» -> «Начать новый проект ...» создать новый проект под названием Calc.

В ходе проекта должен быть создан калькулятор, состоящий из кнопок с цифрами, операциями, знаком «=» и полем ввода.

Калькулятор должен выполнять только одну операцию за один раз (например, «1+3», «100 – 200»).

При нажатии на кнопки с цифрами в пустом поле ввода (или если поле ввода содержит только 0) должно появиться новое число, если же поле ввода уже содержит цифру, то новая цифра должна дописываться справа.

При нажатии на кнопку со знаком операции («+», «–») запоминается сама операция и число в поле ввода.

При нажатии на кнопку «=» происходит операция вычисления результата.

3. Для главной компоненты экран (Screen1) установить свойства:
Заголовок — «CALC»
4. Добавить на экран следующие компоненты (здесь и далее имена компонент сокращены: «ГоризонтальноеРасположение1» — «ГорРасполож1» и «ВертикальноеРасположение1» — «ВертРасполож1» по причине узости окна Компоненты и для удобства при работе в режиме Дизайнер):

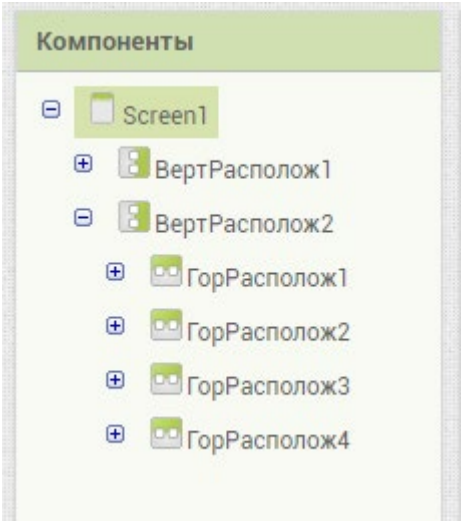


Рисунок 143. Общая схема компонент экрана

5. Установить для всех компонент свойства:
   a. Ширина — Наполнить родительский
   b. Высота — Наполнить родительский
6. В компоненту ВертРасполож1 добавить компоненты:
Надпись1 и установить свойства:
   a. Высота, Ширина — Наполнить родительский
   b.Текст — Результат
   Текст 1(переименовать в ПолеВвода) и установить свойство:
   Только цифры — отметить галочкой (чтобы нельзя было случайно ввести в текстовое поле посторонние символы).
   Компоненту Горизонтальное расположение (переименовать в ГорРасполож5) и установить свойство:
Ширина — Наполнить родительский.

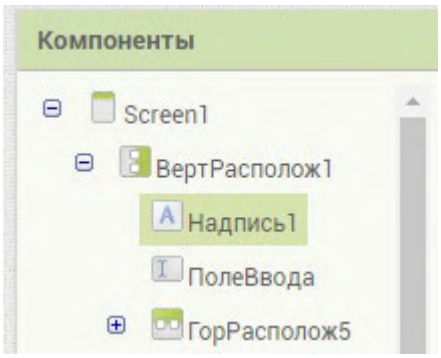


Рисунок 144. Примерная схема компонент контейнера ВертРасполож1

7. В компоненту ГорРасполож5 добавить 5 компонент типа Кнопка и переименовать соответственно в:

plus, minus, mult, div, eq

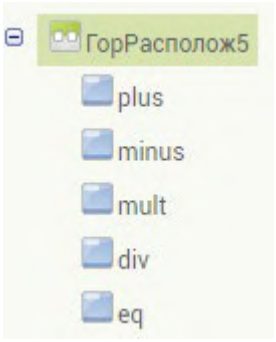


Рисунок 145. Схема именования кнопок операций в контейнере ГорРасполож5

8. Для данных кнопок установить следующие значения свойства Текст: «+», «—», «*», «/», «=».

9. Остальные свойства не менять, кроме кнопки eq, у которой установить свойство:

Ширина — Наполнить родительский

10. Далее компоненты ГорРасполож1, ГорРасполож2, ГорРасполож3, ГорРасполож4 заполнить компонентами типа Кнопка, как на следующем рисунке. Кнопки переименовать в b1, b2 и т.д. У всех кнопок установить свойства:

Ширина, Высота — Наполнить родительский.

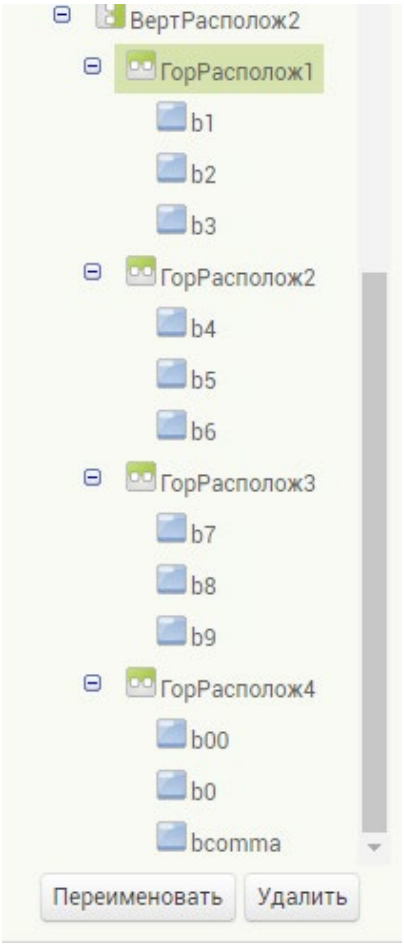


Рисунок 146. Схема компонент в контейнере ВертРасполож2

11. Убедиться, что финальная схема компонент соответствует следующему виду:

Рисунок 147. Примерная схема дизайна приложения

12. Перейти в режим Блоки.
13. В режиме Блоки логика работы приложения представлена тремя основными группами блоков:
    a. Инициализация.
    b. Обработка кнопок с цифрами и операциями.
    c. Обработка кнопки со знаком «=» (компонент eq).

    С точки зрения упрощения кода следовало бы разделить на отдельные блоки работу с цифрами и операциями, но тогда пришлось бы отказаться от колоссальных преимуществ блоков раздела «Любой компонент».
14. Инициализировать следующие глобальные переменные:
    переменная a для хранения левого операнда операции
    переменная sign для хранения знака операции
    переменная spisok для идентификации цифр на кнопках

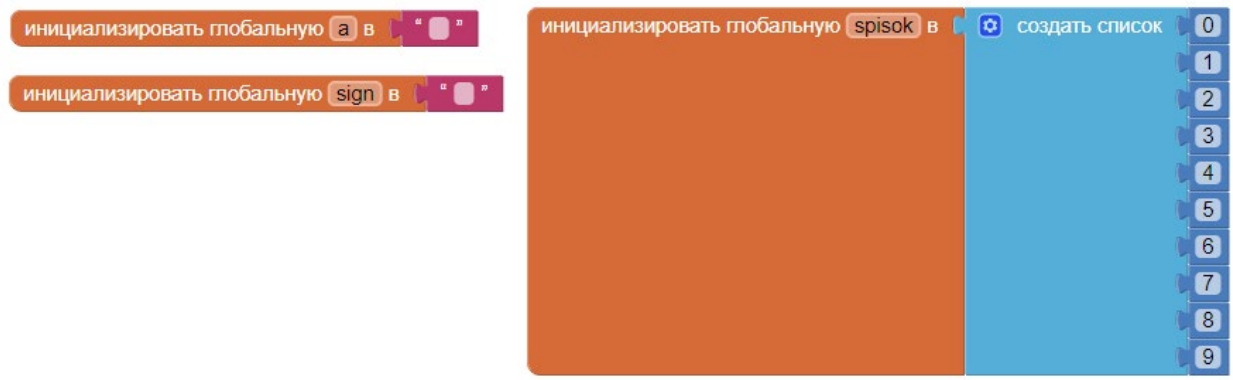


Рисунок 148. Инициализация массива spisok и переменных a, sign

Для идентификации кнопок с операциями глобальная переменная не используется, так как в данной работе представлены только 2 операции (+ / –) и далее список создаётся «на лету» в блоке «when Any Кнопка.Щелчок»:

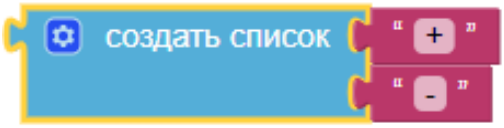


Рисунок 149. Список с элементами «+», «–»

15. Перетащить блок «when Any Кнопка.Щелчок».
16. Заполнить получившуюся конструкцию блоками для идентификации кнопок «+» или «–» и в случае успеха – реализации событий сложения и вычитания.

Для этого создать логическое условие для проверки, находится ли текст нажатой кнопки в создаваемом на лету списке значений операций («+», «–»).

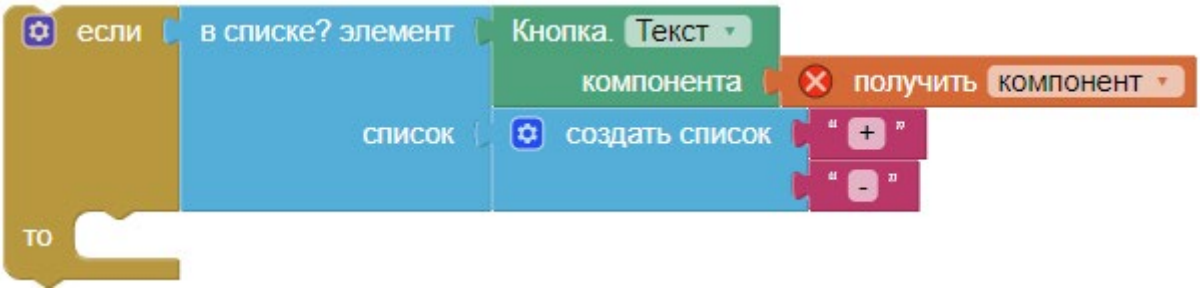


Рисунок 150. Проверка текста любой из нажатых кнопок

Если да, то переменной «a» присваивается текущее значение в поле ввода (например, цифра 100), затем в переменную sign записывается значение операции только что нажатой кнопки (например, «+»). После этого поле ввода очищается.

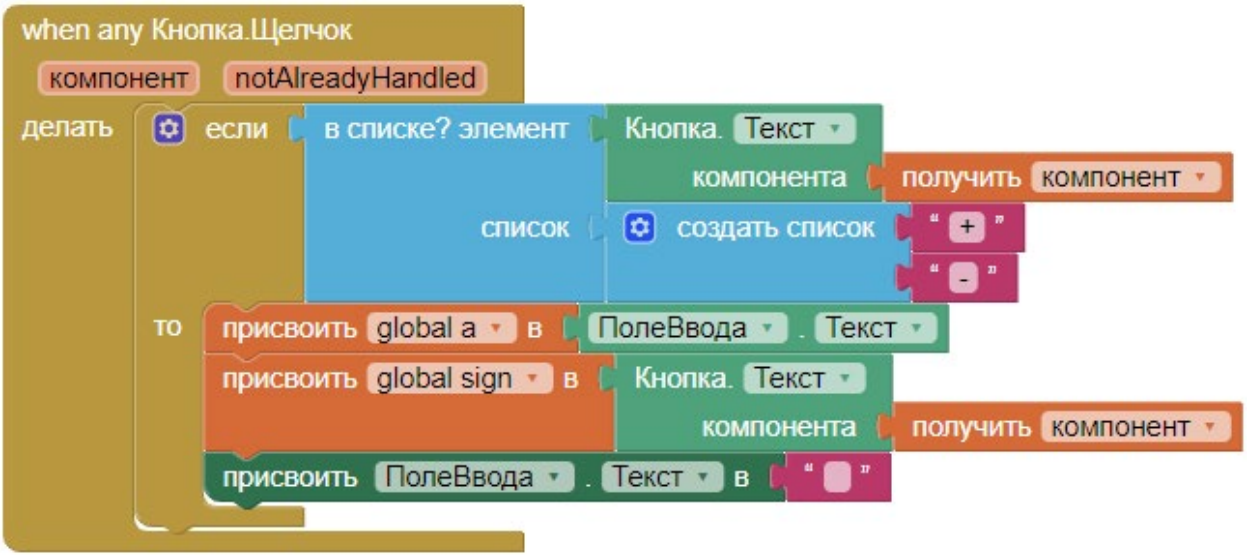
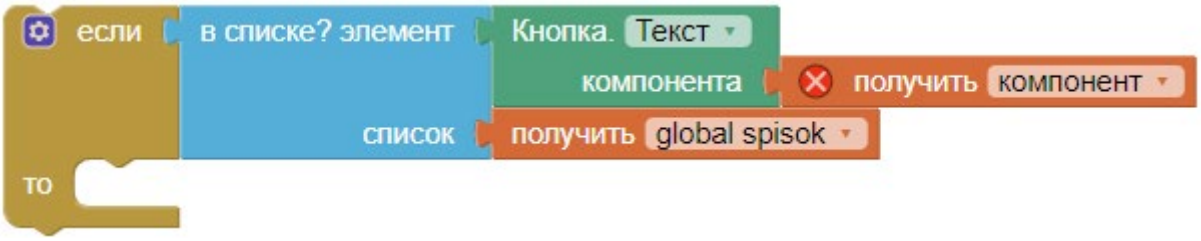


Рисунок 151. Итоговая схема обработчика нажатия на любую кнопку

17. Блоки событий на нажатие кнопок («+», «—») реализованы, далее необходимо реализовать блоки идентификации кнопок с цифрами и реализацию соответствующих событий.

18. Для этого создать логическое условие для проверки, находится ли текст нажатой кнопки в списке цифр глобальной переменной spisok, т.е. принимает ли значение из массива {1, 2, 3, 4, 5, 6, 7, 8, 9, 0}.



19. В случае успеха добавить новое условие, которое, в случае если в поле ввода содержится цифра 0, просто затирает её текущим значением кнопки (целое число не может начинаться с нуля). Если в поле ввода содержится другая цифра, то к ней дописывается значение нажатой кнопки. Например, если в поле отображалась цифра 1, пользователь нажал кнопку «2», то в поле ввода появляется число 12.

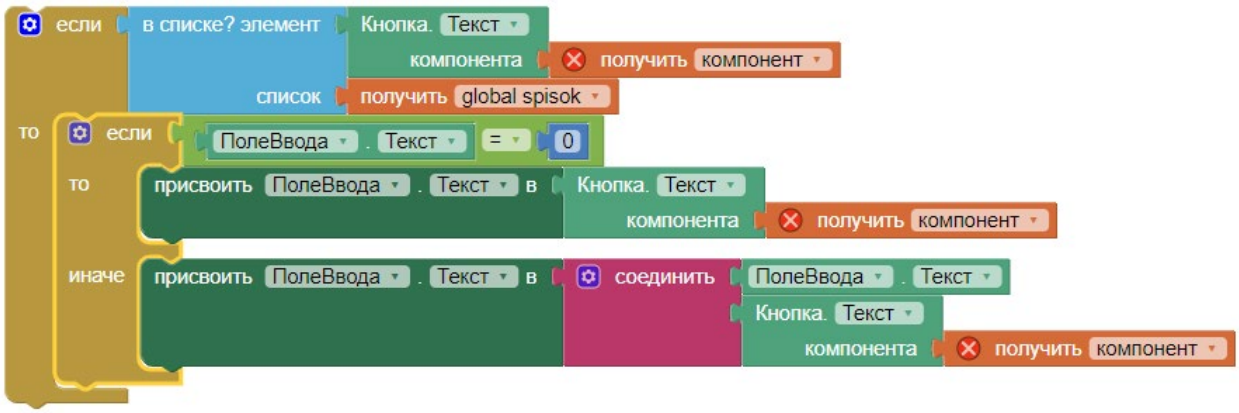


Рисунок 152. Блоки проверки корректности вводимых чисел

20. Необходимо добавить новую конструкцию в общий блок обработки нажатий любых кнопок:

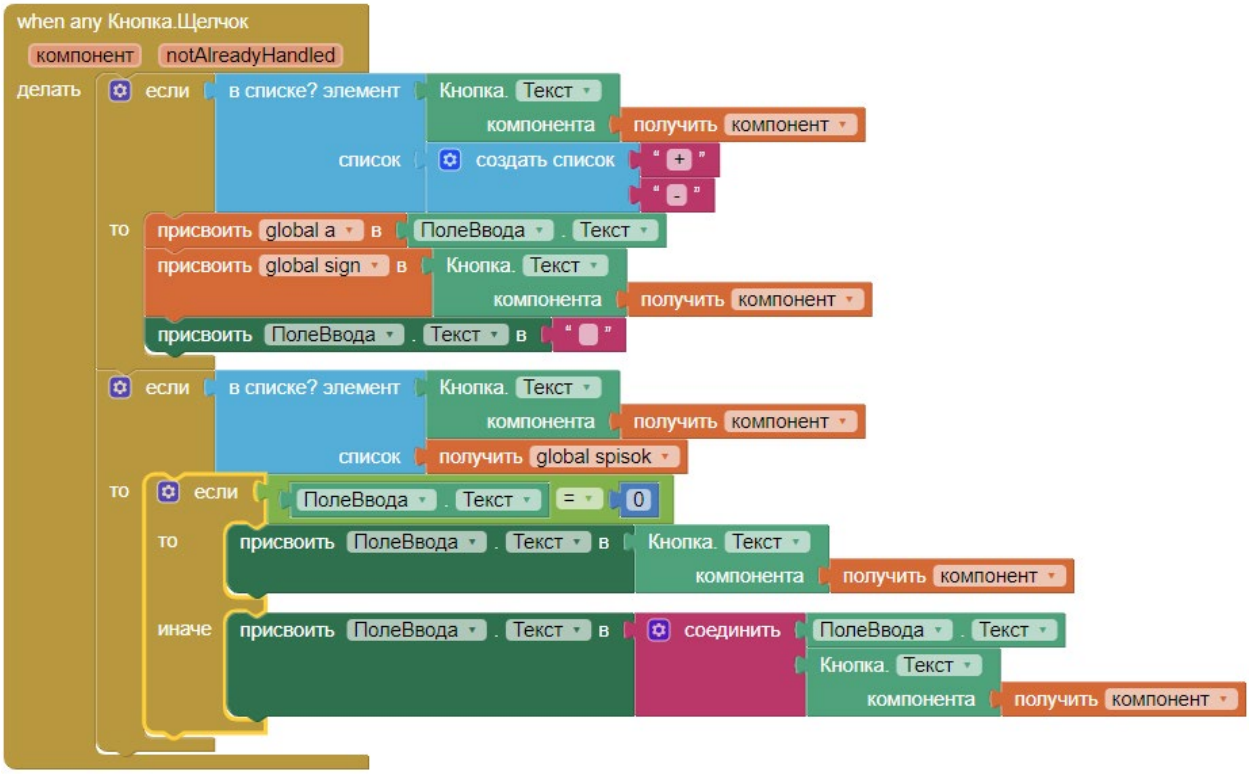


Рисунок 153. Итоговая схема блоков обработчика нажатий на любую кнопку экрана

21. Далее осталось реализовать обработку события нажатия на кнопку eq следующим образом:

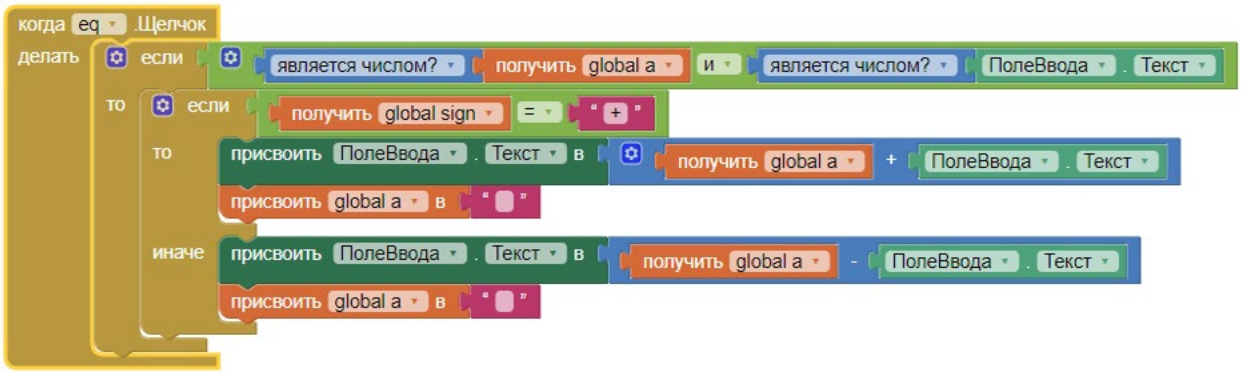


Рисунок 154. Схема блоков для кнопки «=»

Первое условие проверяет, числами ли являются переменная «a» и текущее значение поля ввода.

Второе условие проверяет, нажималась ли перед этим кнопка со знаком «+», тогда происходит сложение глобальной переменной «a» и текущего числа из поля ввода. Результат записывается в поле ввода. Иначе происходит аналогичное вычитание с выводом результата в поле ввода.

22. Финальная комбинация блоков приобретёт следующий вид:

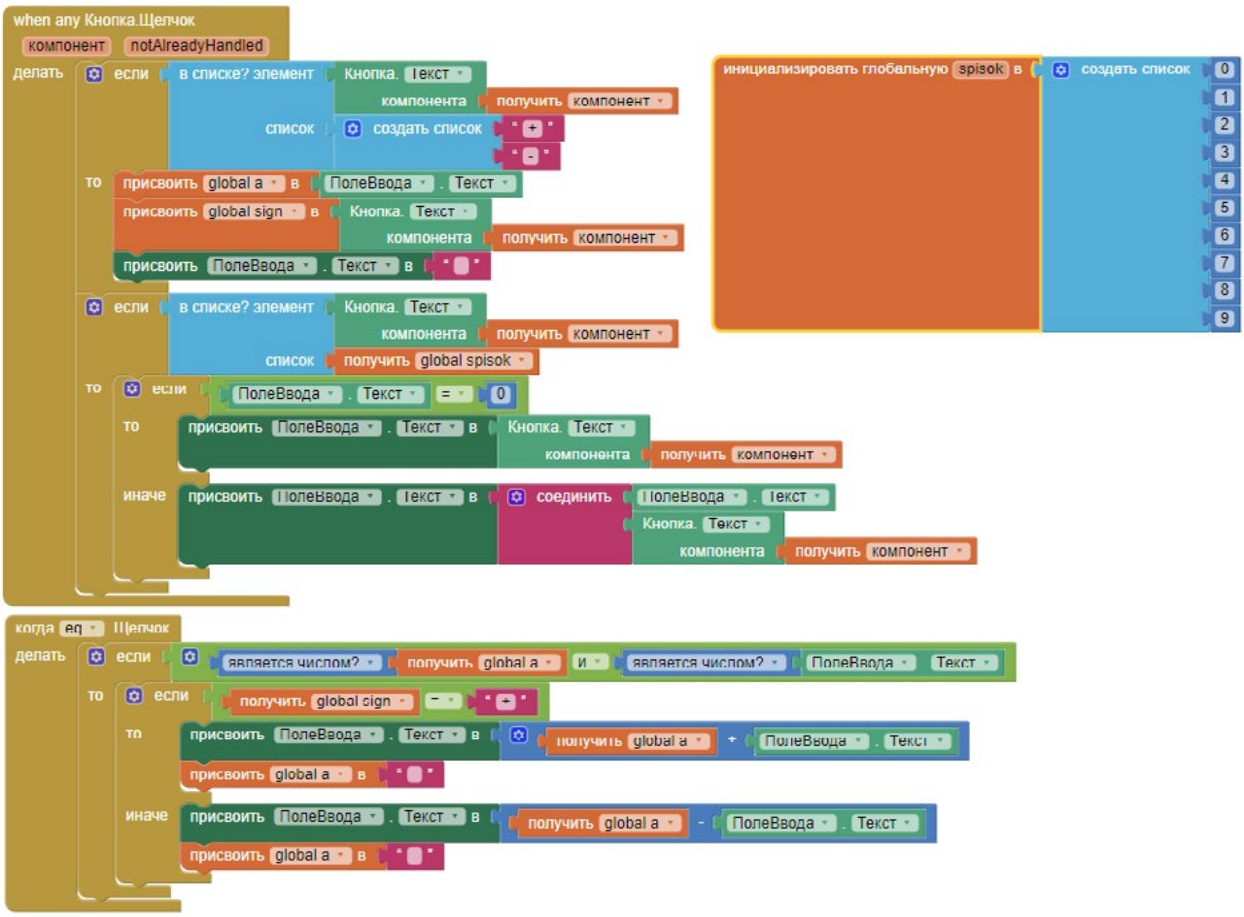


Рисунок 155. Финальная схема блоков кнопочного калькулятора

23. Запустить эмулятор и проверить работу программы.

Дополнительные задания:

1. Усовершенствовать дизайн: выравнивания, цвета, эффекты. На главном экране у свойства Theme выбрать значение Device Default.
2. Добавить возможность ввода вещественных чисел. Для этого усовершенствовать следующий блок (необходимо, чтобы при нажатии на знак запятой ноль не затирался).

Рисунок 156. Блок, который необходимо изменить

3. Добавить обработку кнопок «*», «/».
4. Добавить проверку, что если в случае нажатия на кнопку со знаком «=», глобальная переменная sign пуста (т.е. не нажималась кнопка с операцией, а значит, нет и второго операнда), то и значение переменной «a» следует обнулить.
5. Добавить проверку и учёт различных возможных ошибок обработки событий и работы с текстом, как с числом.

6. Выводить в Надпись1 значение переменной «а» после нажатия на любую кнопку со знаком операции.
7. Модифицировать калькулятор так, чтобы он вычислял не по одной операции за раз, а цепочки выражений, состоящие из множества операций. Например: 2+3 если 4*2.
8. Реализовать функционал кнопки «00», чтобы при нажатии на неё в поле ввода добавлялось сразу 2 ноля (если оно не пустое или не содержит только ноль).

Выводы: в данной работе создаётся приложение «Кнопочный калькулятор» и изучается раздел «Любые компоненты» режима Блоки.

Контрольные вопросы:

Для чего нужны блоки типа «Любой компонент»?

Зачем нужны оранжевые переменные на блоках типа «when any ...»?

Лабораторная работа 5.
Работа с компонентами интерфейса пользователя

Теоретическая часть

В данной работе в том числе используется теоретический материал из дидактических материалов.

Компонента Экран (Screen1) является визуальной компонентой верхнего уровня. На экране, как в контейнере, располагаются все остальные компоненты. Посредством свойств экрана можно:

- задать расположение остальных компонент на экране (ВыровнятьПоГоризонтали, ВыровнятьПоВертикали);
- указать иконку приложения (свойство Иконка);
- задать как цвет фона экрана, так и фоновый Рисунок экрана (свойства ЦветФона и ФоновыйРисунок);
- указать тему приложения (свойство Theme). Например, значение Device Default устанавливает тему типа Material Design;
- указать заголовок приложения (свойство Заголовок);
- указать автомасштабирование компонент экрана для различных экранов мобильных устройств (свойство Sizing со значением Responsive).

Компонента Список используется для представления данных в виде раскрытого вертикального списка. Расположена в разделе «Интерфейс пользователя». Основные свойства следующие:

Высота и Ширина, ЦветФона точно такие же, как у всех остальных компонент;

ЭлементыИзЦепочки — с помощью этого свойства через запятую задаются элементы списка (например: Понедельник, Вторник, Среда).

Рисунок 157. Примерный вид компоненты Список

Выбор означает текущий выбранный элемент списка. Обычно используется в режиме Блоки для определения того, какой элемент выбрал пользователь.

ПоказатьПанельФильтра — добавляется окно поиска элементов списка (используется, если список очень большой).

В режиме Блоки у компоненты Список имеется только один обработчик, который обрабатывает событие выбора элемента списка:

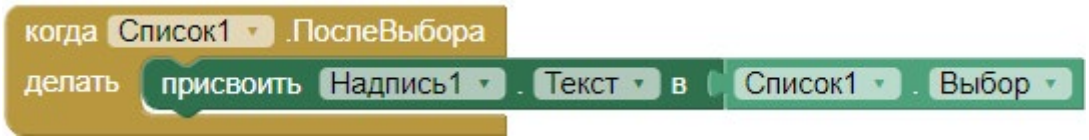


Рисунок 158

Зелёными геттерами и сеттерами задаются и считываются свойства, как у всех компонент.

При помощи сеттера «присвоить Список1.Элементы» компоненту типа Список можно инициализировать значениями массива:

Рисунок 159. Заполнение компоненты Список элементами массива

При помощи сеттера для свойства ЭлементыИзЦепочки компоненту типа Список можно инициализировать значениями слов, разделённых запятой:

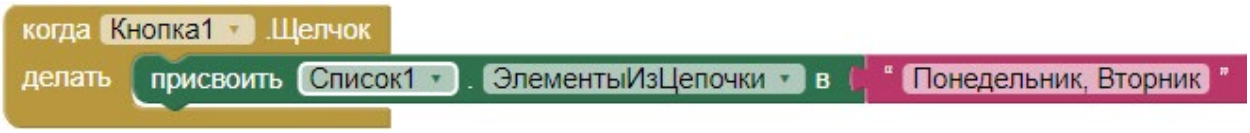


Рисунок 160. Заполнение компоненты Список текстом (с использованием знака запятой)

При помощи геттера «Список1.ИндексВыбора» можно получить индекс (порядковый номер) выбранного в списке элемента:

Рисунок 161. Вывод индекса выбранного элемента списка

Компоненты ВыборИзСписка и ИндикаторОжидания немного отличаются от компоненты Список, но в целом функционируют примерно так же. ВыборИзСписка имеет чуть более богатый набор свойств в режиме Дизайнер, позволяющий настраивать более изысканный вид компоненты. Также имеет дополнительные обработчики в режиме Блоки. Но в данной работе для компонент ВыборИзСписка и ИндикаторОжидания достаточно использовать те же блоки и те же свойства, что и у компоненты Список, а именно: обработчик вида «Список.ПослеВыбора» и свойства «ЭлементыИзЦепочки», «Выбор».

Компонента Switch (переключатель) работает как обычный переключатель или выключатель. Типовые свойства такие же, как и у остальных компонент (Высота, Ширина, РазмерШрифта и т.д.). Специфическим свойством является свойство «On» (Вкл.), которое обозначает, что переключатель включён. Имеет два состояния: «включено» (on) и «выключено», считывать которые можно геттерами:

Рисунок 162. *Г*еттер состояния «Вкл.» (On) переключателя

Отдельного состояния типа выключено нет, так как оно выражается через состояние «включено» (on):

Рисунок 163. Геттер состояния «Выкл.» переключателя

В режиме блоков имеется обработчик события изменения состояния компоненты:

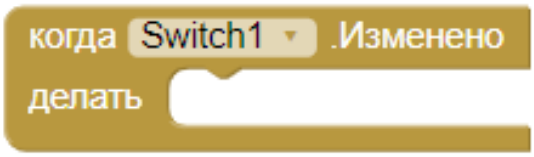


Рисунок 164

Посредством данного блока можно программировать действия, выполняемые при переключении компоненты.

Компонента ВыборщикИзображений позволяет работать с галереей мобильного устройства и осуществлять выбор нужных изображений. Содержит типовые процедуры и обработчики событий, сопутствующие выбору изображений.

Компонента ТекстВРечь используется для прочтения вслух заданного текста. Содержит одну основную процедуру вызова, при этом в качестве аргумента указывается текст сообщения:

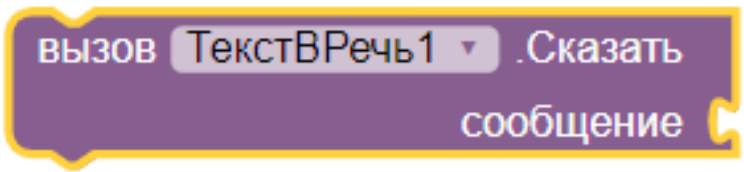


Рисунок 165. Процедура вызова компоненты ТекстВРечь

Язык озвучивания можно указать в свойствах в режиме Дизайнера либо программным путем. Также полезными могут оказаться обработчики событий до и после озвучивания текста, например:

Рисунок 166. Пред- и постобработчики компоненты ТекстВРечь

Практическая часть

Цель работы: освоить работу с некоторыми базовыми компонентами интерфейса приложения: несколькими видами списка, переключателем (Switch), выборщиком изображения; создать приложение, в котором пользователь, выбирая цвет из списка или меняя состояние переключателя, меняет фоновый цвет или изображение экрана.

Ход работы

1. Перейти по адресу http://ai2.appinventor.mit.edu/ и запустить среду АИ (при необходимости авторизоваться на сайте «App Inventor»).
2. Через пункт главного меню «Проекты» -> «Начать новый проект ...» создать новый проект под названием BackColor.
3. Посредством раздела Медиа загрузить файл с изображением, которое будет отображаться в качестве иконки приложения.
4. Настроить свойства экрана (Screen1) следующим образом:
   a. ВыровнятьПоГоризонтали - Центр;
   b. Заголовок — «BackColor»;
   c. Иконка — загруженный ранее файл
   d. Theme — Device Default
5. Добавить на экран компоненты: Надпись1, Switch1, ВыборИзСписка1, Список1, ВыборщикИзображений1.

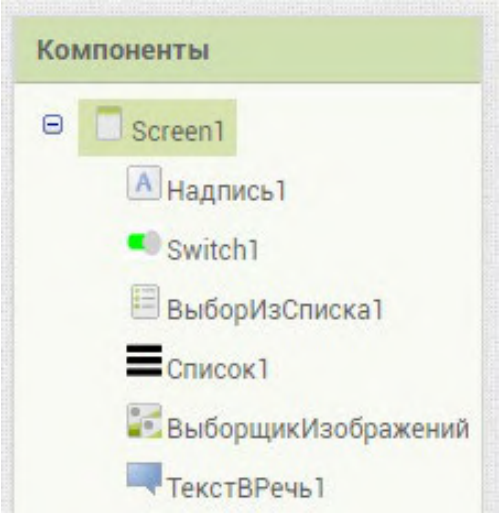


Рисунок 167. Схема компонент экрана

6. Настроить свойства компоненты Надпись1 следующим образом:
   a. текст — «Смена фона»
   b. размер шрифта — 24
7. Настроить свойства компоненты Switch1 следующим образом:
   a. текст — «ч/б»
   b. ЖирныйШрифт — отметить галочкой
8. Настроить свойства ВыборИзСписка1 следующим образом:
   a. ширина — наполнить родительский
   b. текст — «Выбор цвета фона»
   c. ItemBackgroundColor — белый
   d. ItemTextColor — чёрный
   e. элементы из цепочки — yellow, green
   f. ширина — 75 percent
9. Настроить свойства Список1 следующим образом:
   a. ЦветФона — белый
   b. ЦветТекста — чёрный
   c. ширина — 75 percent
10. Настроить свойства ВыборщикИзображений1:
    a. текст — «Выбор фона из галереи ...»
    b. ширина — 75 percent
11. Убедиться, что финальная схема компонент соответствует следующему виду:

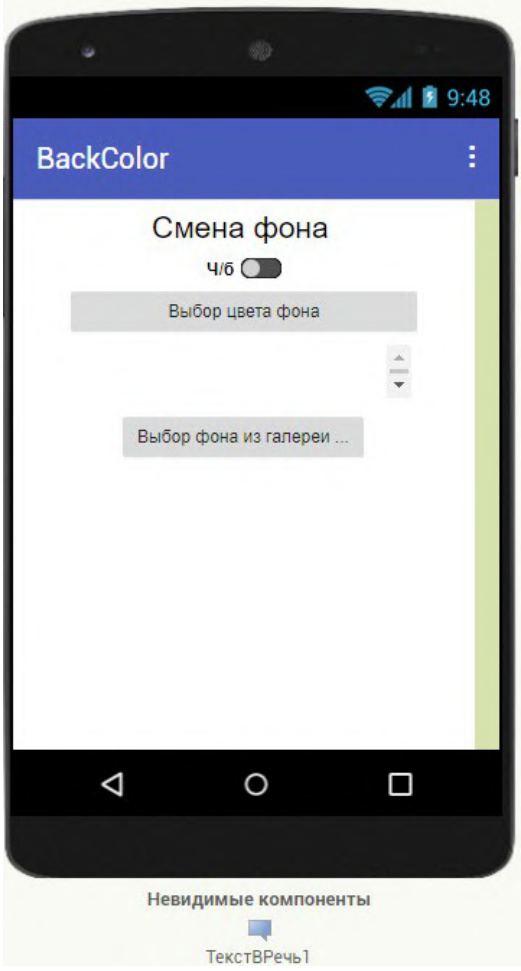


Рисунок 168. Примерный дизайн приложения

12. Перейти в раздел «Блоки».
13. Создать и проинициализировать глобальную переменную spisok, которая содержит цвета.

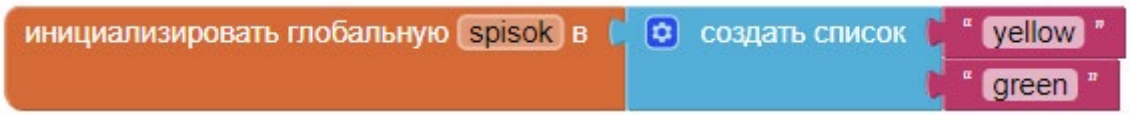


Рисунок 169. Инициализация переменной spisok

14. Инициализировать компоненту Список1 цветами глобальной переменной spisok:

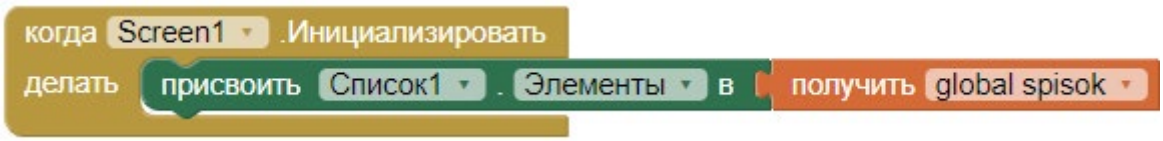


Рисунок 170. Инициализация компонент Список1 и ВыборИзСписка1

15. Создать процедуру ProcColors, которая в зависимости от значения переменной colorText (добавить её с помощью мутатора) возвращает нужный цвет:

Рисунок 171. Процедура ProcColors

Процедура состоит из блока вида «если... то... иначе» из раздела Управление. Блок возвращает то или иное значение в зависимости от условия.

Процедура позволяет не дублировать одинаковый код в двух местах, а именно при выборе элементов в компонентах: ВыборИзСписка1 и Список1.

16. Создать процедуру ProcSetBack, которая в зависимости от значения переменной x (добавить её с помощью мутатора) вызывает предыдущую процедуру ProcColors и устанавливает нужный цвет для фона экрана. Кроме того, эта процедура убирает фоновый Рисунок.

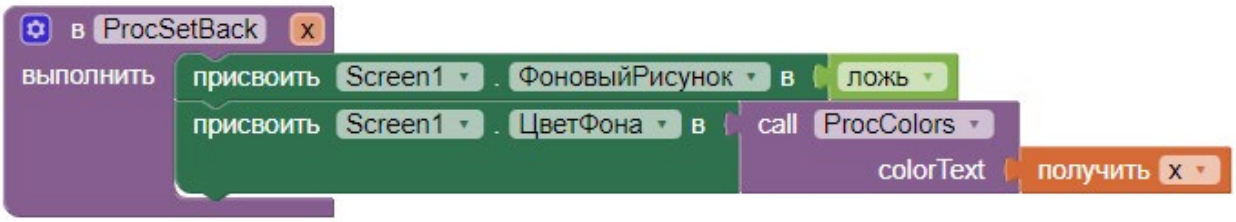


Рисунок 172. Процедура ProcSetBack

17. Создать процедуру ProcOnOff, которая в зависимости от значения переменной off (добавить её с помощью мутатора) устанавливает режим видимости для некоторых компонент экрана.

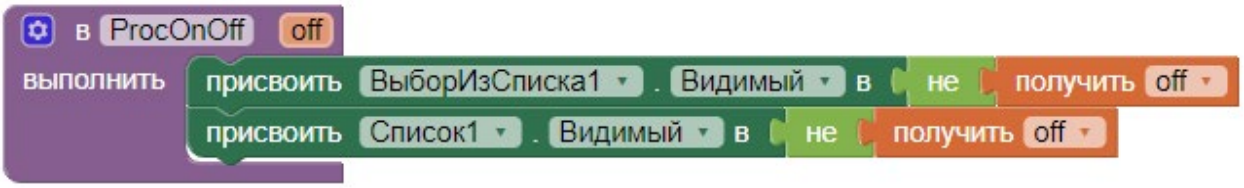


Рисунок 173. Процедура ProcOnOff

18. Перетащить на окно просмотра блоков обработчик событий выбора элемента для компоненты ВыборИзСписка1 и из раздела Процедуры добавить блок «вызвать» для процедуры ProcSetBack. В качестве параметра x указать свойство Выбор. Создать аналогичный обработчик для компоненты Список1

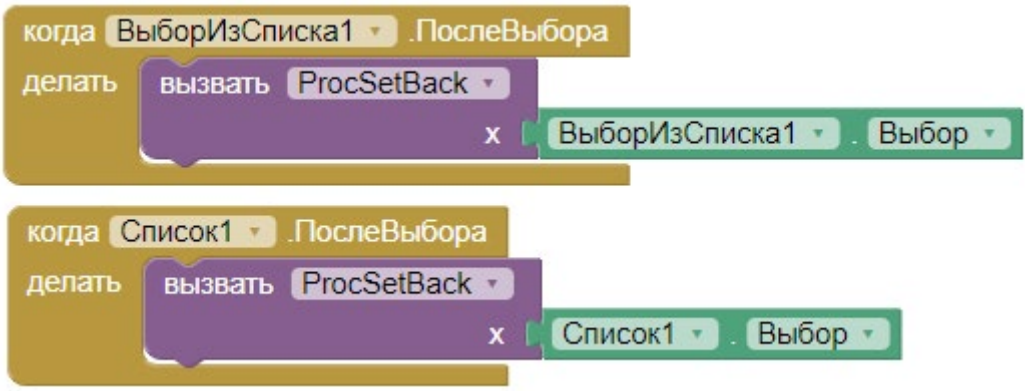


Рисунок 174. Обработчики событий выбора элемента в списке

19. Добавить обработчик для компоненты Switch1. В логике обработки дополнительно учитывается изменение цвета текста компонента с чёрного на белый, когда фон становится черным, и наоборот. Также посредством процедуры ProcOnOff меняется режим видимости списков.

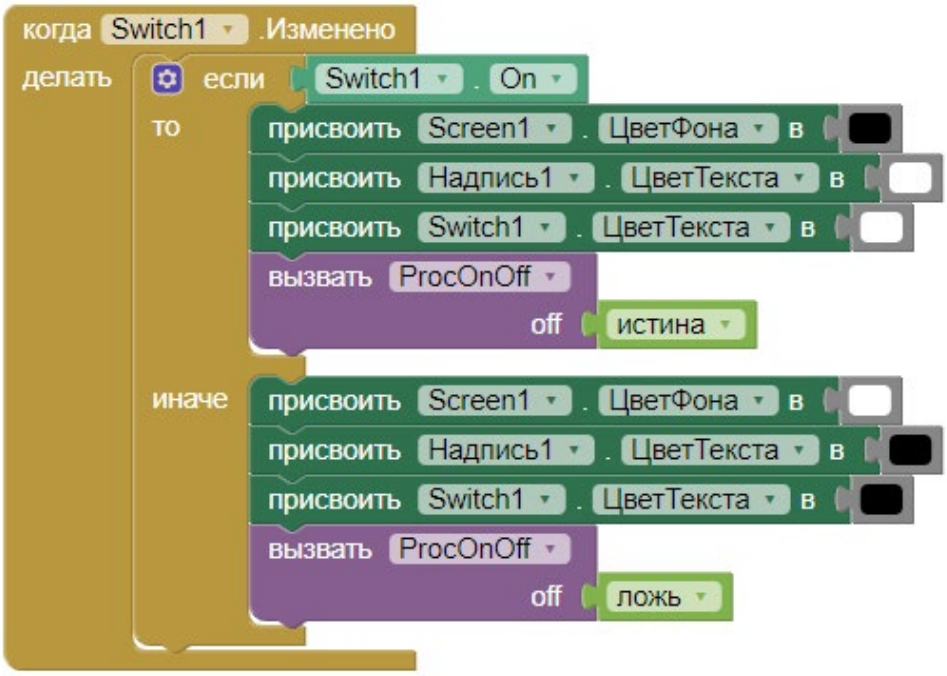


Рисунок 175. Обработчик изменения положения переключателя

20. Добавить возможность изменения фона экрана посредством выбора из галереи для компоненты ВыборщикИзображений1:

Рисунок 176. Обработчик выбора для компоненты ВыборщикИзображений1

21. В итоге финальная схема блоков приобретает следующий вид:

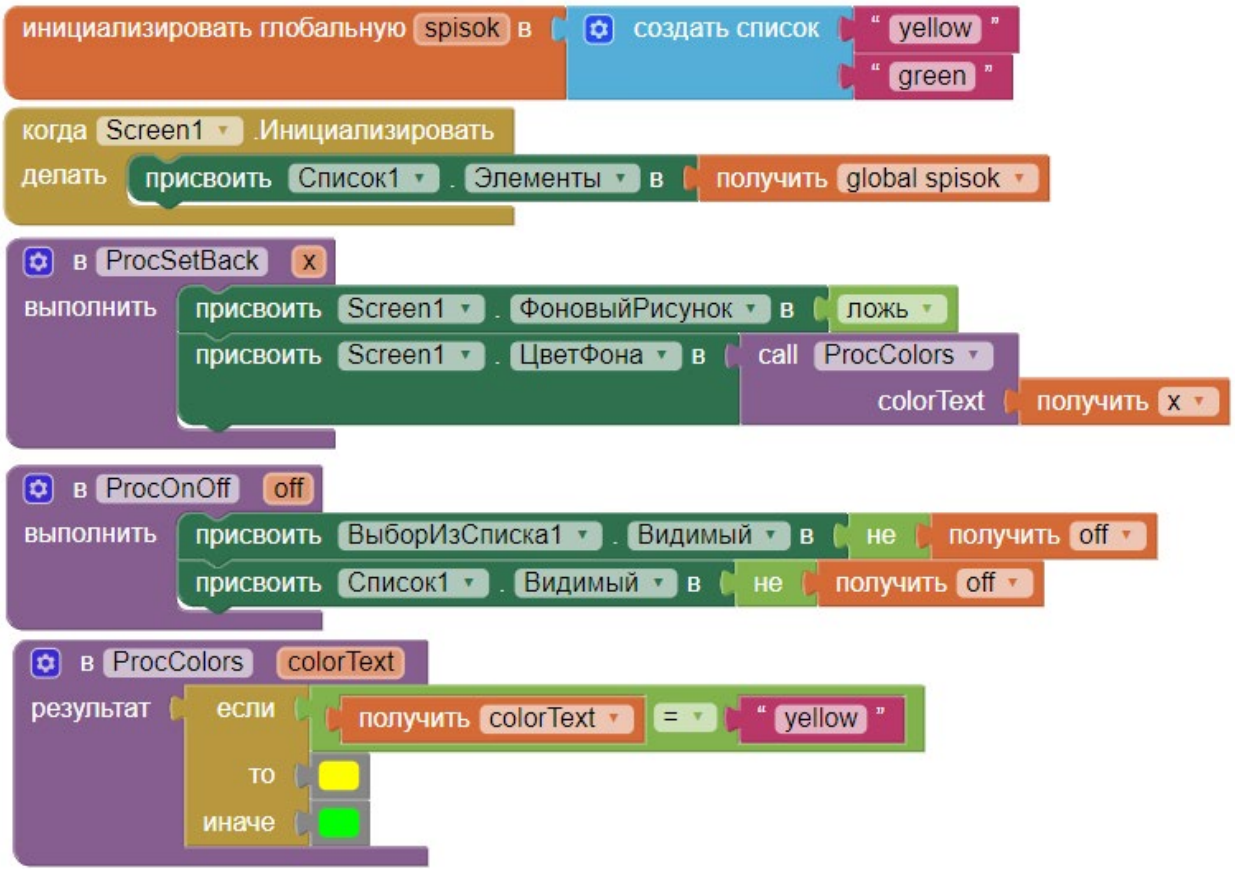


Рисунок 177. Финальный вид схемы блоков приложения, часть 1

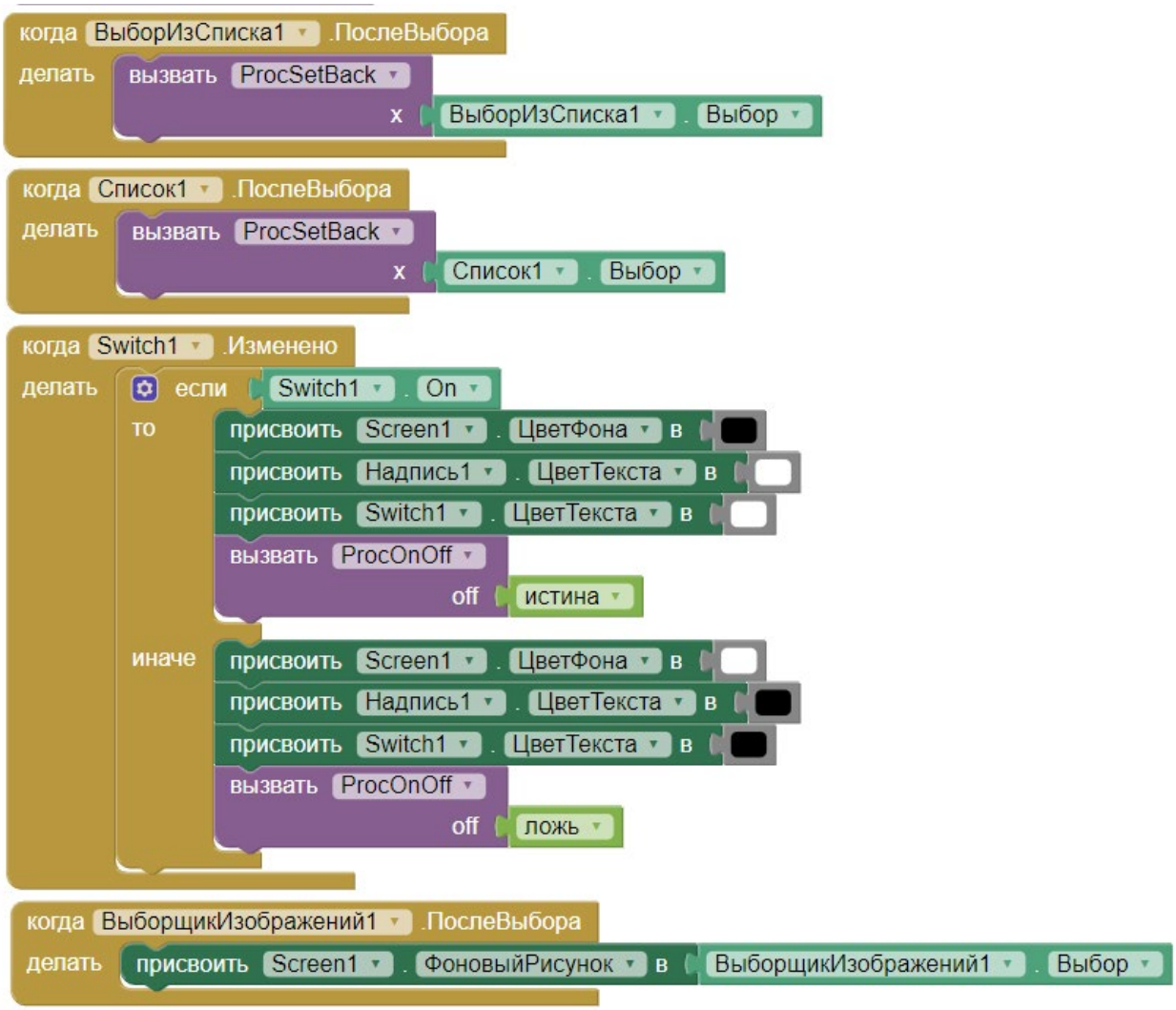


Рисунок 178. Финальный вид схемы блоков приложения, часть 2

22. Запустить эмулятор и проверить работоспособность программы.

Дополнительные задания:

1. Добавить озвучивание выбранного цвета при помощи компоненты ТекстВРечь1. Для этого достаточно видоизменить процедуру ProcSetBack следующим образом:

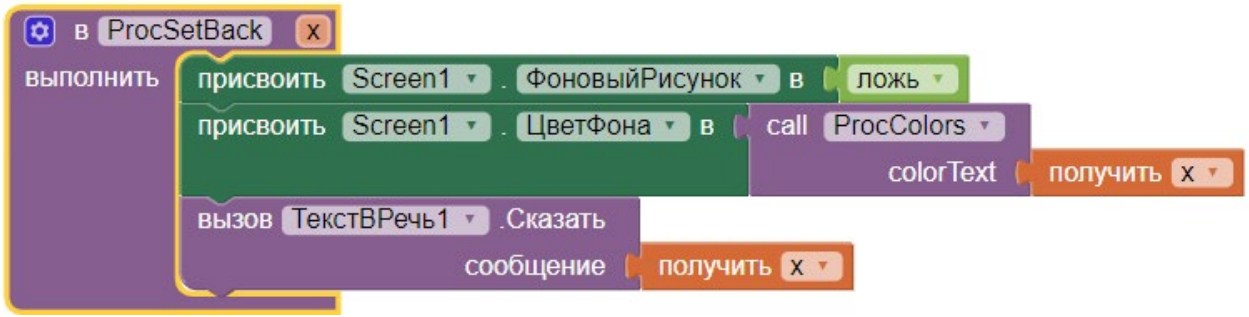


Рисунок 179. Модификация процедуры ProcSetBack

2. Создать приложение, где в зависимости от выбора в списке показывалось бы то или иное изображение на экране. Использовать компоненты Изображение и Список.
3. Создать приложение, в котором элементы списка 2 заполнялись бы в зависимости от выбранного элемента списка 1.
4. Добавить компоненту ИндикаторОжидания и реализовать её функционал аналогично компонентам Список, ВыбориИзСписка.

Выводы: в данной лабораторной работе изучается работа с компонентами Список, ВыборИзСписка, ИндикаторОжидания, Переключатель, Выборщик-Изображения, Текст-ВРечь.

Контрольные вопросы:

1) Опишите, что собой представляют списки
2) Для чего нужен блок Switch?
3) Какие основные блоки используются для работы со списками?

Лабораторная работа 6. Игра «Счастливая семёрка»

Теоретическая часть

В данной работе в том числе используется теоретический материал из дидактических материалов и урока №2.

Компонента Изображение используется для отображения картинки. Может использоваться как напрямую на экране, так и с компонентой Выборщк

Изображений. Расположена в разделе «Интерфейс пользователя». Основные свойства следующие:

Высота и Ширина, ЦветФона точно такие же, как у всех остальных компонент.

Изображение — задается изображение в виде файла. Предварительно должно быть загружено в проект посредством кнопки «Загрузить файл» в окне Медиа. Можно использовать форматы jpg и png:

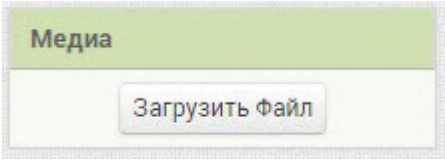


Рисунок 180. Загрузка медиафайлов

Угол поворота — позволяет изменять угол наклона изображения, «вертеть» его. Работает не во всех версиях Андроид.

Масштабировать Изображение До Соответствия — размер изображения изменяется под размер компонента. Используется при работе с компонентой ВыборщикИзображений.

В режиме Блоки у компоненты имеются типовые зелёные блоки геттеров и сеттеров и один обработчик кликов:

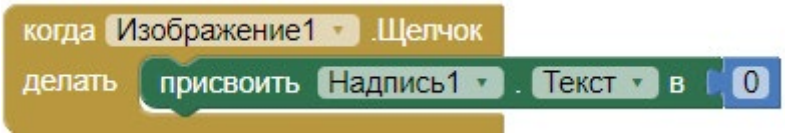


Рисунок 181. Обработчик нажатия на Изображение

Чтобы он функционировал, необходимо, чтобы свойство Clickable было отмечено галочкой или установлено через сеттер. Только тогда в режиме Блоки можно применять обработчики кликов Изображения, иначе обработчик работать не будет даже при наличии.

Практическая часть

Цель работы: освоить работу с компонентой Изображение, типовыми компонентами раздела «Интерфейс Пользователя» и «Расположение», блоками раздела Логика. Реализовать игру «Счастливая семёрка».

Ход работы

1. Перейти по адресу http://ai2.appinventor.mit.edu/ и запустить среду АИ (при необходимости авторизоваться на сайте «App Inventor»).
2. Через пункт главного меню «Проекты» -> «Начать новый проект» ... создать новый проект под названием Seven.

Необходимо реализовать приложение-игру со следующими правилами: при встряхивании телефона случайным образом генерируются и отображаются три числа, если одно из них является семёркой, то пользователь получает 0,25 очка, если две семёрки — 0,5 очка, если все три числа — семёрки, то пользователь получает 1 очко, иначе — поражение.

3. Перенести компоненты из раздела «Интерфейс пользователя» на экран приложения (Screen1) и переименовать согласно следующей схеме:

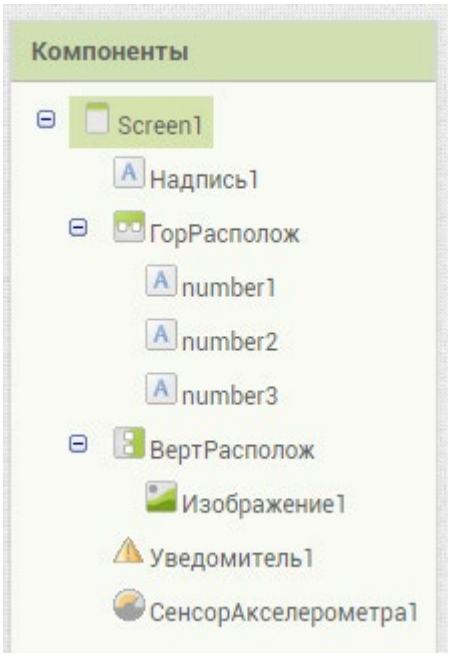


Рисунок 182. Примерная схема компонент экрана

4. Установить следующие свойства компоненты экран (Screen1):
   a. ВыровнятьПоГоризонтали по центру — Центр
   b. заголовок — «Seven»
5. Для надписи Надпись1:
   a. текст — «777»
6. Для горизонтального расположения ГорРасполож:
   a. ВыровнятьПоГоризонтали по центру — Центр
   b. ширина — Наполнить родительский

7. Для трёх надписей внутри расположения ГорРасполож:
   a. ширина — Наполнить родительский
   b. РазмерШрифта — 24
8. Для вертикального расположения ВертРасполож:
   a. ширина — Наполнить родительский
   b. ВыровнятьПоГоризонтали — Центр
   c. высота — 200 pixels
9. Для Изображения1:
   a. изображение — загрузить Рисунок GoldenKub.png через раздел Медиа
   b. видимый — нет (убрать галочку)
10. Убедиться, что финальная схема компонент соответствует следующему дизайну:

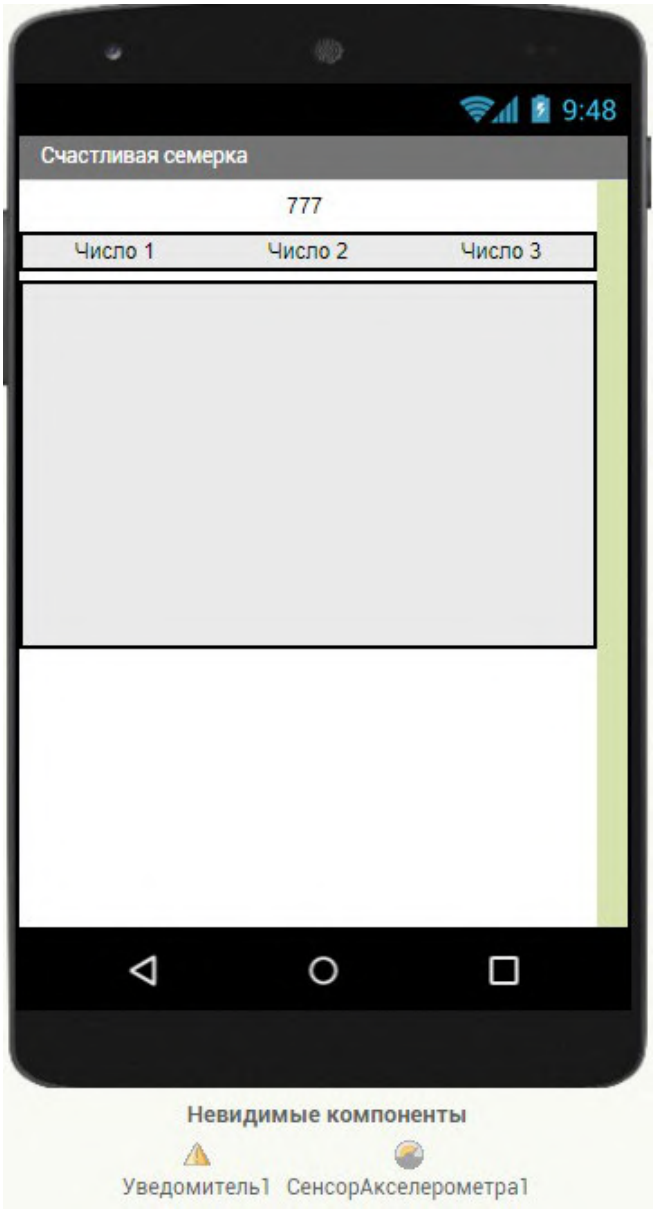


Рисунок 183. Дизайн приложения

11. Перейти в раздел «Блоки».
12. Добавить для сенсора акселерометра обработчик событий встряхивания мобильного устройства:

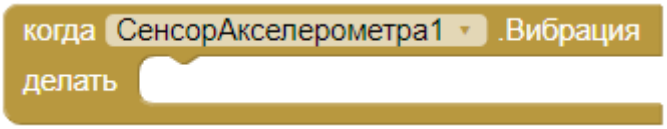


Рисунок 184. Обработчик события встряхивания мобильного устройства

13. Инициализировать переменные для подсчёта побед и поражений. Переменная подсчёта побед является вещественной, так как. при подсчёте используются не только целые числа:

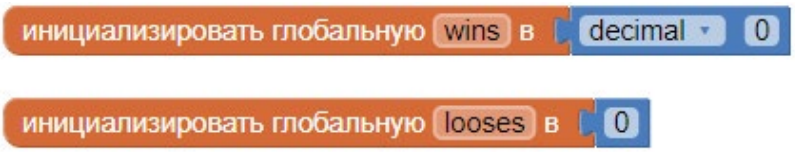


Рисунок 185. Переменные wins, looses

14. Определить процедуру Proc1 для присвоения полям number1, number2, number3 случайных целых чисел в диапазоне от 0 до 9.

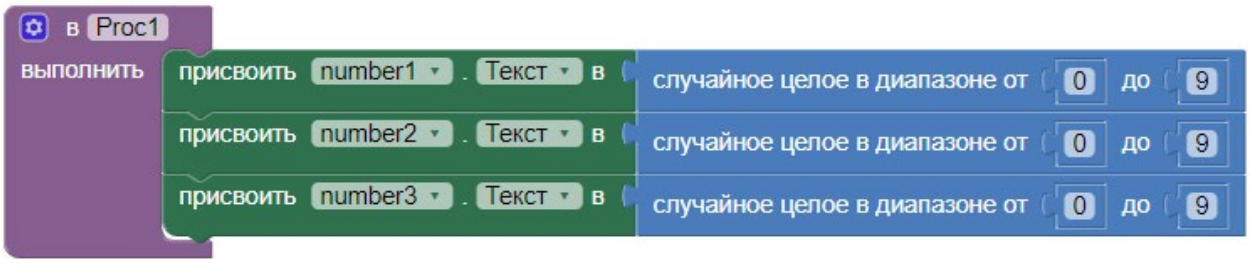


Рисунок 186. Процедура, присваивающая текстовым полям значения случайных чисел

15. Добавить процедуру Proc1 в обработчик для сенсора акселерометра:

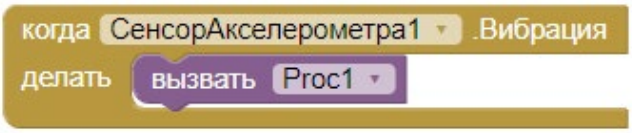


Рисунок 187. Вызов процедуры Proc1 в обработчике сенсора акселерометра

16. Перетащить из раздела Логика условие «логическое ИЛИ» и добавить в него 3-й параметр при помощи мутатора:

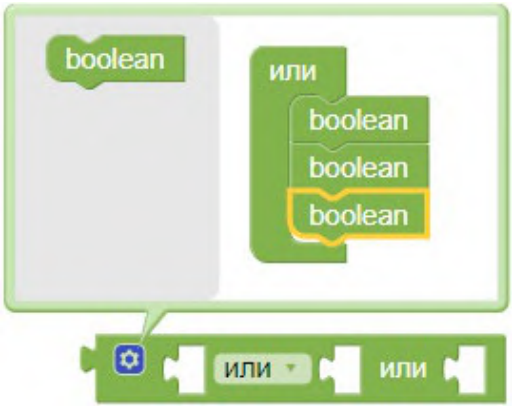


Рисунок 188. Добавление параметра при помощи мутатора

17. Внутри каждого параметра добавить логическое условие равенства вида для каждой из надписей приложения.

Рисунок 189. Проверка равенства текста надписи number1 значению 7

В итоге создать общее условие, проверяющее, равен ли текст значению 7 хотя бы в одной из надписей: number1, number2, number3. Общее условие формируется как «логическое ИЛИ», так как достаточно срабатывания хотя бы одного условия.

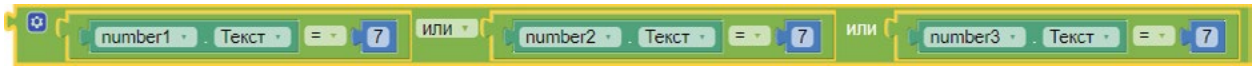


Рисунок 190. Общий вид условия

18. Поместить получившийся блок условий в новую процедуру is1Seven, которая будет проверять, что выпала хотя бы одна семёрка из трёх:

Рисунок 191. Проверка генерации хотя бы одной семёрки

19. Создать процедуры, проверяющие варианты попарного появления семёрок:

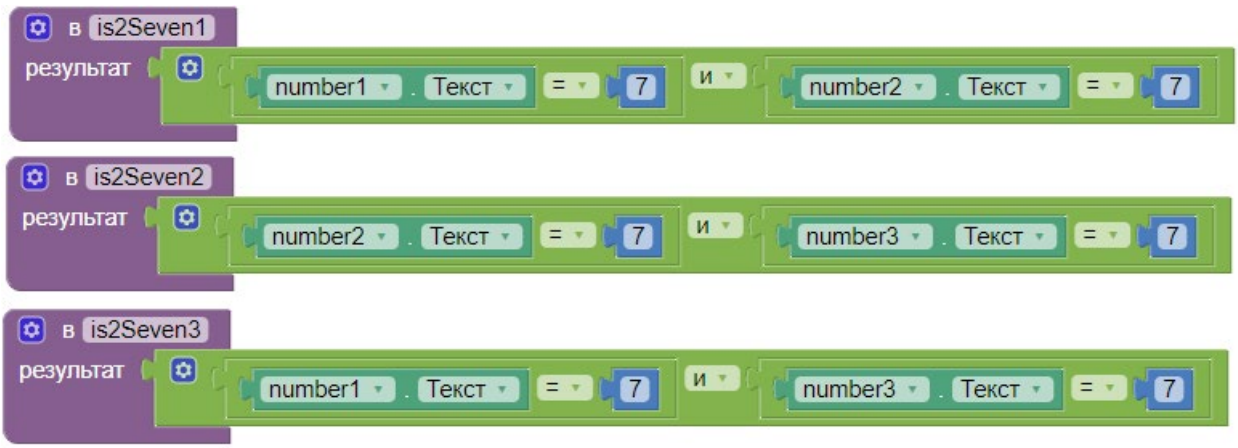


Рисунок 192. Процедуры is2Seven1, is2Seven1, is2Seven1

20. Создать общую процедуру is2Seven1 для случая, что случился один из трёх попарных вариантов (isSeven1, isSeven2, isSeven3), описанных в предыдущем пункте. Вызовы процедур типа «call is2Seven1» появляются в разделе «Процедуры» сразу же после создания соответствующих про-цедур.

Рисунок 193. Проверка события генерации двух семёрок

21. Создать процедуру для случая генерации трёх семёрок:

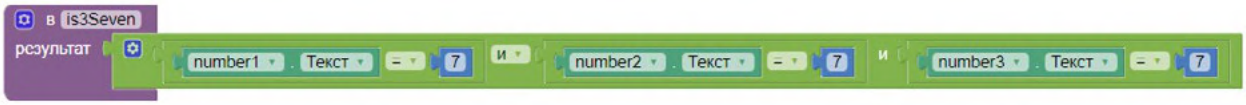


Рисунок 194. Проверка события генерации трех семёрок

22. В обработчик сенсора акселерометра добавить процедуру Proc1 для генерации случайных чисел. Добавить и видоизменить при помощи мутатора блок «если ...то... иначе»:

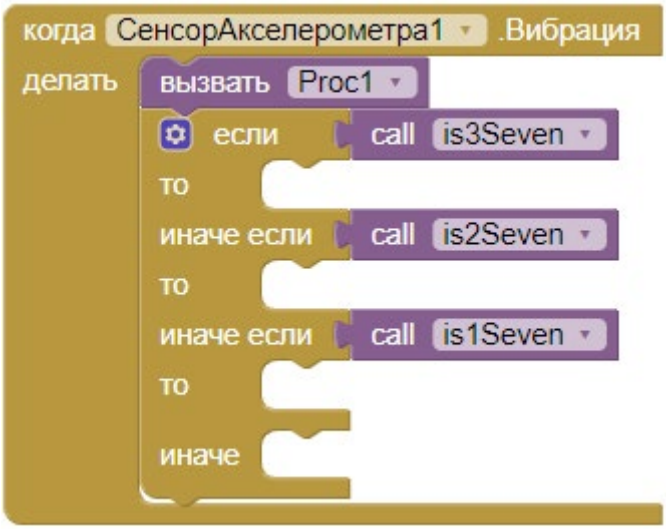


Рисунок 195. Общий вид условия в обработчике нажатия на кнопку Start

23. В случае появления хотя бы одной семёрки появляется изображение золотого куба и надпись «Полная победа! 777!». Использовать не целое, а вещественное число из математического блока «decimal»:

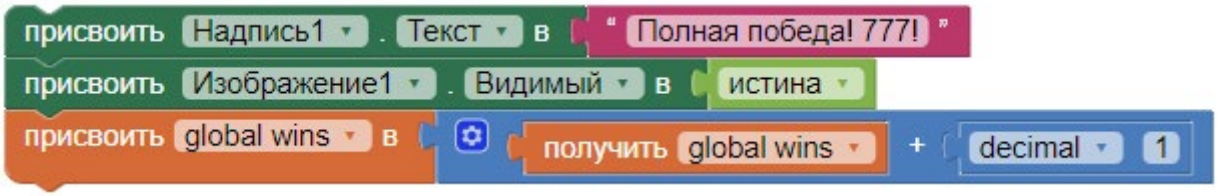


Рисунок 196. Сеттеры для компонент Надпись1, Изображение1 и переменной wins в случае выигрыша

24. Иначе появляются надписи: «Полупобеда», «Семёрка» или «Поражение» (без изображения золотого куба). Обработчик события для сенсора акселерометра должен иметь примерно следующий вид:

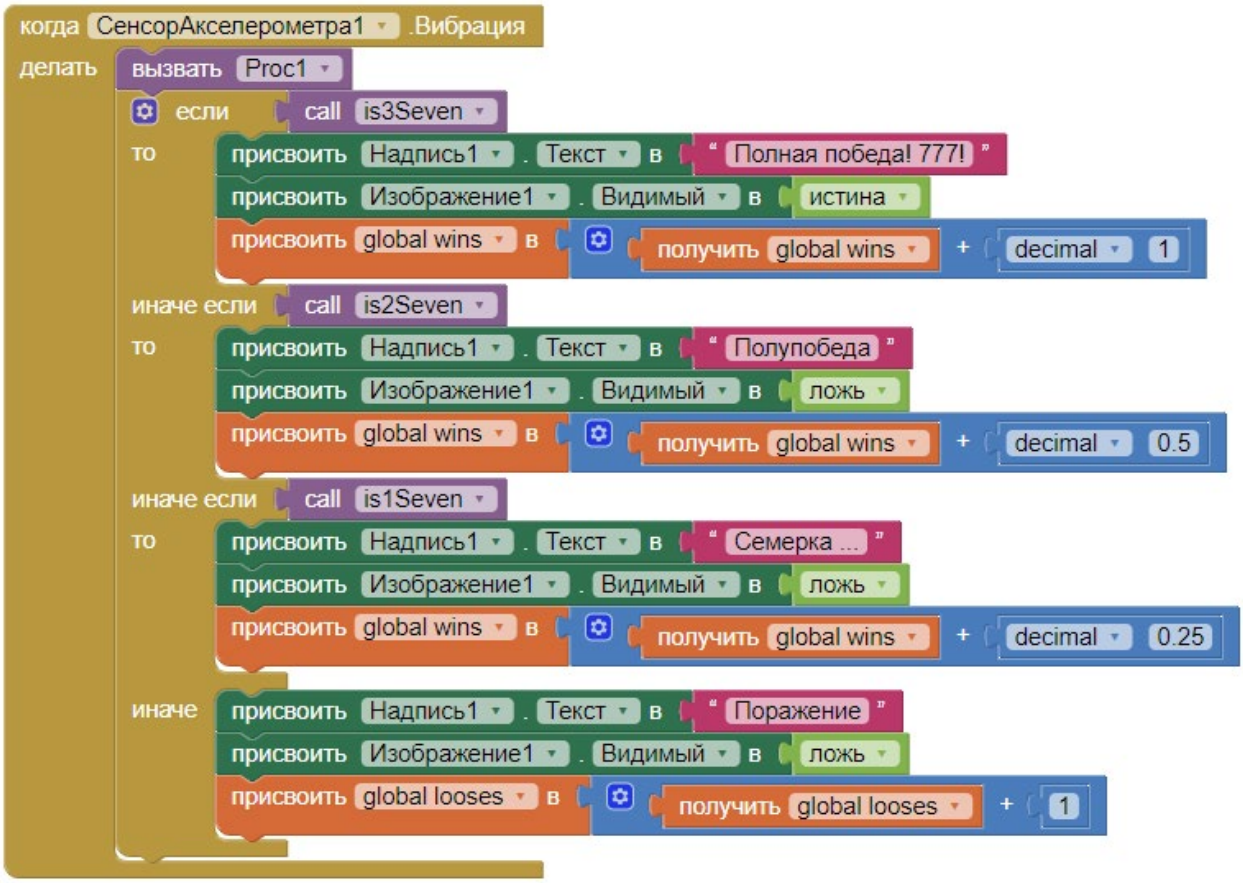


Рисунок 197. Программные блоки, выполняемые при встряхивании мобильного устройства

25. Данное приложение не получится испытать в эмуляторе, так как невозможно воспроизвести потряхивание мобильного устройства. Поэтому необходимо подключить мобильное устройство через USB и протестировать работу приложения.

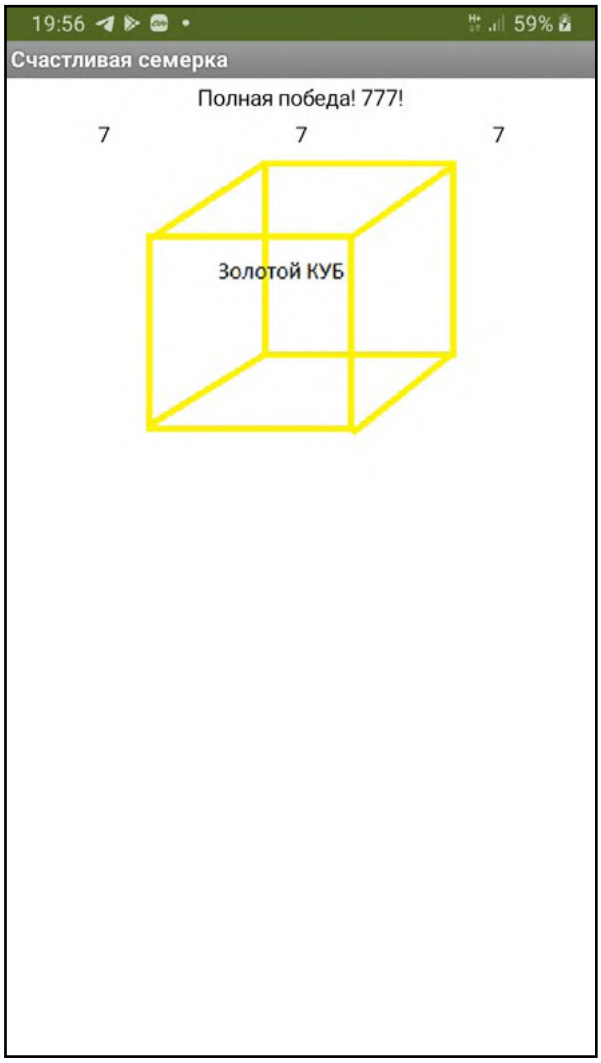


Рисунок 198. Экран приложения в мобильном устройстве

26. Поменять местами блоки проверки условий is3Seven и is2Seven, т. е. сначала проверять, не выпало ли две семёрки, и только потом — не выпало ли три. Подумать, почему приложение перестало правильно работать.

Рисунок 199. Перемена местами блоков is3Seven и is2Seven

27. Упростить процедуру is3Seven. По аналогии с is2Seven в ней можно вызывать любые две процедуры из уже созданных процедур is2Seven1, is2Seven2, is2Seven3. Это проще, чем создавать новое условие, проверяющее значения выпавших чисел.

28. Улучшить дизайн приложения: изменить шрифты, цвета, добавить вспомогательные элементы интерфейса. Протестировать работу приложения.

Дополнительные задания:

1. Создать глобальные переменные number1, number2, number3, хранить и извлекать сгенерированные случайные числа только посредством переменных, а не через геттеры надписей number1, number2, number3.
2. Добавить следующие изменения в приложение:
   a. В случае победы давать вибрацию длительностью 0,25 секунды.
   b. В случае генерации 2 или 3 семёрок установить вибрацию длительностью в 1 секунду.
3. Добавить обработчик клика на Изображение, который выводит простое уведомление со счётом поражений и побед:

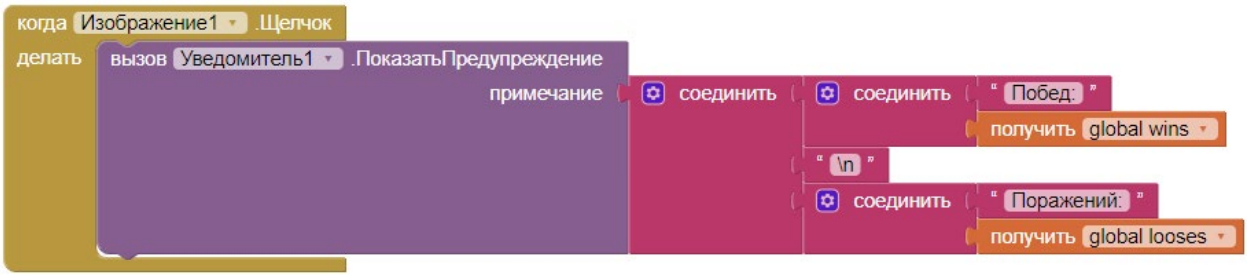


Рисунок 200. Вызов уведомления со счётом игры

Выводы: в данной лабораторной работе изучаются базовые компоненты и блоки среды АИ с акцентом на блоки раздела Логика.

Контрольные вопросы:

1. Почему первым должен проверяться случай с тремя семёрками?
2. Для чего нужен оператор «если... иначе...»?
3. Как расположить несколько текстов горизонтально?
4. Как добавить изображение на экран?
5. Как скрыть компоненту?

Лабораторная работа 7. Игра «Сопоставь цвета»

Теоретическая часть

В данной работе в том числе используется теоретический материал из дидактических материалов и из урока 3 «Анимация».

Практическая часть

Цель работы: освоить работу с компонентой Шар. Закрепить работу с компонентой Часы. Создать игру «Сопоставь цвета».

Ход работы

1. Запустить среду АИ, перейдя по адресу http://ai2.appinventor.mit.edu/, и зарегистрировавшись при необходимости.

2. Посредством пункта главного меню «Проекты» -> «Начать новый проект ...» создать новый проект под названием MatchTheColors.

Необходимо создать приложение-игру со следующей логикой: пользователь должен успеть нажать пальцем на шар, если цвета шара и надписи совпали. Цвета меняются согласно интервалу таймера. Ведётся подсчёт успешных и неуспешных попыток. По истечении заданного времени игра заканчивается.

3. Настроить свойства экрана (Screen1) следующим образом:
   заголовок — Сопоставь цвета
4. Добавить на экран компоненты следующие компоненты:

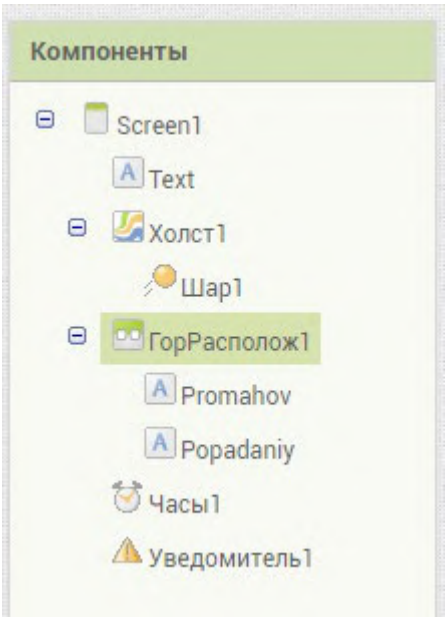


Рисунок 201. Схема компонент экрана

5. Настроить свойства надписи Text следующим образом:
   Текст — «Цветная надпись»;
   Размер шрифта — 24.
6. Настроить свойства горизонтального расположения ГорРасполож1 и надписей Promahov, Popadaniy следующим образом:
   а. ширина — наполнить родительский

7. Убедиться, что финальная схема компонент соответствует следующему виду:

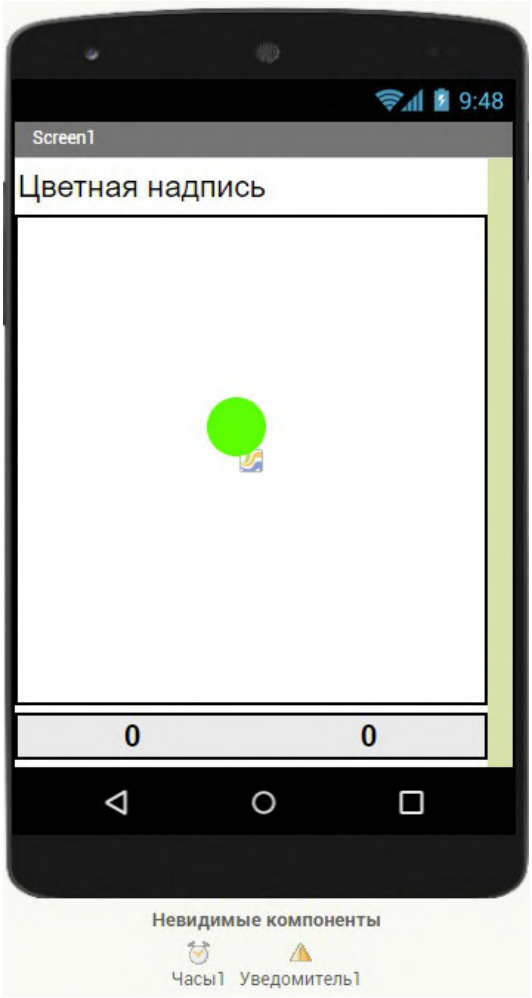


Рисунок 202. Примерная схема дизайна приложения

8. Перейти в режим Блоки.
9. Работа программы представлена тремя основными группами блоков:
   a. блоки инициализации
   b. блоки, выполняющиеся под контролем таймера компоненты Часы1
   c. блоки обработчика касания компонента Шар1
10. В блоках инициализации задаются переменные:
    a. time — определяет продолжительность игры
    b. colorOfShar — цвет шара Шар1
    c. colorOfText — цвет надписи Text
    d. countOfPopadaniy — количество правильных сопоставлений цветов шара и надписи
    e. countOfPromahov — количество ошибочных сопоставлений

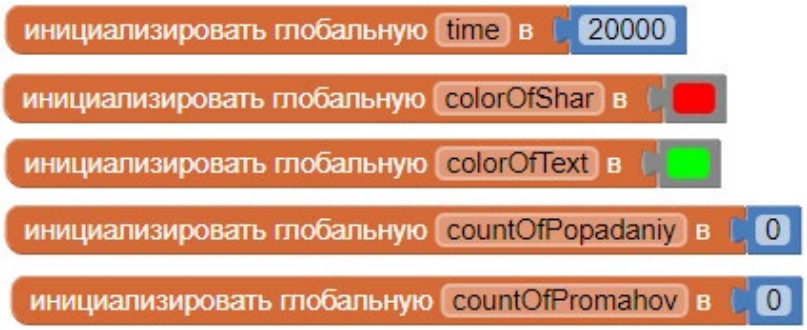


Рисунок 203. Блоки инициализации переменных.

11. В блоке обработки событий Таймера последовательно выполняются следующие команды:

a. переменным colorOfShar и colorOfText присваиваются случайные цвета из создаваемых «на лету» списков цветов

b. цвет шара принимает значение colorOfShar

c. цвет фона надписи Text принимает значение colorOfText

d. шар становится доступным для касания

e. переменная time уменьшается на значение интервала таймера (1 секунда)

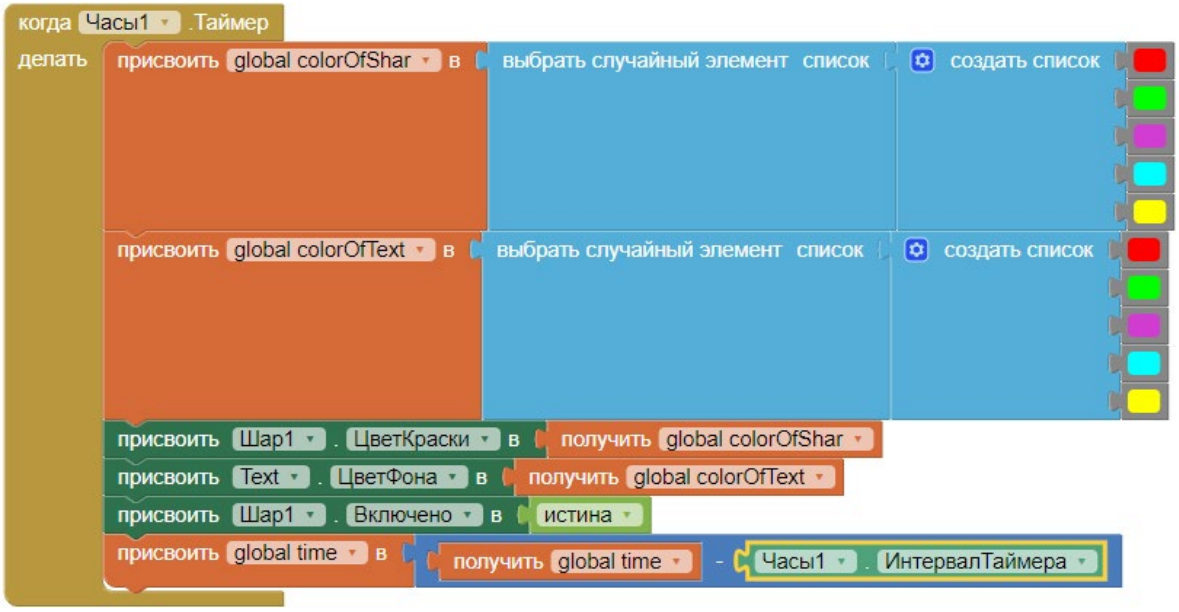


Рисунок 204. Блоки в обработчике событий таймера

12. В блоке обработчика касаний Шара последовательно выполняются следующие команды:

a. если время игры истекло, то появляется уведомление в виде всплывающего сообщения «Игра закончена, дружок ...» с количеством верных попаданий и промахов

b. если время игры не истекло, то проверяется новое условие, совпадают ли цвета шара и фона надписи Text, в этом случае переменная для количества попаданий увеличивается на единицу

c. иначе, если цвета не совпадают, на единицу увеличивается переменная для подсчёта количества промахов

d. компонент Шар1 перестаёт быть доступным до начала следующего интервала (такта) Таймера компоненты Часы1

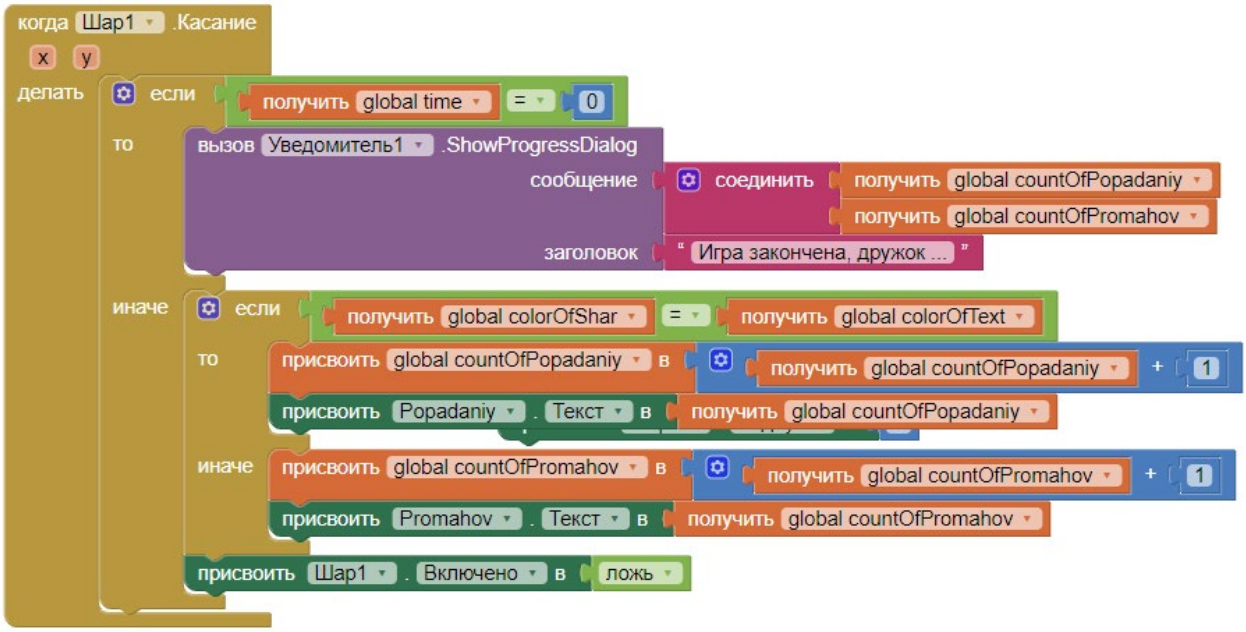


Рисунок 205. Обработчик касания шара

13. Запустить эмулятор и проверить работу приложения.

Выполнить Дополнительные задания:

1. Перенести два одинаковых списка цветов из блоков обработки интервалов Таймера в глобальную переменную-список. После чего в блоке обработки интервалов Таймера ссылаться оба раза только на эту переменную, а не создавать лишние списки «на лету».

Рисунок 206. Создание списка цветов

2. Усложнить игру и дополнительно выводить значение цвета в надписи Text согласно интервалам Таймера. Для этого создать список с обозначениями цветов и случайно выбирать одно из значений в блоке обработки событий Таймера.

Рисунок 207. Создание списка обозначений цветов

3. В режиме Дизайнер добавить надписи с текстом: «Количество промахов:» и «Количество попаданий:». Можно добавить для них ещё одну компоненту ГоризонтальноеРасположение и разместить перед надписями «Promahov», «Popadaniy».
4. Проверить работу приложения.

Выводы: в данной лабораторной работе создаётся игра «Сопоставь цвета» и осваивается работа с компонентами раздела «Рисование и анимация». Дополнительно прорабатываются типовые программные блоки.

Контрольные вопросы:

1. Как увеличить значение переменной на 1?
2. Что такое компонента Шар?

Лабораторная работа 8. Игра «Шары»

Теоретическая часть

В данной работе в том числе используется теоретический материал из дидактических материалов и из урока 3 «Анимация».

Практическая часть

Цель работы: Освоить работу с компонентами Холст, Спрайт, Шар из раздела Анимация. Создать игру «Шары».

Ход работы

1. Перейти по адресу http://ai2.appinventor.mit.edu/ и запустить среду АИ (при необходимости авторизоваться на сайте «App Inventor»)
2. Через пункт главного меню «Проекты» -> «Начать новый проект ...» создать новый проект под названием Game2Sprite.

Необходимо создать одноэкранное приложение, состоящее из Спрайта в форме куба с надписью «IT-куб» внутри и двух Шаров. Спрайт можно перетаскивать ЛКМ. При инициализации главного экрана приложения задаются координаты скорости, направления и интервалы (такты) движения шаров. Шары двигаются с изначально заданными скоростью, курсом и интервалом (тактом) движения. При попадании об стенку или друг об друга скорость и интервалы движения меняются случайным образом. При попадании одного из шаров в спрайт появляется Уведомление и пользователь может сделать выбор: прекратить игру или начать её заново.

3. Перенести компоненты из раздела «Интерфейс пользователя» на экран приложения (Screen1) согласно следующей схеме:

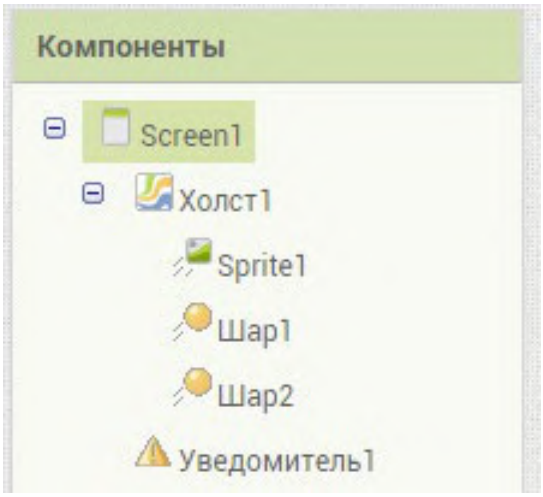


Рисунок 208. Общая схема компонент экрана

4. У главного экрана Screen1 установить свойства:
   a. заголовок — Уверните от шаров
   b. ВыровнятьПоГоризонтали — Центр
   c. Theme — Device Default
5. У компоненты Холст1 установить свойство:
   a. Изображение — itcube2.png;
   b. X — 149
   c. Y — 240
6. У Sprite1 установить свойства:
   a. ширина — Наполнить родительский
   b. подсказка — «Записать слово ...»
7. У Шар1 установить свойства:
   a. ЦветКраски — Синий
   b. для удобства можно переименовать в СинийШар
8. У Шар2 установить свойства:
   ЦветКраски — Красный
9. Для удобства можно переименовать в КрасныйШар;
10. Убедиться, что финальная схема компонент соответствует следующему виду:

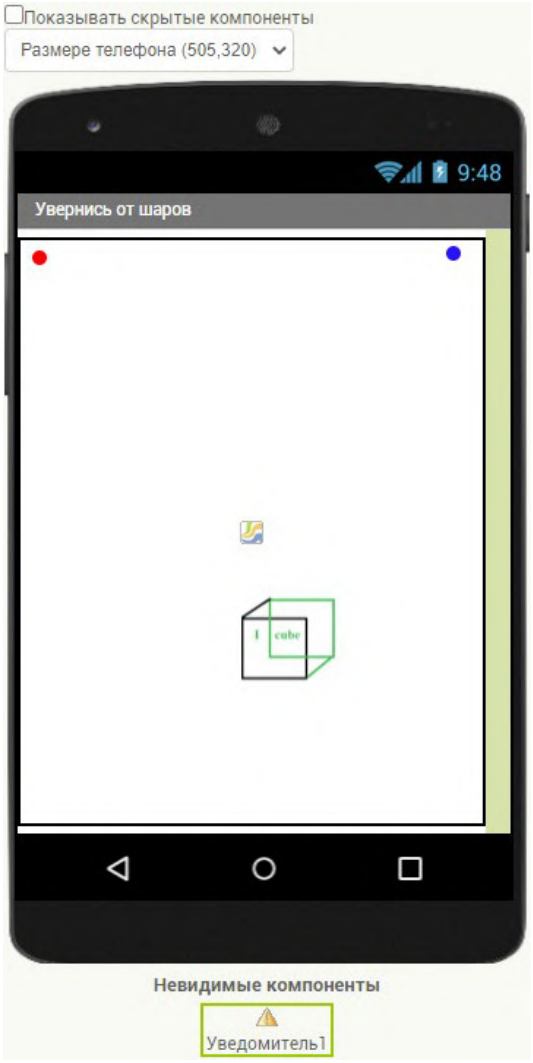


Рисунок 209. Примерная схема дизайна приложения

11. Перейти в режим Блоки;
12. В режиме блоки первого экрана находятся три основные группы блоков:
    a. блоки работы с шарами
    b. блоки работы со спрайтом
    c. блоки работы с уведомителем
13. Определить процедуру Proc1 с настройками скорости, интервала и курса для шаров:

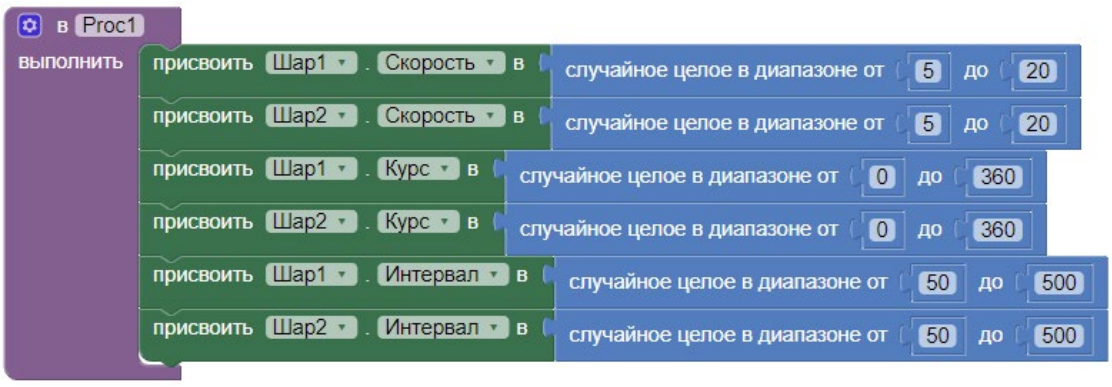


Рисунок 210. Процедура инициализации параметров шаров

14. Вызвать процедуру Proc1 при инициализации экрана:

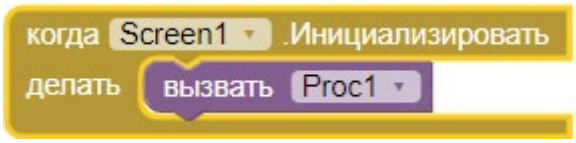


Рисунок 211. Обработчик инициализации экрана

15. Создать обработчики событий достижения шарами краев компоненты Холст1:

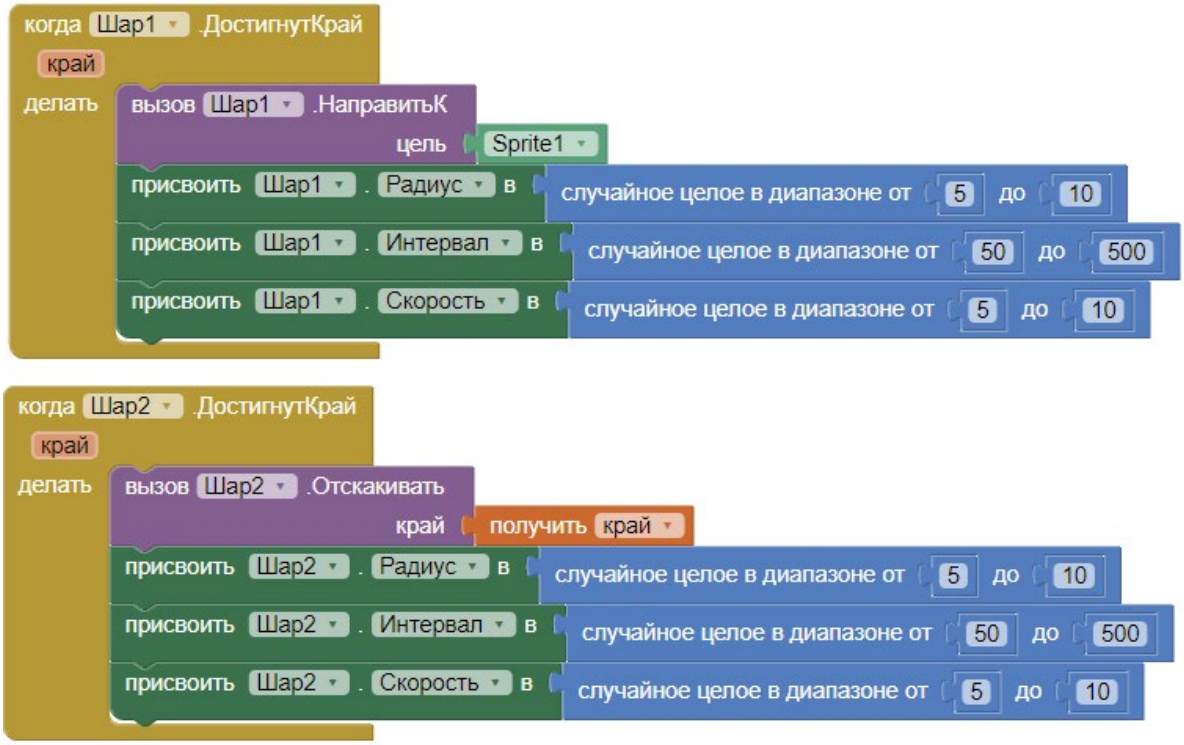


Рисунок 212. Обработчики достижения шарами границ холста

Шар1 после достижения границ области холста сразу направляется к спрайту. А Шар2 отскакивает от границы (края) согласно законам физики.

16. Задать случайное изменение курса и удвоение скоростей движения шаров при их столкновении друг с другом:

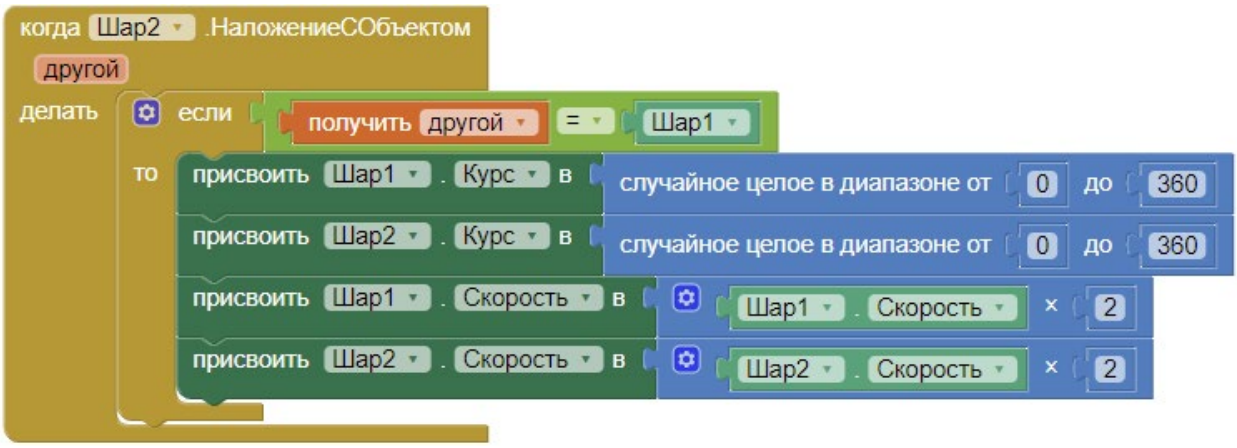


Рисунок 213. Обработчик столкновений шаров друг с другом

17. Создать следующий обработчик события столкновения ЛЮБОГО из шаров со спрайтом:

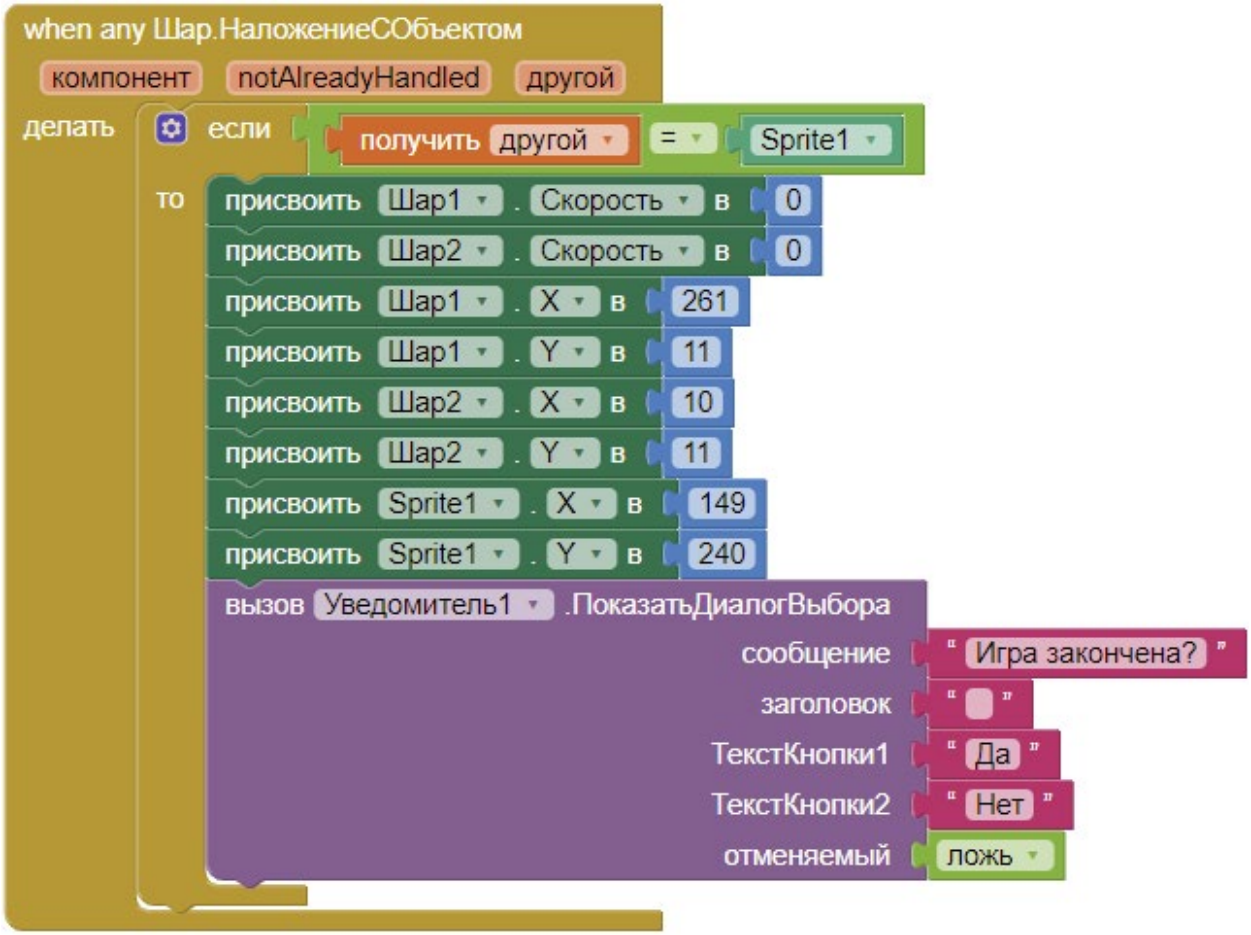


Рисунок 214. Обработчик столкновений любого из шаров со спрайтом

Здесь нюансом является использование блоков из раздела «Любой компонент», чтобы не дублировать один и тот же код для первого и второго шара.

В случае столкновения любого из шаров со спрайтом:

- обнуляются скорости шаров, и они раскидываются по начальным координатам, заданным в режиме Дизайнер
- вызывается Уведомитель1, посредством которого можно выбрать варианты «Да/Нет» для продолжения или завершения игры

18. Создать обработчик события для кнопок Да/Нет Уведомителя1:

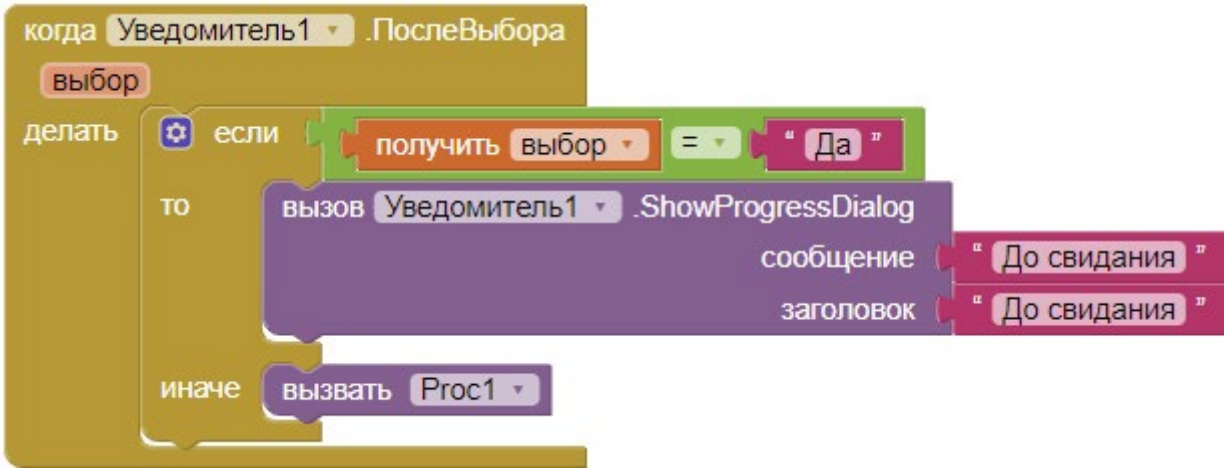


Рисунок 215. Обработчик выбора кнопок уведомителя

Если пользователь нажимает «Нет», то игра выдаёт завершающее сообщение «До свидания», иначе вызывается процедура Proc1 и шары запускаются заново.

19. Создать две группы блоков, отвечающих за функционирование компоненты Спрайт1:

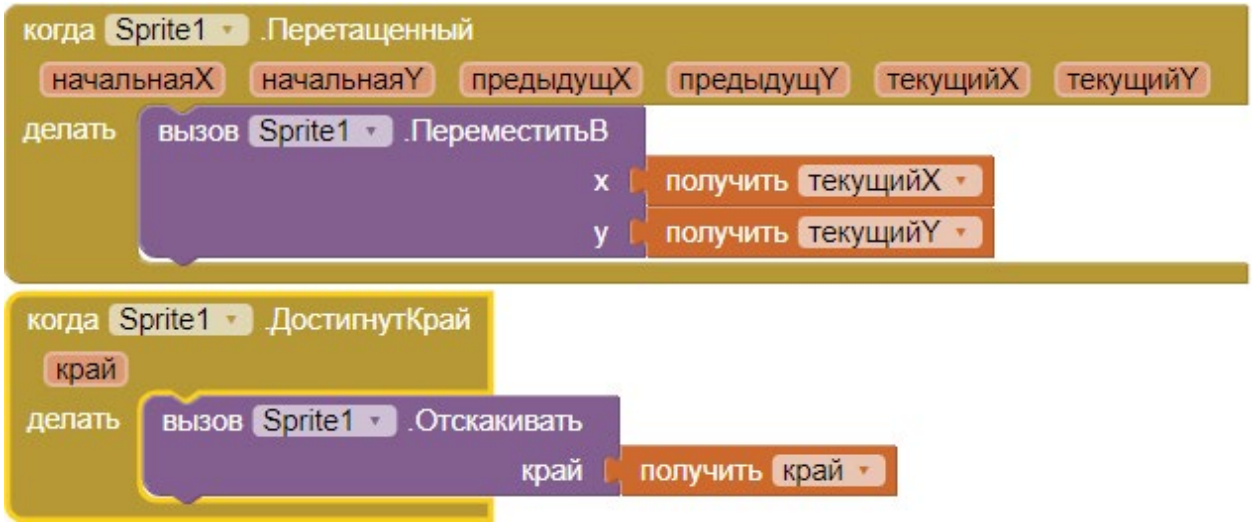


Рисунок 216. Обработчики событий перетаскивания спрайта пальцем и достижения им границ холста

Первый обработчик событий реализует логику перемещения спрайта при перетаскивании его пальцем (мышью в эмуляторе)

Второй обработчик ограничивает перемещение спрайта за границы холста.

20. Запустить эмулятор и проверить работу приложения.

Дополнительные задания:

1. Добавить третий зелёный шар и настроить логику его движения аналогично синему шару.
2. Увеличить значения всех скоростей ещё в 2 раза.
3. Запустить эмулятор и проверить работу приложения.
4. Модифицировать логику игры с помощью блока процедуры «вызов Шар1.ПереместитьВ», перемещающей Шар или Спрайт в заданную точку с координатами (x, y).

Например, реализовать логику перемещения Шара1 в определённую координату при уменьшении скорости до 10.

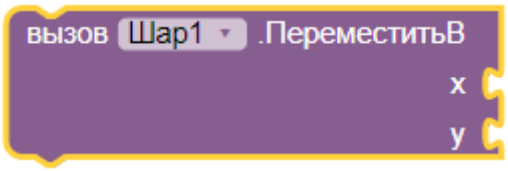


Рисунок 217. Вызов процедуры перемещения шара в точку с координатами x,y

5. Добавить компоненту Надпись и реализовать в ней подсчёт очков.

Выводы: в данной лабораторной работе создаётся приложение, в котором осваивается работа со прайтам и шарами из раздела «Рисование и анимация».

Контрольные вопросы:

1. Чем компонента Спрайт отличается от Шара?
2. Для чего нужен обработчик события достижения края?

Лабораторная работа 9. Web-приложение

Теоретическая часть

В данной работе в том числе используется теоретический материал из дидактических материалов.

Компонента WebПросмотрщик используется для просмотра Web-страниц. Расположена в разделе «Интерфейс пользователя». Основные свойства следующие:

Высота и Ширина, ЦветФона — точно такие же, как у всех остальных компонент;

ДомашняяСтраница — задаётся страница, которая по умолчанию загружается в компоненте;

ПерейтиПоСсылка — если отмечено галочкой, разрешается переход по ссылкам в загруженных страницах. После этого в режиме Блоки можно использовать процедуры «call WebПросмотрщик1.goBack», «call WebПросмотрщик1.goForward» и «call WebПросмотрщик1.goHome», при помощи которых осуществлять переход назад/вперёд по уже пройденным ссылкам:

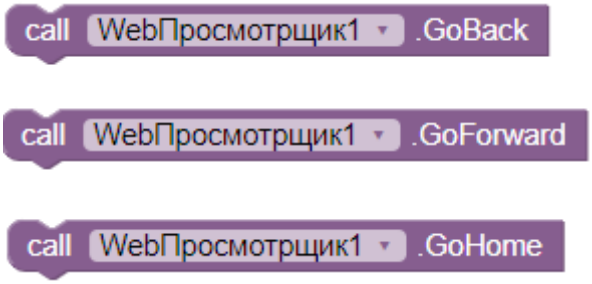


Рисунок 218. Вызов процедур перехода по ссылкам назад, вперед и домой

В режиме Блоки у компоненты имеются процедуры, типовые геттеры, сеттеры, обработчики загрузки страницы, ошибок при загрузке и ряд других обработчиков.

В данной работе используется процедура, дающая компоненте команду для перехода на заданную страницу:

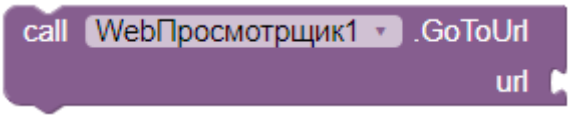


Рисунок 219. Вызов процедуры перехода по указанному адресу

Практическая часть

Цель работы: ознакомиться с компонентой WebПросмотрщик, создать Web-приложение.

Ход работы

1. Перейти по адресу http://ai2.appinventor.mit.edu/ и запустить среду АИ (при необходимости авторизоваться на сайте «App Inventor»).
2. Через пункт главного меню «Проекты» -> «Начать новый проект ...» создать новый проект под названием Web1.

Необходимо создать Web-приложение из компонент Список и WebПросмотрщик. При выборе элемента из списка, в окне компоненты WebПросмотрщик должна отображаться информация по этому элементу из поисковика Google.

3. Перенести компоненты из раздела «**Интерфейс пользователя**» на экран приложения (Screen1) согласно следующей схеме:

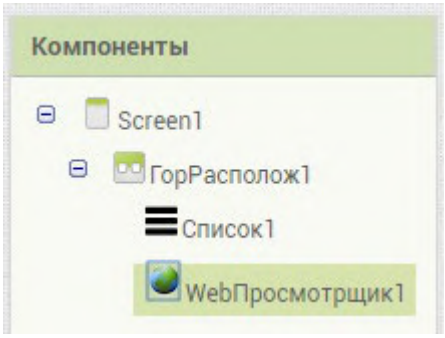


Рисунок 220. Общая схема компонент экрана

4. У главного экрана Screen1 установить свойства:
   a. заголовок — Web
   b. ВыровнятьПоГоризонтали — Центр
5. У ГорРасполож1 установить свойства:
   a. ширина — Наполнить родительский
   b. высота — Наполнить родительский
6. У Список1 установить свойства:
   a. ширина — 30 percent
   b. высота — Наполнить родительский
   c. ЦветТекста — черный
   d. ЦветФона — белый
7. У WebПросмотрщик1 установить свойства:
   a. ширина — Наполнить родительский
   b. высота — Наполнить родительский
   c. ДомашняяСтраница — https://www.google.com?q

8. Убедиться, что финальная схема компонент соответствует следующему виду:

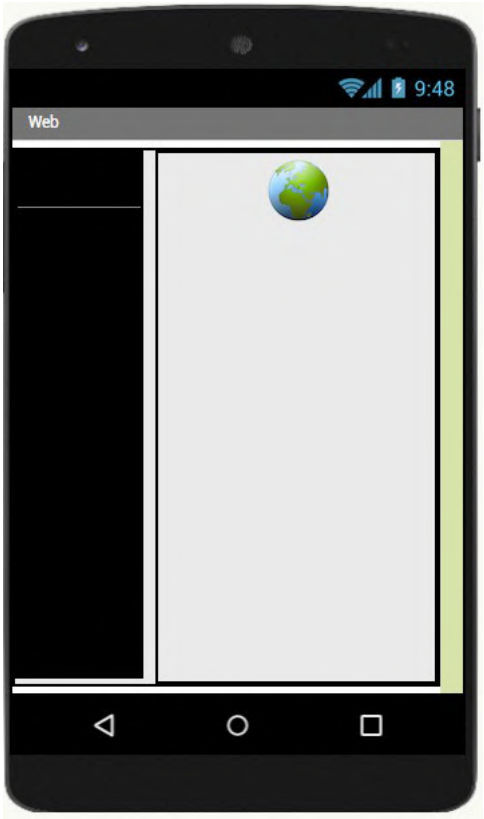


Рисунок 221. Примерная схема дизайна приложения

9. Перейти в режим Блоки.
10. В режиме блоки две основные группы блоков:
    a. для заполнения Список1
    b. для реализации поиска элементов списка в Интернет
11. Для заполнения Список1 создать глобальную переменную zapisi и проинициализировать её списком (массивом):

Рисунок 222. Инициализация массива

12. В обработчике события инициализации экрана присвоить элементам компоненты Список1 значения массива, хранящегося в переменной zapisi.

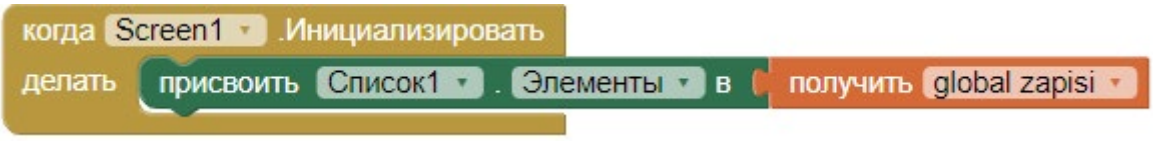


Рисунок 223. Инициализация экрана

13. Реализовать обработчик события выбора элемента списка:

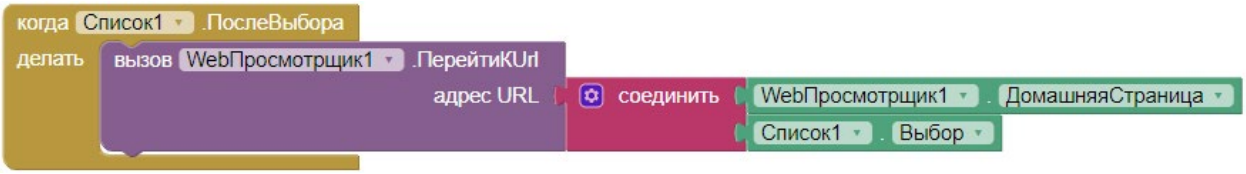


Рисунок 224. Обработка события выбора элемента списка

Поскольку домашней страницей у WebПросмотрщик1 уже указан поисковик Google с началом строки запроса («/search?q=»), то остаётся только склеить текст с адресом домашней страницы и выбранным в компоненте Список1 текстовым элементом.

14. Запустить эмулятор, проверить работу приложения.
15. Перейти в режим Дизайнер и сменить домашнюю страницу на поисковик Яндекс (уточнить правильную форму адреса для дальнейшего склеивания с поисковым словом).
16. Проверить работу приложения в эмуляторе.
17. Добавить внизу экрана кнопки: «Вперед», «Назад», «Домой». Добавить в обработчики нажатий на кнопки процедуры: «call WebПросмотрщик1.goBack», «call WebПросмотрщик1.goForward» и «call WebПросмотрщик1.goHome» и реализовать соответствующие функции перехода по ссылкам.
18. Проверить работу приложения в эмуляторе.

Выводы: в данной лабораторной работе создаётся простое приложение для поиска слов в сети Интернет по заранее заданному списку.

Контрольные вопросы:

Для чего нужна компонента WebПросмотрщик?

Что означает свойство ДомашняяСтраница компоненты WebПросмотрщик?

Лабораторная работа 10. Переводчик

Теоретическая часть

В данной работе в том числе используется теоретический материал из дидактических материалов.

С помощью среды АИ можно создавать как одно-, так и многоэкранные приложения. Для этого достаточно нажать на кнопку ДобавитьЭкран в панели меню проекта:

Рисунок 225. Меню проекта среды АИ

И в появившемся окне вписать название нового экрана:

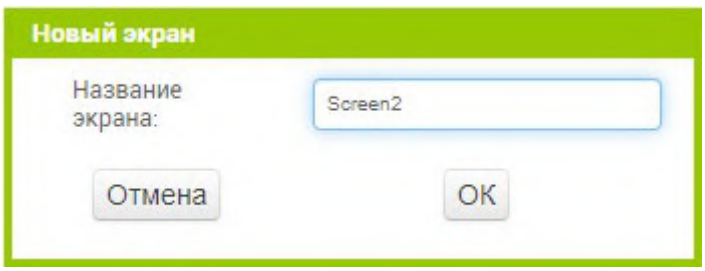


Рисунок 226. Диалоговое окно для названия экрана

Далее выбор экрана можно осуществлять посредством кнопки Screen:

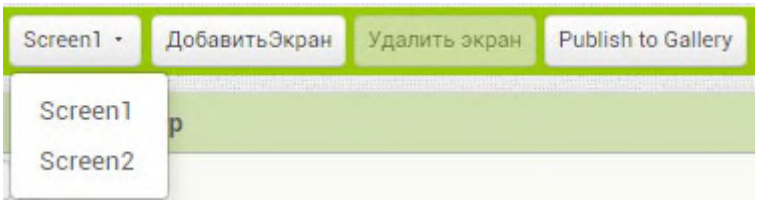


Рисунок 227. Выбор нужного экрана

Управление экранами и передачей данных между ними осуществляется посредством целой группы блоков в разделе встроенных блоков Управление:

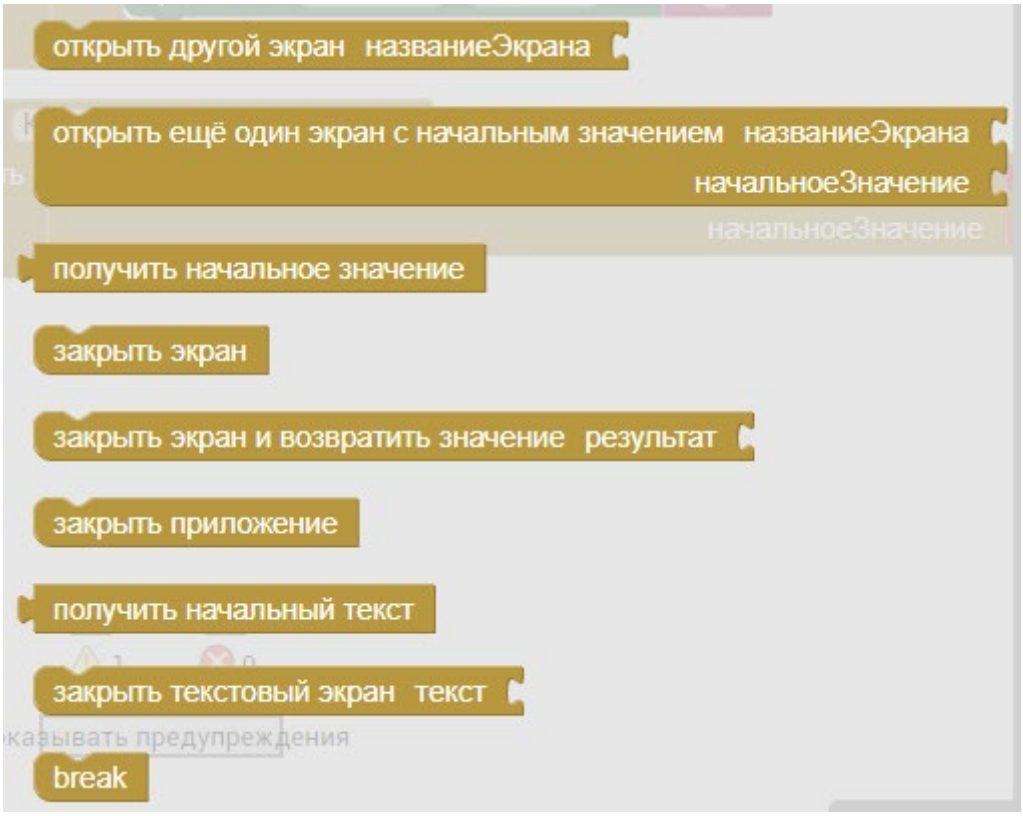


Рисунок 228. Блоки управления экранами

В данной работе используется блок «открыть ещё один экран с начальным значением» для вызова второго экрана и передачи туда значения. В качестве начального значения передается некоторый параметр: текст, число, цвет и т.д. В качестве параметра НазваниеЭкрана передаётся текст с названием нужного экрана.

Также используется блок «получить начальное значение» во втором экране для инициализации переменной и для получения данных, поступивших из главного экрана.

Если необходимо просто открыть другой экран, достаточно использовать блок «открыть другой экран» и добавить к нему справа название нужного экрана.

Дополнительно в данной работе используется компонента Яндекс.Переводчик из раздела «Медиа». Компонента имеет всего четыре программных блока. Здесь используется обработчик события получения перевода текста. Блок содержит две внутренние переменные, возвращающие кодОтвета (равный 200, если перевод успешен) и сам пере-

ведённый текст. Прочие коды ответов API Яндекс.Переводчика можно найти на Яндексе и использовать при анализе работы компоненты.

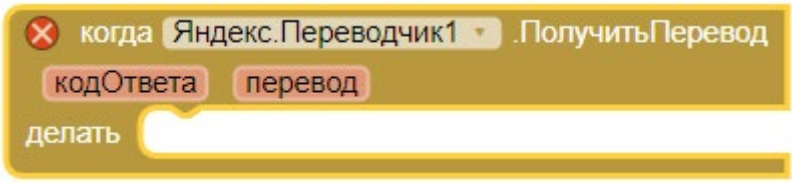


Рисунок 229. Обработчик события получения перевода текста

И второй используемый блок — блок процедуры вызова переводчика. На вход которого подаются параметры: язык перевода (например: en, ru) и переводимый текст.

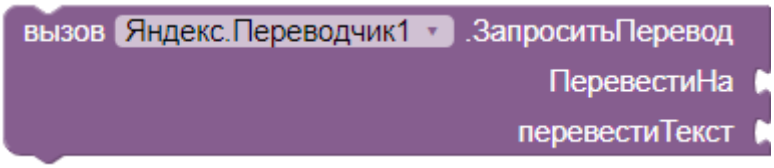


Рисунок 230. Процедура перевода текст компонентой Яндекс.Переводчик

Практическая часть

Цель работы: Освоить работу с несколькими экранами. Создать приложение «Переводчик». Научиться передавать данные между главным и вторым экраном. Освоить работу с компонентой Яндекс.Переводчик.

Ход работы

1. Перейти по адресу http://ai2.appinventor.mit.edu/ и запустить среду АИ (при необходимости авторизоваться на сайте «App Inventor»)
2. Через пункт главного меню «Проекты» -> «Начать новый проект ...» создать новый проект под названием Translator.
3. Перенести компоненты из раздела «Интерфейс пользователя» на экран приложения (Screen1) согласно следующей схеме:

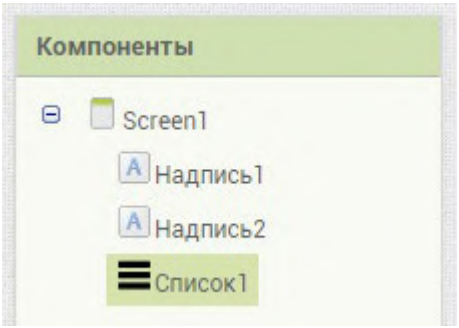


Рисунок 231. Общая схема компонент экрана Screen1

4. У главного экрана Screen1 установить свойства:
   a. заголовок — Translator
   b. ВыровнятьПоГоризонтали — Центр
   c. иконка — указать Рисунок, предварительно загруженный в разделе Медиа
   d. Theme — Device Default

5. У Надпись1 установить свойства:
   a. ВыравниваниеТекста — Центр
   b. РазмерШрифта — 24
6. У Список1 установить свойства:
   a. ширина — Наполнить родительский
   b. ЦветТекста — чёрный
   c. ЦветФона — белый
7. Добавить в приложение второй экран Screen2.
8. Перенести и переименовать компоненты (Надписи, Кнопку и Горизонтальные расположения) из раздела «Интерфейс пользователя» на экран (Screen2) согласно следующей схеме:

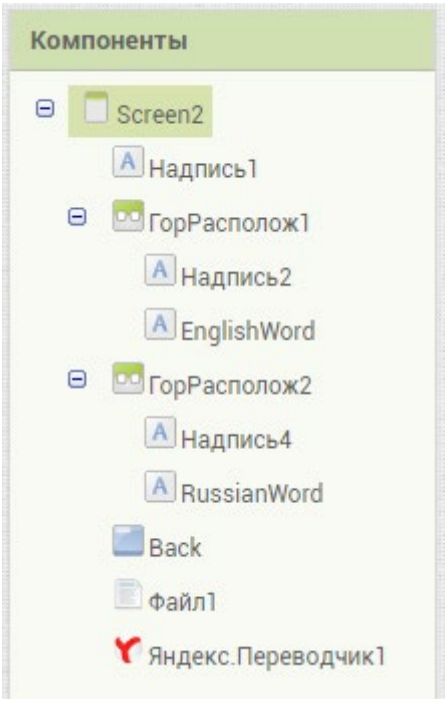


Рисунок 232. Общая схема компонент экрана Screen2

9. У экрана Screen2 установить свойства:
   a. заголовок — Перевод
   b. ВыровнятьПоГоризонтали — Центр
10. У всех надписей установить свойства:
    a. ВыравниваниеТекста — Центр;
    b. РазмерШрифта — 24.
11. У Надпись1 установить свойства:
    a. текст — «Результат перевода:»
12. У надписи EnglishWord установить свойства:
    a. ЦветТекста — чёрный
    b. текст — «Hi!»
13. У надписи RussianWord установить свойства:
    a. ЦветТекста — чёрный
    b. текст — «Привет!»

14. У надписи Надпись2 установить свойство Text — «English:», а у надписи Надпись4 установить свойство Text — «Русский язык:»
15. У горизонтальных расположений установить свойство:
ширина — Заполнить родительский
16. Убедиться, что финальная схема компонент экранов соответствует следующему виду:

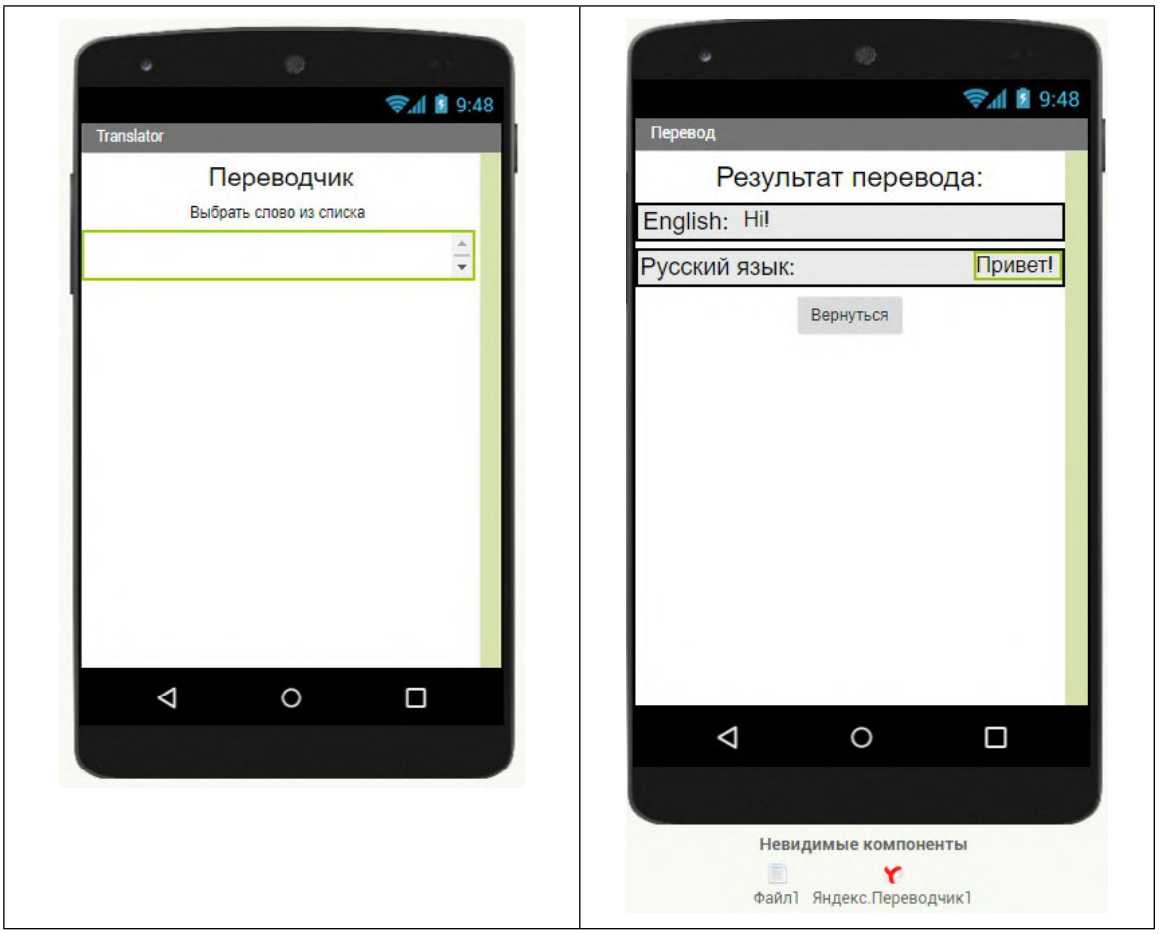


Рисунок 233. Примерная схема дизайна экранов приложения

17. Перейти в режим Блоки для реализации программного кода.
18. Инициализировать компоненту Список1 значениями глобальной переменной список:

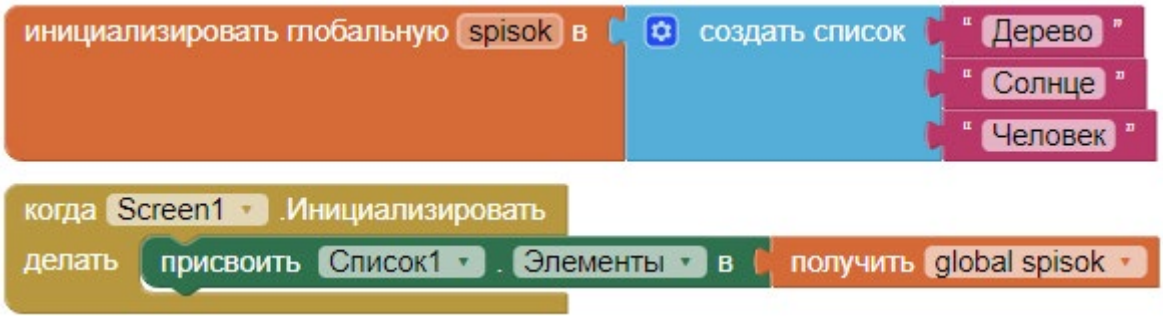


Рисунок 234. Инициализация компоненты Список1

19. Реализовать логику перехода на второй экран после нажатия на элемент в компоненты Список1:

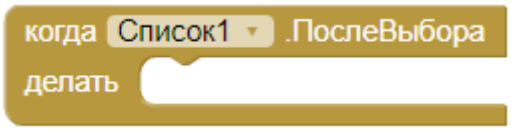


Рисунок 235. Обработчик выбора элементов списка

20. Из раздела Управление добавить блок «открыть ещё один экран ...», указать название второго экрана и передаваемое значение:

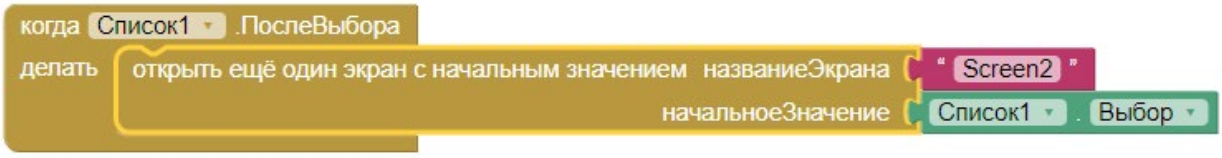


Рисунок 236. Блоки обработчика выбора элементов списка

21. Работа с главным экраном закончена, перейти на второй экран Screen2.

22. Создать глобальную переменную vibor. Перенести в неё полученное из первого экрана начальное значение. Т.е. индекс выбранного в компоненте Список1 элемента:

Рисунок 237. Инициализация переменной vibor значением из первого экрана

23. Реализовать блок инициализации второго экрана, чтобы всё нужные события происходили сразу при его загрузке. А именно, компонента Яндекс.Переводчик переводила бы текст из списка на английский язык:

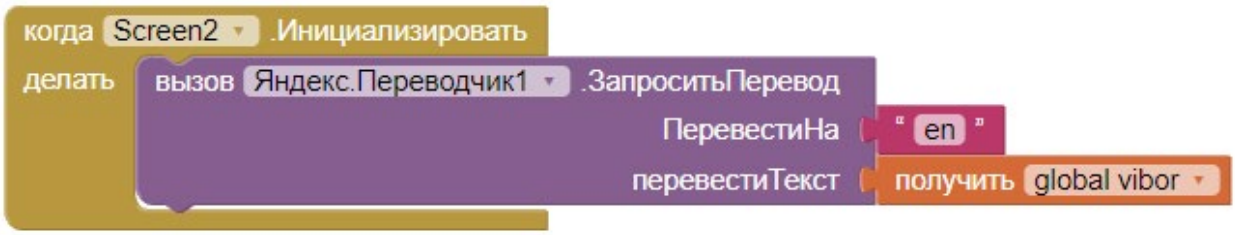


Рисунок 238. Блок инициализации второго экрана

24. Создать обработчик операции перевода и в нём присвоить надписям EnglishWord, RussianWord нужные значения:

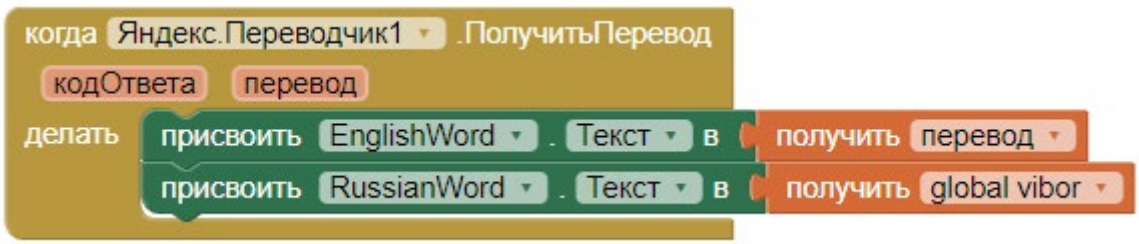


Рисунок 239. Запись результатов в надписи второго экрана

25. Добавить обработчик нажатия на кнопку Back, чтобы приложение возвращалось на первый экран:

Рисунок 240. Настройка кнопки назад мобильного устройства

26. Финальная схема блоков второго экрана приложения приобретёт следующий вид:

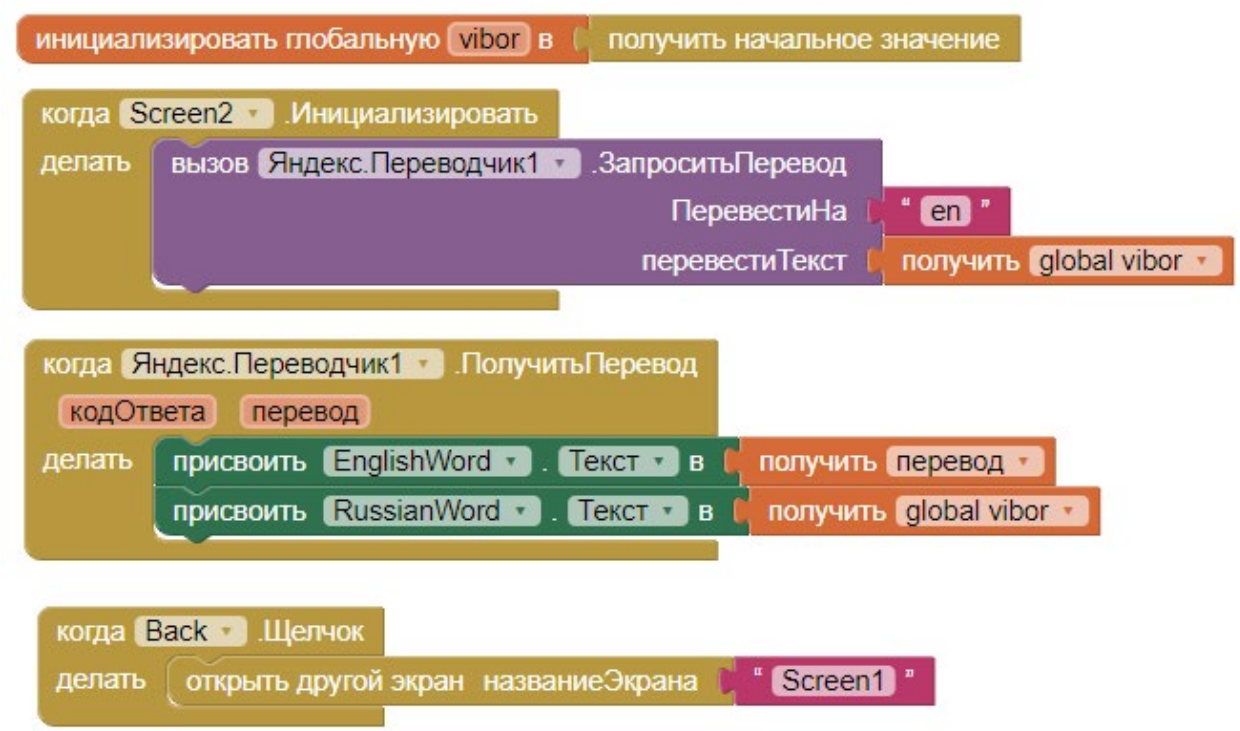


Рисунок 241. Финальная схема блоков второго экрана

27. Запустить эмулятор и проверить работу приложения.

Дополнительные задания:

1. Передать на второй экран не выбранное слово (свойство Выбор компоненты Список1), а индекс выбранного в списке элемента (Свойство ИндексВыбора). В этом случае, на второй экран будет передаваться только численное значение (1, 2, 3 и т. д.). Реализовать код для обработки индекса и организации кода. Необходимо вручную прописать соответствие индексов словам, например, если индекс равен 1, это соответствует слову «Дерево».
2. Организовать хранение исходного списка слов в файле. Добавить инициализацию списка на первом экране словами из файла. Использовать компоненту Файл из раздела Хранилище:

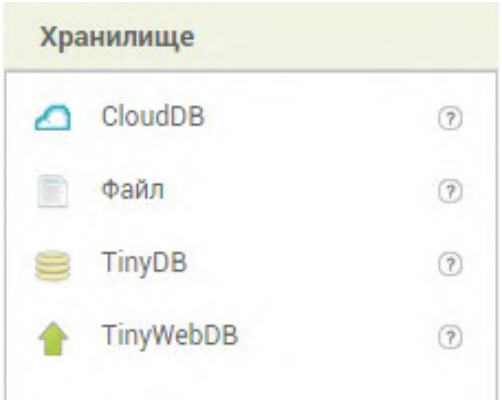


Рисунок 242. Компонента Файл

Достаточно использовать в обработчике кнопки Back два основных блока для сохранения и считывания текста из компоненты Файл1.

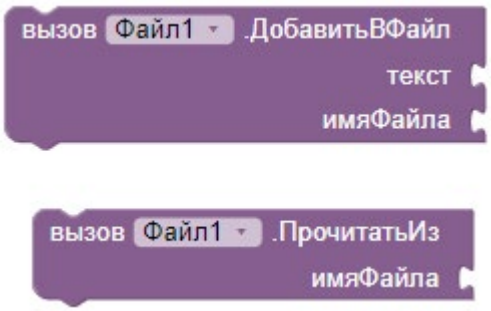


Рисунок 243. Блоки компоненты Файл1.

3. Удлинить список до 15 слов.
4. Реализовать возможность добавления новых слов на первом экране. Для этого добавить текстовое поле, куда будут вводиться новые слова и кнопку, в обработчике которой будет происходить сохранение нового слова в файл.
5. Усовершенствовать дизайн приложения: добавить вспомогательные компоненты, поменять размеры шрифтов, цвета и т.д.
6. Сделать переводчик с нескольких языков на русский. Для этого добавить компоненту ВыборИзСписка (или ИндикаторОжидания), в которых будет храниться список исходных языков. Соответственно, надо предусмотреть заполнение основного списка словами соответствующего языка, т.е. в зависимости от результата выбора в компоненте ВыборИзСписка. Для простоты реализовать данный вариант без компоненты Файл, с обычной инициализацией компонент при загрузке экрана.

Вывод: в данной лабораторной работе создаётся многоэкранное приложение с возможностью передачи данных между экранами.

Контрольные вопросы:

На каком экране можно менять иконку приложении?

Какой экран нельзя удалить?

Какие коды ответов имеются у компоненты Яндекс.Переводчик?

Как при помощи блоков реализовать переход на другой экран?

Лабораторная работа 11. Записная книжка

Теоретическая часть

В данной работе в том числе используется теоретический материал из дидактических материалов и из предыдущей лабораторной работы.

Практическая часть

Цель работы: Освоить на более высоком уровне взаимодействие между экранами. Создать приложение «Записная книжка».

Ход работы

1. Перейти по адресу http://ai2.appinventor.mit.edu/ и запустить среду АИ (при необходимости авторизоваться на сайте «App Inventor»)
2. Через пункт главного меню «Проекты» -> «Начать новый проект ...» создать новый проект под названием Web2.

Необходимо создать приложение со следующей логикой работы: главный (первый) экран должен состоять из текстового поля и кнопки. При каждом нажатии на кнопку текст в поля сохраняется в массив. При долгом нажатии на кнопку открывается второй экран, точно такой же по виду и функционалу, как в лабораторной работе № 9. При нажатии на кнопку назад во время нахождения на втором экране происходит возврат на первый экран. При переходах между экранами массив слов сохраняется и может быть далее дополнен на главном экране новыми значениями.

3. Перенести компоненты из раздела «Интерфейс пользователя» на главный экран приложения (Screen1) согласно следующей схеме:

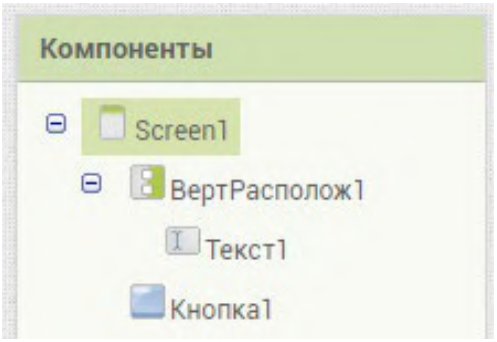


Рисунок 244. Общая схема компонент экрана Screen1

4. У экрана установить свойства:
   a. заголовок — Web
   b. ВыровнятьПоГоризонтали — Центр
5. У компоненты ВертРасполож1 установить свойство:
   ширина — Наполнить родительский
6. У Текст1 установить свойства:
   a. ширина — Наполнить родительский
   b. подсказка — «Записать слово ...»
7. У Кнопка1 установить свойства:
   a. ширина — Наполнить родительский
   b. высота — Наполнить родительский
   c. размер шрифта — 36

8. Убедиться, что финальная схема компонент соответствует следующему виду:

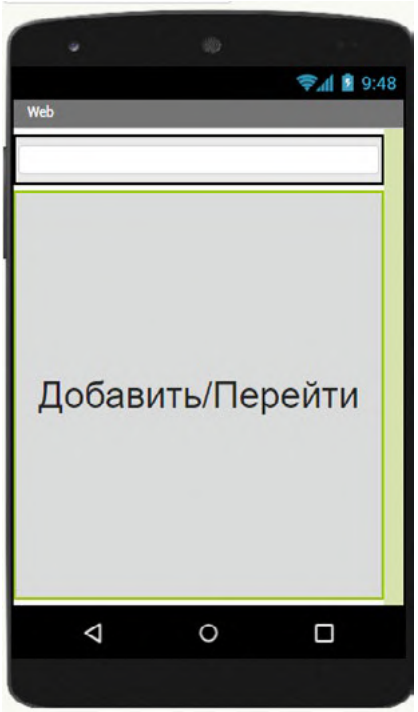


Рисунок 245. Примерная схема дизайна приложения

9. Создать второй экран Screen2.
10. Сделать его полностью идентичным экрану из лабораторной работы 11.

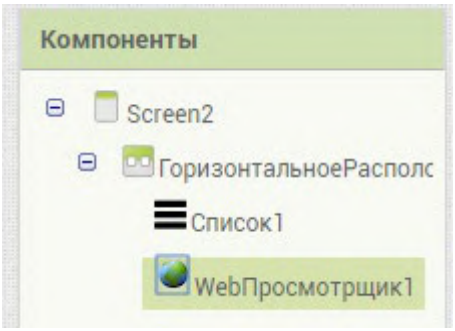


Рисунок 246. Общая схема компонент второго экрана Screen2

11. Переключиться на главный экран Screen1 и перейти в режим Блоки.
12. В режиме Блоки главного экрана Screen1 находятся три основные группы блоков:
    a. инициализация массива zapisi
    b. обработка щелчка для компоненты Кнопка1
    c. обработка долгого нажатия на Кнопка1
13. Для сохранения массива при возвращении назад из экрана Screen2 необходимо на главном экране Screen1 добавить блок «получить начальное значение» из раздела Управление:

Рисунок 247. Инициализация переменной zapisi на экране Screen1

14. Обработка щелчка для кнопки реализуется следующим образом:

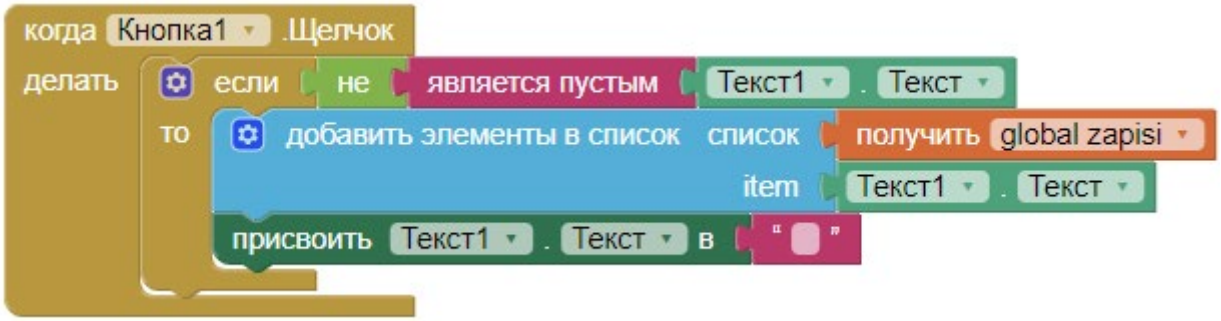


Рисунок 248. Обработчик нажатия на кнопку

Сначала следует проверка логического условия не является ли пустым текстовое поле Текст1. Если оно содержит текст, тогда текст добавляется к элементам массива zapisi. После чего текстовое поле очищается.

15. Обработка долгого нажатия на кнопку реализуется следующим образом:

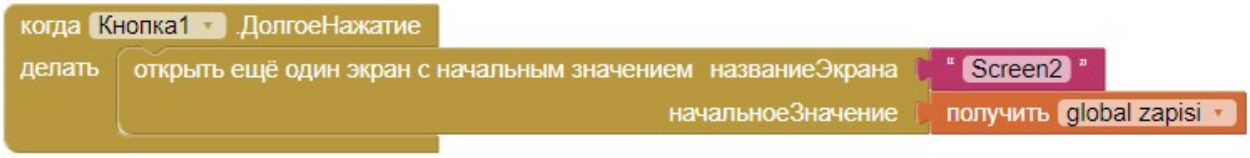


Рисунок 249. Обработчик долгого нажатия на кнопку

В результате происходит переход на второй экран Screen2 и ему передаётся массив zapisi с сохранёнными до этого слов и выражений.

16. В режиме Блоков перейти на второй экран Screen2.

17. Программные блоки второго экрана ничем не отличаются от блоков в лабораторной работе 11, за исключением блока:

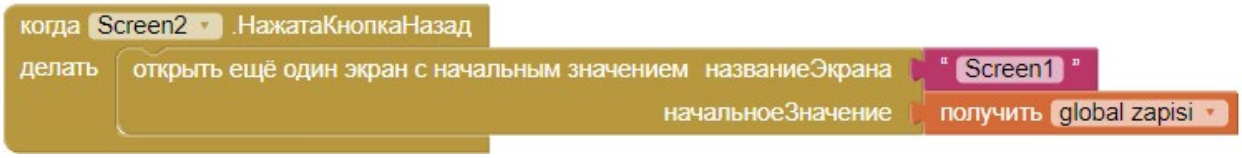


Рисунок 250. Обработчик нажатия на сенсорную кнопку «Назад» мобильного устройства

Данный блок необходим для обратной передачи массива сохранённых пользователем слов на главный экран. Где он сможет далее пополняться.

18. В результате программные блоки второго экрана Screen2 приобретают вид:

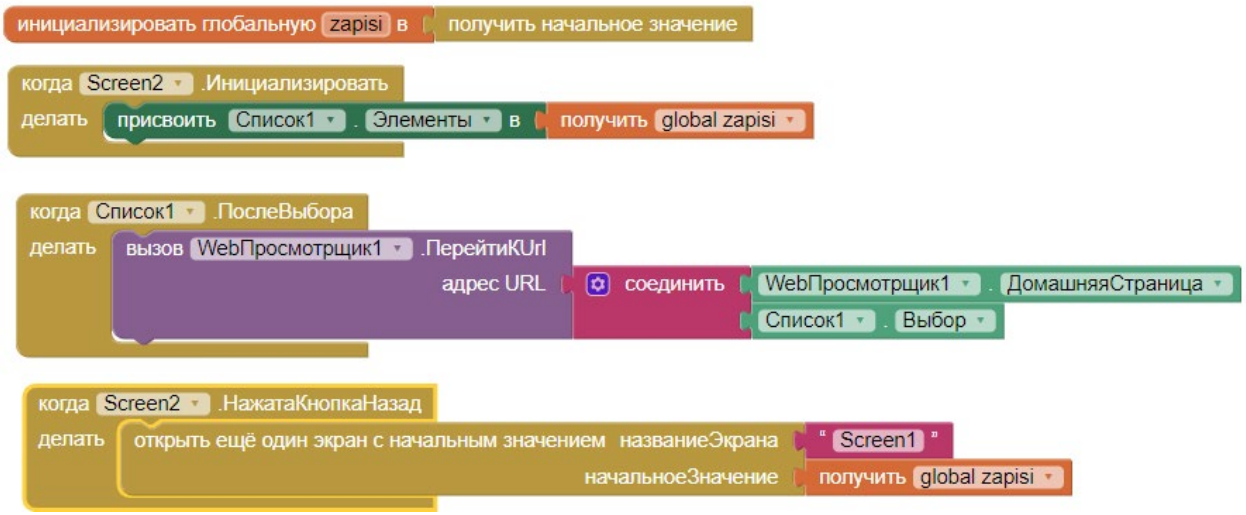


Рисунок 251. Общая схема блоков экрана Screen2

19. Запустить эмулятор, проверить работу приложения.

Дополнительные задания:

1. Осуществить проверку при добавлении выражения в массив (существует ли уже такое выражение в массиве?). Можно использовать блок:

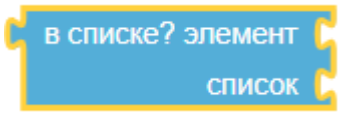


Рисунок 252. Блок поиска элемента в массиве

2. Добавить возможность редактированиz и удаления сохранённых пользователем слов. Для этого можно добавить третий экран для работы с массивом, содержащий компоненты для обработки массива: текстовые поля, надписи, кнопки. Потребуется так же передавать и сохранять массив zapisi в новом экране. При необходимости можно добавить компоненты типа Уведомитель.
3. Реализовать хранение и редактирование массива слов при помощи компоненты Файл

Выводы: в данной лабораторной работе дополнительно прорабатывается создание многоэкранных приложений и взаимодействие между экранами, создаётся приложение для сохранения введённых пользователем слов и поиска их в сети Интернет.

Контрольные вопросы:

Как можно сохранять данные при передаче между экранами?

Как добавлять новые элементы в массив?

Лабораторная работа 12. Переводчик со словарём

Теоретическая часть

В данной работе в том числе используется теоретический материал из дидактических материалов и лабораторной работы №10.

В среде АИ в режиме Блоки среди встроенных разделов имеется раздел Dictionaries (словари), который предоставляет очень удобные инструменты для хранения и управления данными в виде словарей. Словарем называется структура данных, которая хранит данные в виде пар «ключ»-«значение»:

«ключ1»-«значение1»
«ключ2»-«значение2»
«ключ3»-«значение3»

В отличие от массива (списка, не путать с компонентой Список в режиме Дизайнер), который хранит данные в виде «значение»:

«значение1»
«значение2»
«значение3»

В структуре типа «ключ-значение» очень удобно осуществлять поиск элементов по ключу. Все элементы уникальным образом идентифицированы. Как пример, можно представить, что ключи — это уникальные идентификаторы сотрудников предприятия, а значения — это их фамилии. При этом словари в АИ могут в качестве ключей и значений содержать не только текстовые данные, но и цифры, цвета и прочие типы данных.

При этом разные элементы как в словаре, так и в массиве могут принимать одинаковые значения. Например, и ключ2, и ключ 3 могут содержать одну и ту же цифру 5 или текст «Привет».

Раздел Dictionaries содержит большое количество блоков. В данной работе используются следующие блоки:

Блок для создания словаря. В нём указываются попарно ключ — key и значение — value для сохраняемых данных.

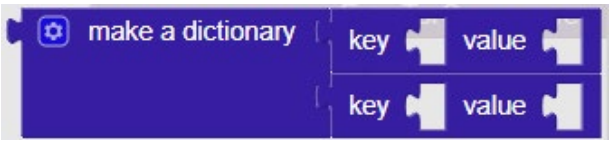


Рисунок 253. Блок для создания словаря

По умолчанию блок содержит две пары ключей и значений. При помощи мутатора количество пар можно изменить:

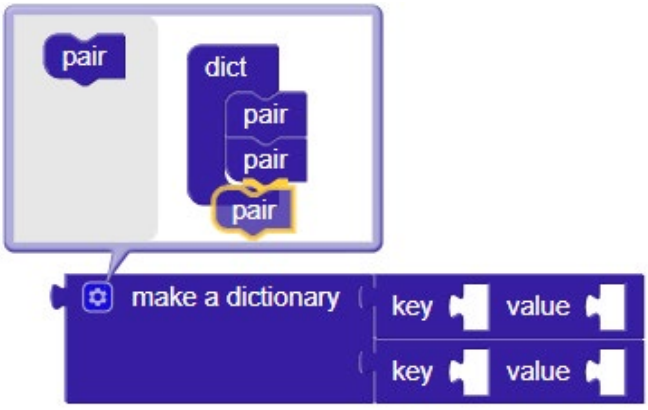


Рисунок 254. Применение мутатора

Блоки «get keys», «get values» используются для получения списков всех ключей или значений словаря.

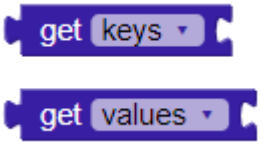


Рисунок 255. Геттеры для ключей и значений

Например:

Рисунок 256. Работа с геттерами

Блок «get value for key» используется для получения значения для конкретного ключа из словаря:

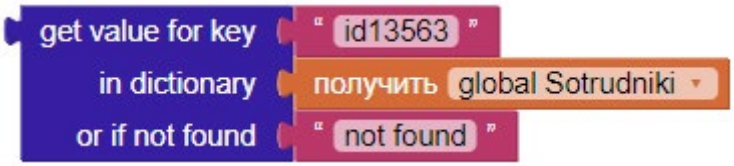


Рисунок 257. Получение значения из словаря по ключу

Также полезными могут оказаться следующие блоки:

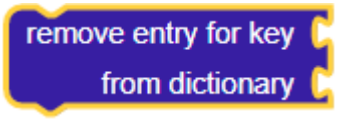


Рисунок 258. Блок удаления пары по ключу из словаря

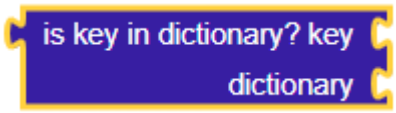


Рисунок 259. Блок проверки существует ли ключ в словаре

Рисунок 260. Является ли структура данных словарем

Рисунок 261. Получает количество элементов в словаре

Практическая часть

Цель работы: Освоить блоки раздела Dictionaries (словарь). Закрепить навыки работы с несколькими экранами.

Ход работы

1. Перейти по адресу http://ai2.appinventor.mit.edu/ и запустить среду АИ (при необходимости авторизоваться на сайте «App Inventor»)
2. Через пункт главного меню «Проекты» -> «Начать новый проект ...» создать новый проект под названием Translator2.

В данной работе приложение строится на основе приложения Переводчик из лабораторной работы №10. В старой версии Переводчика отсутствует словарь, только список исходных слов на одном языке. В данной версии используется словарь, ключи и значения которого передаются между экранами.

В новой версии приложения, чтобы добавить слово, достаточно указать его в словаре. Добавление и удаление новых слов не влияет на общую логику программных блоков. Внешне новое приложение будет выглядеть точно так же, как и приложение в лабораторной работе № 10. Отсутствуют невидимые компоненты Яндекс.Переводчик и Файл.

3. Создать интерфейс приложения с двумя экранами полностью идентичный интерфейсу из лабораторной работы № 9.
4. Очистить у компоненты Список1 свойство ЭлементыИзЦепочки.
5. Перейти в режим Блоки для реализации программного кода.
6. Реализовать следующую логику работы первого экрана (Screen1) приложения:

Инициализировать переменную slovar и заполнить ее нужными значениями английских слов и их переводов. При необходимости использовать мутатор и увеличить количество слов в словаре:

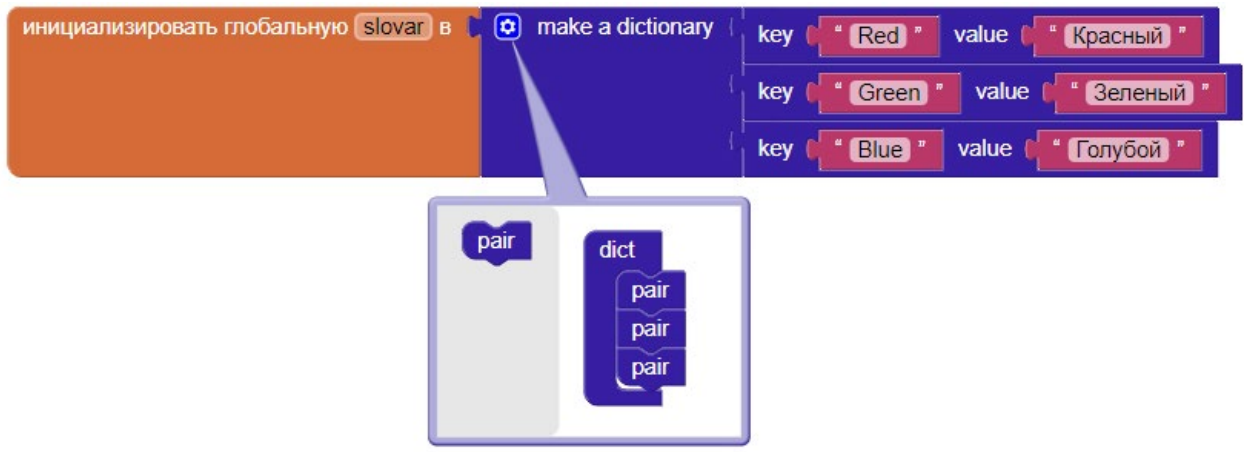


Рисунок 262. Инициализация словаря

Инициализировать переменную spisok (для дальнейшего хранения массива из пары слов) и процедуру для формирования выражения, состоящего из элементов ключей словаря, разделённых знаком запятой. Потому что для заполнения полей компоненты типа Список необходим подобный формат данных: («Red, Green, Blue»).

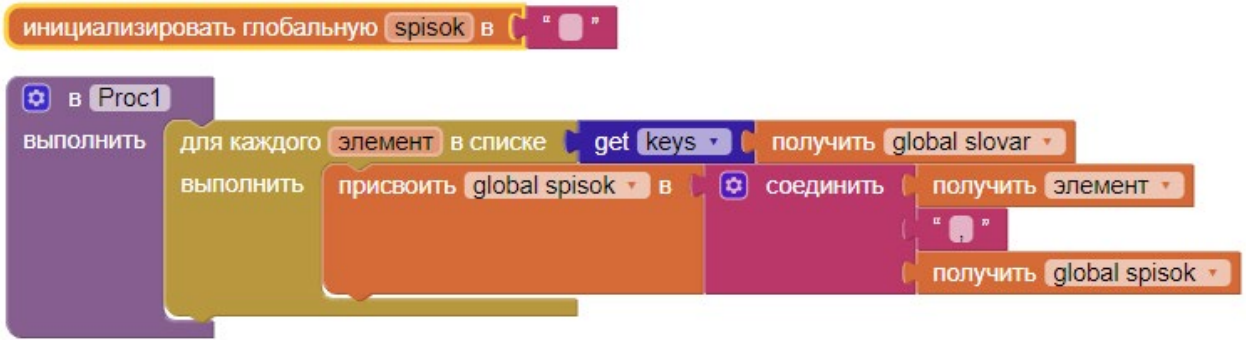


Рисунок 263. Процедура заполнения массива spisok значениями ключей словаря

В процедуре Proc1 переменная spisok заполняется значениями ключей (т.е. английскими словами) из словаря.

Происходит перебор в цикле ключей словаря из переменной slovar. В цикле к каждому ключу добавляется знак «,» и полученное выражение склеивается с текущим содержимым переменной spisok. Таким образом, переменная spisok обновляется и дополняется при каждом проходе списка:

До проходов: Spisok=””

Проход 1: Spisok=”Red, “

Проход 2: Spisok=”Red, Green, “

Проход 3: Spisok=”Red, Green, Blue, “

7. В обработчике события инициализации экрана добавить вызов процедуры Proc1 и установить свойство ЭлементыИзЦепочки компоненты Список1 равным содержимому переменной spisok (т. е. дать команду заполнить Список1 английскими словами):

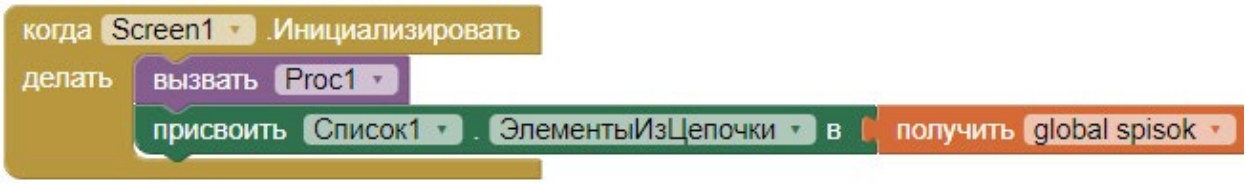


Рисунок 264. Инициализация экрана

8. Усовершенствовать блок обработки выбора элемента из Список1 следующим образом:

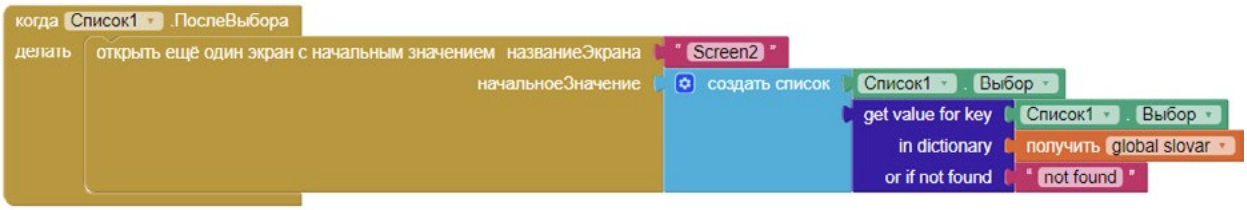


Рисунок 265. Обработчик выбора элемента компоненты Список1

Здесь, по сравнению с лабораторной работой № 10, изменение затронуло только параметр «начальноеЗначение» блока «открыть еще один экран с начальным значением». А именно, в качестве начального значения стал поступать массив из двух слов:

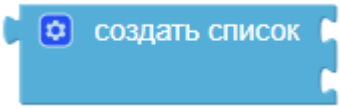


Рисунок 266. Блок создания массива из двух элементов

Первым словом является выбранный пользователем элемент компоненты Список1:

Рисунок 267. Элемент, выбранный из компоненты Список1

Второе слово извлекается из словаря. Куда в качестве ключа поступает первое слово:

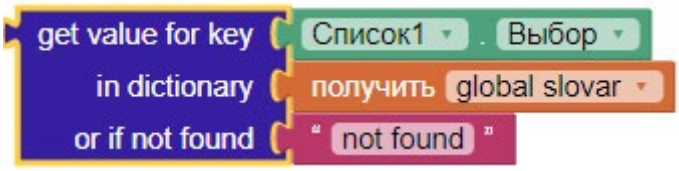


Рисунок 268. Выбор значения из словаря по ключу

В результате, общий вид программных блоков первого экрана имеет примерно следующий вид:

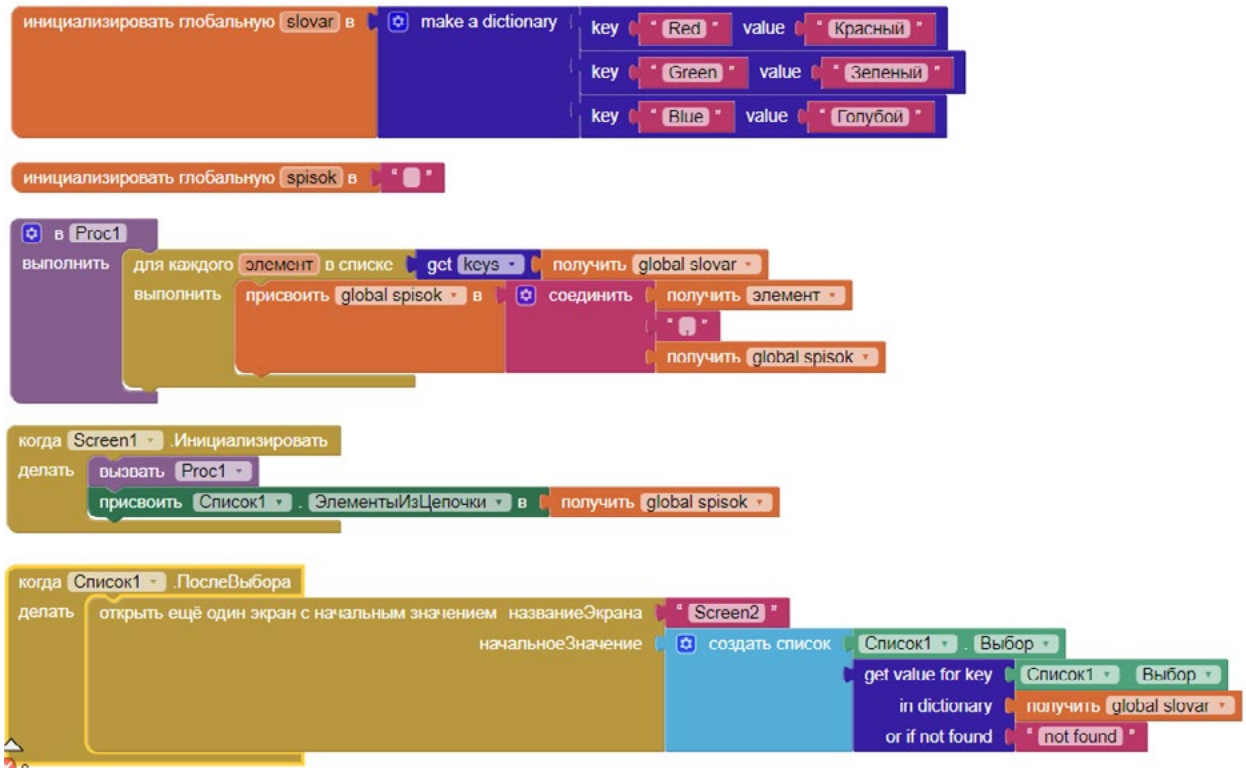


Рисунок 269. Общая схема блоков приложения

Внимание: не путать компоненту Список1 и массив, получаемый при работе голубого блока «создать список». Это совершенно разные вещи. Список1 является компонентом приложения, создаваемым в режиме Дизайнер. А второй является массивом, содержащим значения. Причиной такой путаницы является некорректный перевод на русский язык среды АИ.

9. Переключиться на второй экран.
10. Проинициализировать переменную paraSlov переданным из первого экрана списком. Для этого, как обычно, используется блок «получить начальное значение».

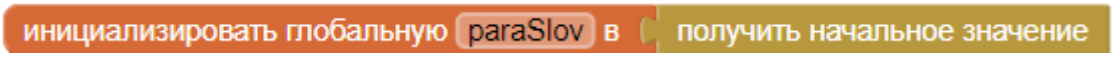


Рисунок 270. Инициализация переменной на втором экране данными из первого экрана

11. Реализовать блок обработки события инициализации (запуска) второго экрана:

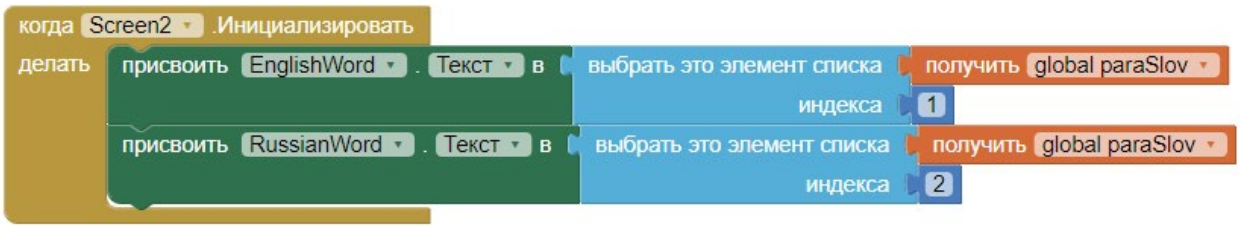


Рисунок 271. Блок обработки инициализации экрана

12. В блоке присвоить надписям EnglishWord и RussianWord значения первого и второго слова соответственно. Для этого используется просто блок «выбрать это элемент списка», в котором указывается список (массив) для выбора и номер индекса.

Рисунок 272. Выбор элемента из массива по индексу

13. Для удобства пользователя, чтобы каждый раз не открывать приложение заново, добавить обработчик события нажатия на кнопку «Назад» (стрелка) мобильного устройства:

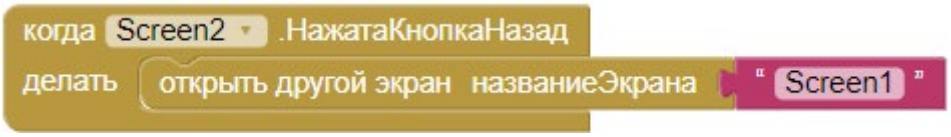


Рисунок 273. Обработчик нажатия на сенсорную кнопку «Назад» мобильного устройства

В результате, при нажатии на сенсорную кнопку «Назад» мобильного устройства, будет осуществляться возврат на главный экран приложения.

14. Финальная схема блоков приложения имеет следующий вид:

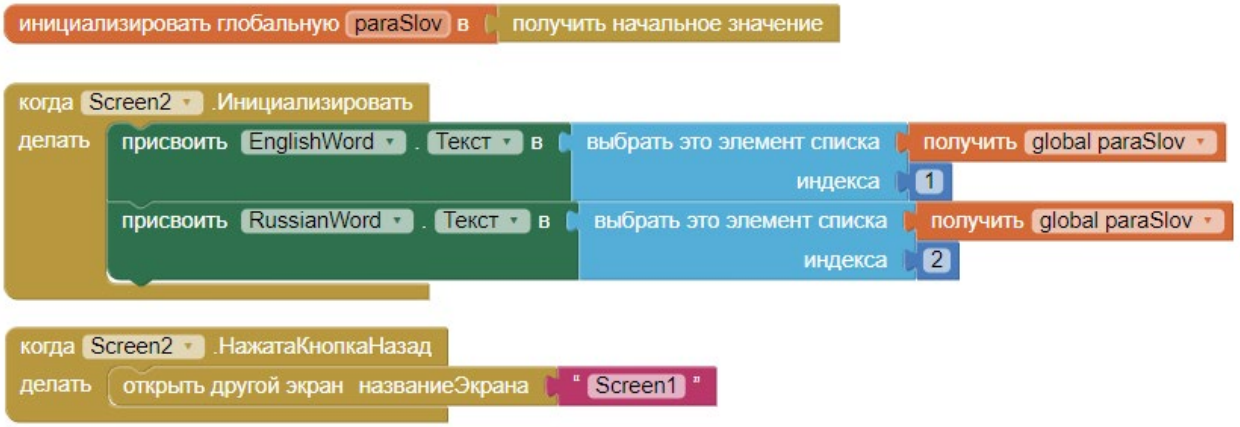


Рисунок 274. Общий вид программных блоков второго экрана приложения.

15. Запустить эмулятор и проверить работу приложения.
16. Добавить новые пары слов в словарь:

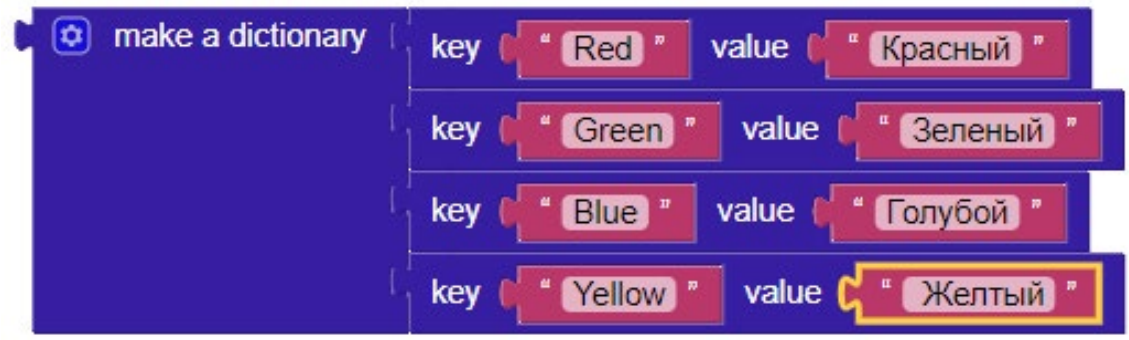


Рисунок 275. Добавление новых пар слов в словарь

17. Проверить работу приложения.

Выводы: новое приложение позволяет добавлять слова без вмешательства в основную логику работы программы.

Контрольные вопросы:

Что такое словарь в среде АИ?

Как работает словарь?

Чем словарь отличается от массива?

Как задать действие для кнопки «назад» мобильного устройства?

Лабораторная работа 13. СМС

Теоретическая часть.

В данной работе в том числе используется теоретический материал из дидактических материалов.

Компонента СенсорМестоположения используется для определения геолокационных координат мобильного устройства. Расположена в разделе Сенсоры. Является невидимой на экране компонентой. Свойства компоненты в режиме Дизайнер имеют технический характер, поэтому их можно вообще не трогать и использовать значения по умолчанию. Свойства необходимые для работы находятся в режиме Блоки.

В режиме Блоки основным обработчиком событий для сенсора является «когда СенсорМестоположения1.Местоположение изменено». Обработчик реагирует на перемещение мобильного устройства в пространстве и имеет 4 внутренних параметра: широта, долгота, высота и скорость. О назначении переменных можно догадаться по их названиям. Для получения доступа к переменным достаточно, как обычно при работе с переменными блоков, ЛКМ кликнуть на оранжевом названии переменной и в появившемся окошке с геттером и сеттером выбрать нужное и использовать в комбинации с другими блоками:

Рисунок 276. Применение внутренней переменной блока

Компонента имеет множество геттеров и сеттеров. Достаточно ознакомиться о основными, например, геттерами свойств:

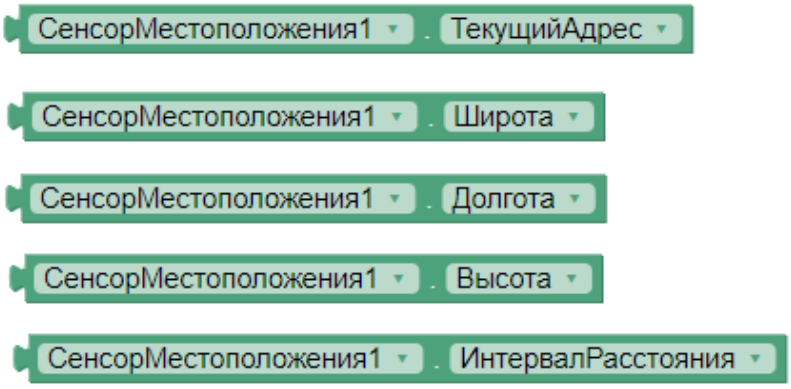


Рисунок 277. Геттеры свойств сенсора местоположения

Каждому геттеру имеется парный ему сеттер. Возвращаемые значения понятны из названий параметров. Кроме ИнтервалРасстояния, параметр, который означает, через сколько метров произойдет событие «местоположение изменено». Например, если данный параметр установить равным 5, это означает, что сенсор просигнализирует о событии «местоположение изменено» только когда мобильное устройство переместится на расстояние не менее чем 5 метров.

Свойство ТекущийАдрес возвращает адрес мобильного устройства. При отсутствии адреса можно воспользоваться свойствами Широта и Долгота.

Компонента Текст (раздела Общение) используется для отправки СМСсообщений и является невидимой компонентой. Свойства в режиме Дизайнер можно не указывать, основная работа с компонентой производится в разделе Блоки.

Количество обработчиков, процедур, геттеров и сеттеров компоненты невелико. Имеется только один обработчик, предназначенный для работы со входящими СМС. Он содержит два внутренних параметра: число (номер с которого пришло сообщение) и текст сообщения.

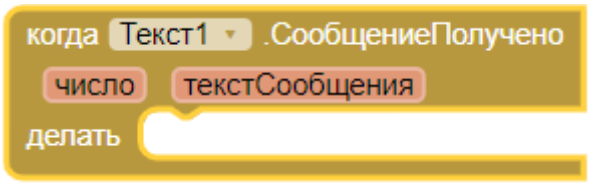


Рисунок 278. Обработчик события поступления смс на мобильное устройство.

Для отправки СМС имеются две процедуры. Первая нуждаются в подтверждении отправки, вторая «SendMessageDirect» отправляет СМС автоматически. Процедуры можно поместить, например, в обработчик события нажатия на кнопку.

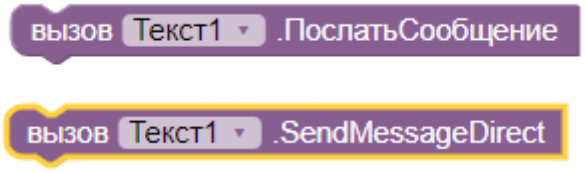


Рисунок 279. Процедуры отправления СМС.

С помощью сеттера «Присвоить Текст1.Телефонный номер» можно указать номер для отправки СМС. Также это свойство можно задать из окна Свойства в режиме Дизайнер при необходимости.

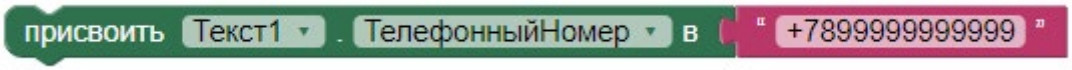


Рисунок 280. Сеттер телефонного номера

Компонента Камера расположена в разделе Медиа и также является невидимой компонентой. Свойств в режиме Дизайнер не имеет. В режиме Блоки представлена всего тремя блоками. Одним обработчиком события, что съёмка произошла, одной процедурой, запускающей фотосъёмку и геттером, возвращающим камеру:

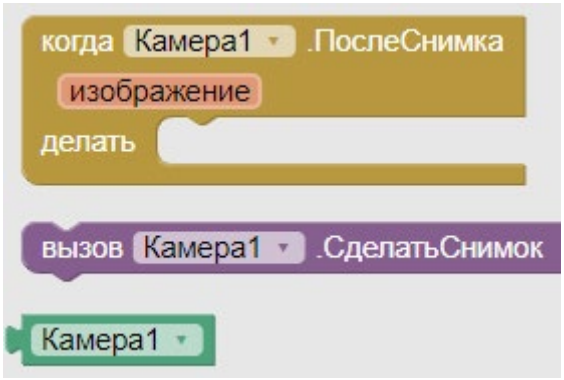


Рисунок 281. Блоки компоненты Камера.

Оранжевый параметр (аргумент) «изображение» в блоке «когда Камера1.ПослеСнимка» даёт доступ к последнему сделанному изображению. Достаточно подвести ЛКМ к параметру и перетащить геттер в нужное место.

Компонента Публикация расположена в разделе Общение и также является невидимой компонентой. Свойств в режиме Дизайнер не имеет. В режиме Блоки представлена всего четырьмя блоками. В данной работе интерес представляют только процедуры. Первая процедура позволяет отправить файл, вторая файл и сообщение, третья только сообщение:

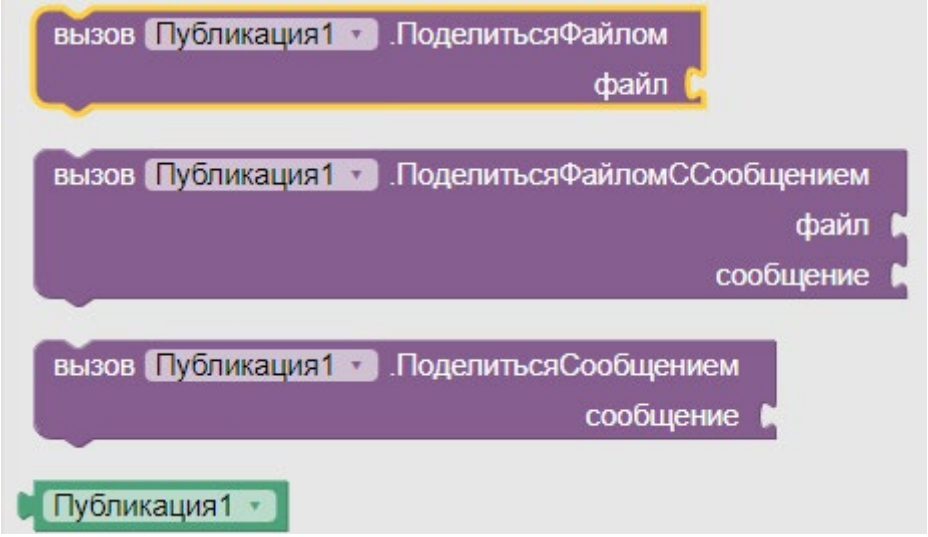


Рисунок 282. Блоки компоненты Публикация.

При отправлении файла или сообщения при помощи компоненты Публикация пользователь должен выбрать во всплывающем окне посредством чего он хочет отправить данные: СМС, WhatsApp, Gmail и т.д.

Практическая часть

Цель работы: Ознакомиться с компонентами СенсорМестоположения, Камера, Публикация, Текст (раздел Общение). Создать приложение для экстренной отправки на заданный номер СМС-сообщение с геолокационными координатами пользователя.

Ход работы

1. Перейти по адресу http://ai2.appinventor.mit.edu/ и запустить среду АИ (при необходимости авторизоваться на сайте «App Inventor»).
2. Через пункт главного меню «Проекты» -> «Начать новый проект ...» создать новый проект под названием SMS.
3. Перенести компоненты из раздела интерфейс пользователя на экран приложения (Screen1) согласно следующей схеме:

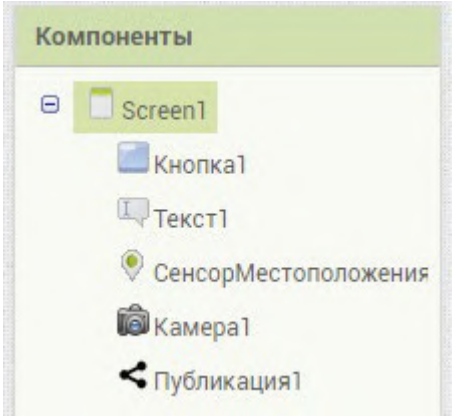


Рисунок 283. Схема компонент экрана

4. Для главного экрана Screen1 установить свойства:
   а. заголовок — СМС
5. Для Кнопка1 установить свойства:

a. ширина — Наполнить родительский
b. высота — Наполнить родительский
c. текст — СМС

6. Убедиться, что финальная схема компонент соответствует следующему виду:

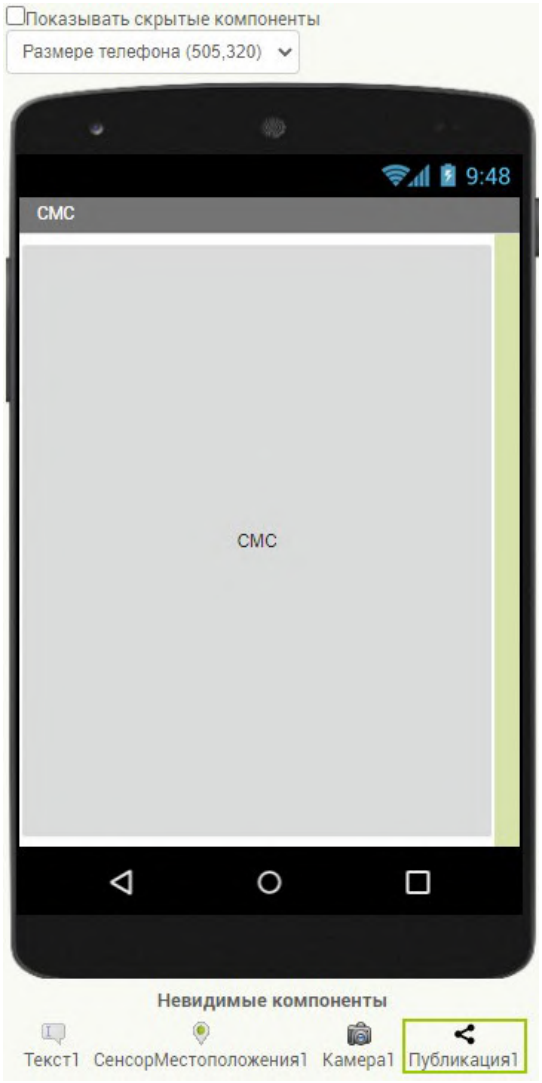


Рисунок 284. Примерная схема дизайна приложения

7. Перейти в режим Блоки.
8. В режиме блоки первого экрана находятся три основные группы блоков:
   a. блоки инициализации номер телефона и включения сенсора местоположения
   b. блоки обработки нажатия на кнопку
   c. блоки обработки события съемки камерой
9. Инициализировать номер телефона, на который будут отправляться СМС. Включить сенсор местоположения. Добавить посредством процедуры AskForPermission экрана запрос на разрешение для отправлений СМС приложением (произойдёт один раз при первом запуске приложения):

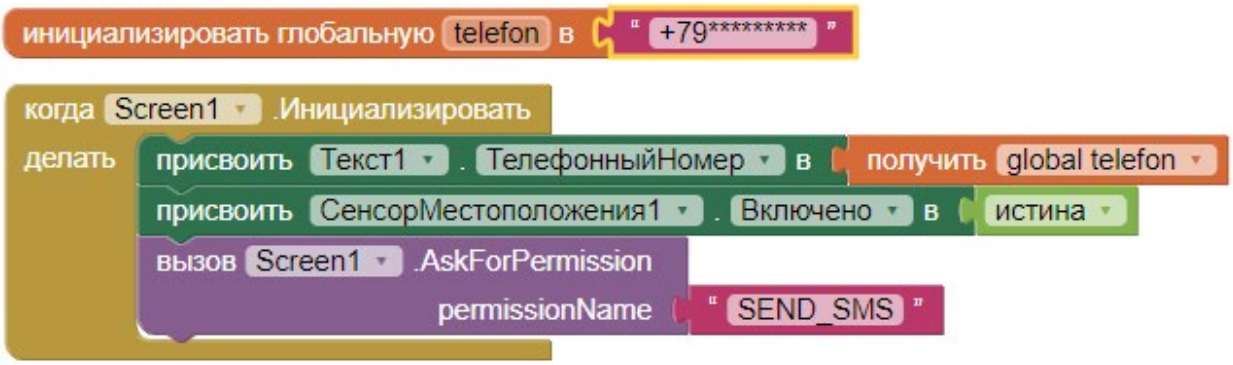


Рисунок 285. Обработчик инициализации экрана

10. Создать обработчик события нажатия на компоненту Кнопка1:

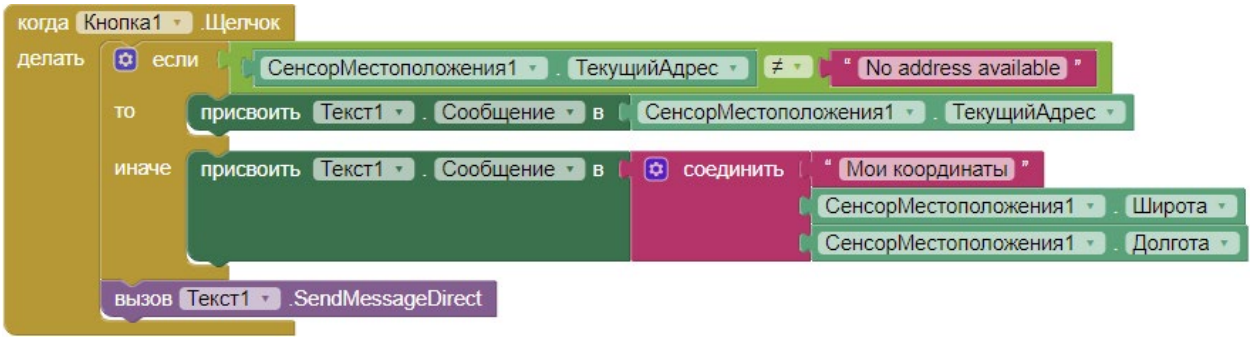


Рисунок 286. Обработчик нажатия на кнопку

Если невозможно определить адрес, то в **прямом** СМС-сообщении передаются текущая широта и долгота мобильного устройства.

11. Создать дополнительный обработчик события для долгого нажатия на компоненту Кнопка1 и вызвать внутри него процедуру «вызов Камера1.СделатьСнимок»:

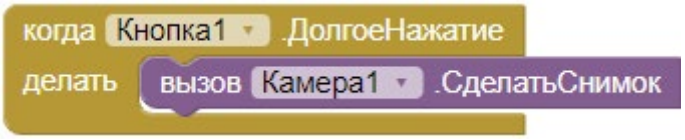


Рисунок 287. Обработчик события «долгое нажатие» на кнопку

12. Создать обработчик для результата съёмки камерой. Вызвать меню «Поделиться ...» для передачи снимка и координат местоположения съёмки:

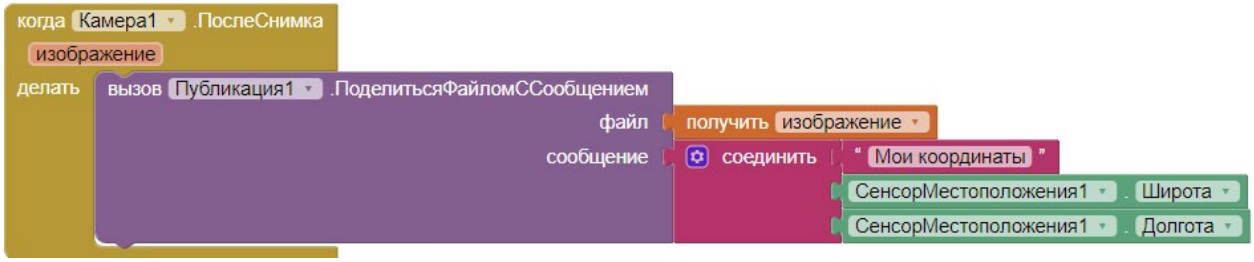


Рисунок 288. Обработчик результатов съемки

13. Проверить работу приложения с помощью эмулятора или в режиме USB не получится из-за ограничений на отправку СМС. Поэтому необходимо построить Приложение (сохранив его в виде apk или сгенерировав QR-код), загрузить на мобильное устройство и запустить. Предварительно необходимо разрешить установку приложений от сторонних разработчиков.

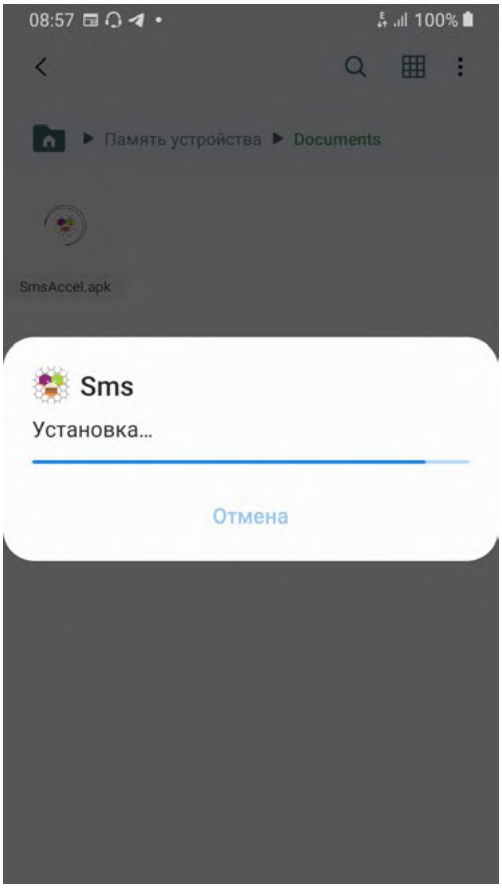


Рисунок 289. Установка приложения

14. Добавить в режиме Дизайнер компоненту СенсорАкселерометра1.

15. Добавить возможность отправления СМС за счёт потряхивания телефона. Для этого вместо обработчика нажатия на компоненту Кнопка1 добавить обработчик для акселерометра, который срабатывает, когда пользователь трясёт мобильное устройство.

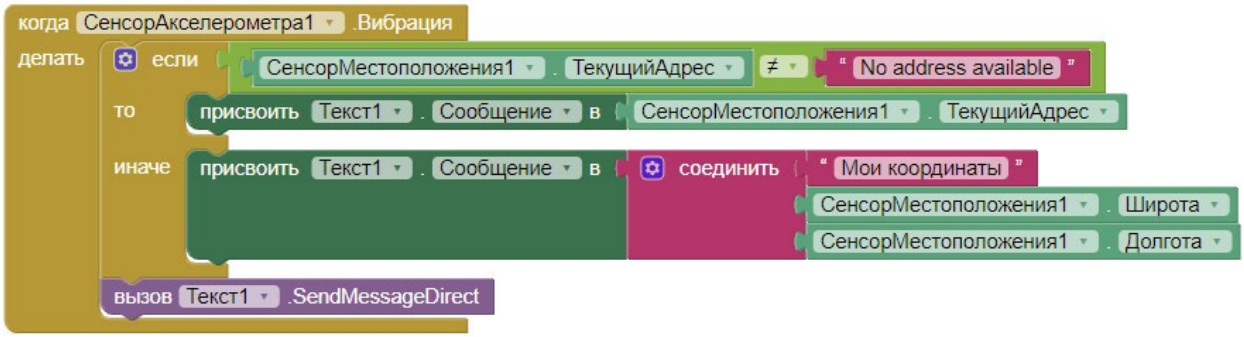


Рисунок 290. Обработчик сенсора акселерометра

16. При необходимости можно изменить значения свойства МинимальныйИнтервал акселерометра, чтобы увеличить или уменьшить чувствительность сенсора к тряске.

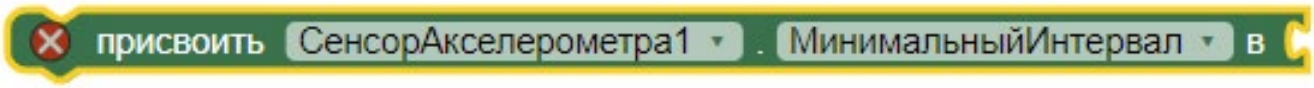


Рисунок 291. Сеттер свойства МинимальныйИнтервал

17. Портировать приложение снова на мобильное устройство и проверить работу.

Дополнительные задания:

1. Добавить возможность изменения номера телефона для отправки СМС через настройки приложения. Для этого добавить второй экран с нужными компонентами интерфейса пользователя (например, текстовое поле, надпись, кнопка). Реализовать переход на второй экран либо через отдельную кнопку, либо через сенсор акселерометра. Например, если потряхивание мобильного устройства происходит только в плоскости xOy. Для этого можно использовать обработчик «когда СенсорАкселерометра1.УскорениеИзменилось».

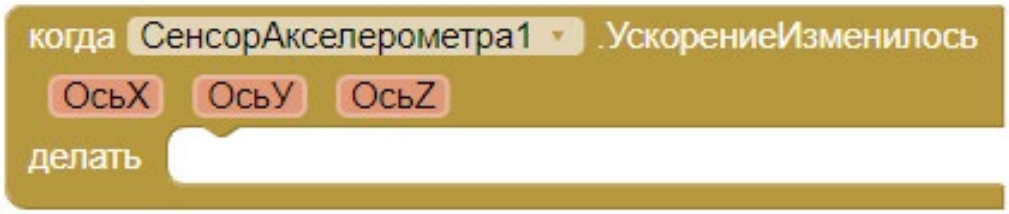


Рисунок 292. Обработчик события изменения ускорений мобильного устройства по координатным осям

Переход на экран настроек можно реализовать и более простым способом, посредством компонент раздела Интерфейс Пользователя. Например, с помощью компонент ГоризонтальноеРасположение и ВыборИзСписка сделать меню в верхней части экрана приложения. Для компоненты ВыборИзСписка при желании можно задать фоновый Рисунок в виде шестеренки.

Установить для компоненты ГорРасполож1 свойства:

a. высота — 7 percent
b. ширина — Наполнить родительский

Для компоненты ВыборИзСписка1:

c. высота — Наполнить родительский
d. ширина — 25 percent
e. текст — «...»
f. жирный — включено

Управляя видимостью компоненты расположения можно скрывать и показывать меню на экране.

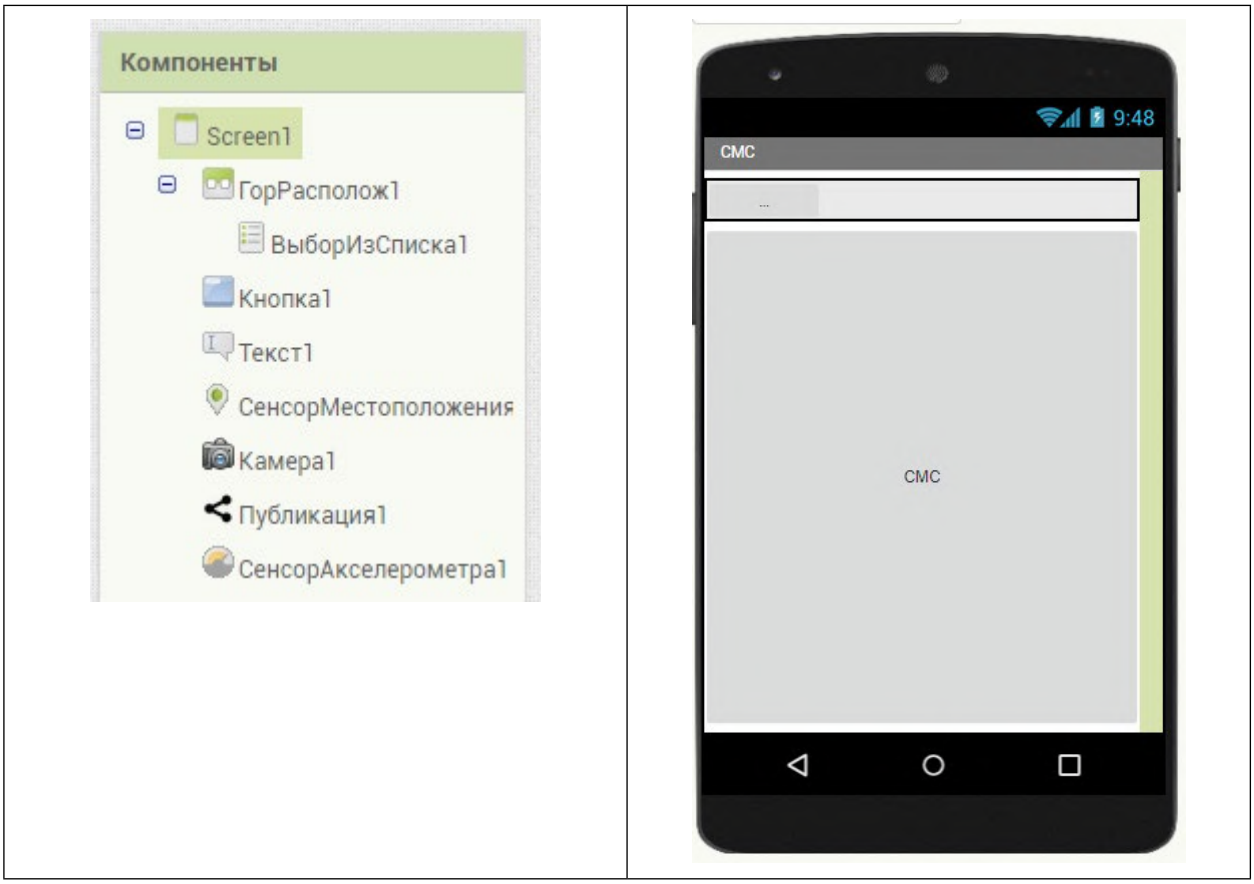


Рисунок 293. Примерная схема дизайна меню

Выводы: в данной лабораторной работе создаётся приложение для отправки экстренных СМС-сообщений с местоположением пользователя.

Контрольные вопросы:

Для чего нужна компонента Публикация?

Как передать снимок компоненты Камера?

Для чего нужен акселерометр?

Лабораторная работа 14. Работа с хранилищем

Теоретическая часть

В данной работе в том числе используется теоретический материал из дидактических материалов.

Если в качестве места хранения данных использовать компоненты типа Список, или переменные в виде массивов и словарей, то при завершении работы с приложением и выгрузке его из памяти все эти данные стираются и пропадают. Поэтому остаётся два варианта — либо никогда не выгружать приложение, всегда держать его в качестве фонового приложения, что неудобно и трудноосуществимо, либо использовать некоторое хранилище данных. В таких случаях хорошо подходят локальные хранилища данных, т.е. хранилища, расположенные в памяти мобильного устройства. Например, легковесная, ориентированная на работу с документа база данных TinyDB[6].

Для использования этой базы данных в разделе Хранилища расположена компонента TinyDB.

Компонента имеет всего одно свойство Namespace (пространство имен), которое обозначает название xml-файла, в котором будут храниться данные. Проще говоря, это название структуры внутри текущей базы данных, в которой будут храниться данные. При смене пространства имён данные начнут сохраняться и извлекаться из другой структуры. При возврате на предыдущее пространство имен, все процессы сохранения и извлечения данных опять будут осуществляться с данными предыдущего пространства.

Компонента TinyDB является невидимой, т.е. не участвует в дизайне экрана приложения. Вся основная работа с ней производится в режиме Блоки. Например, при помощи следующего сеттера можно сменить пространство имен:

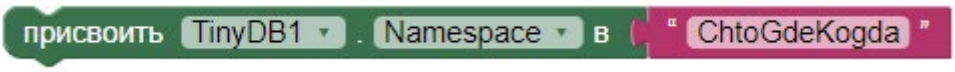


Рисунок 294. Сеттер пространства имен компоненты TinyDB1

В данной работе смена пространства имен не используется.

Компонента TinyDB содержит небольшое количество блоков:

Рисунок 295. Блоки компоненты TinyDB

Здесь используется понятие тег, что означает ключ записи в базе данных. По сути, происходит такая же работа, как со словарями. Так же имеются пары ключ-значение.

Например, при помощи процедуры «вызов TinyDB1.Сохранить значение» можно сохранить в базу данных выражение «Рыжий муравей» с тегом(ключом) «ant»:

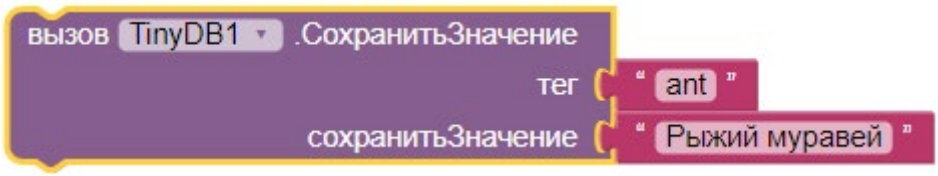


Рисунок 296. Сохранение значения в базе данных по тегу (ключу)

Чтобы получить значение достаточно использовать обратную процедуру «вызов TinyDB1.Получить значение»:

Рисунок 297. Получение значений из базы данных по тегу (ключу)

При помощи процедуры «вызов TinyDB1.получить тэги» можно получить значения всех тегов в текущем пространстве имен компонента TinyDB1 в виде списка, например:

или

Рисунок 298. Получение всех тегов (ключей)

Процедуры «очистить все» и «очистить тег» очищают либо всю базу, либо значение указанного тега соответственно.

Практическая часть

Цель работы: ознакомиться с компонентой TinyDB. Модифицировать приложение Переводчик из лабораторной работы № 10, переместив хранилище из массива в локальную базу данных TinyDB.

Ход работы

1. Перейти по адресу http://ai2.appinventor.mit.edu/ и запустить среду АИ (при необходимости авторизоваться на сайте «App Inventor»)
2. Выбрать в главном меню АИ пункт Проекты — Мои проекты.
3. Открыть проект, созданный в лабораторной работе №1 0.
4. Через пункт главного меню «Проекты» -> «Сохранить проект как ...» сохранить проект под названием Storage.

5. Интерфейс нового проекта идентичен интерфейсу старого проекта, но необходимо добавить невидимую компоненту TinyDB из раздела Хранилище:

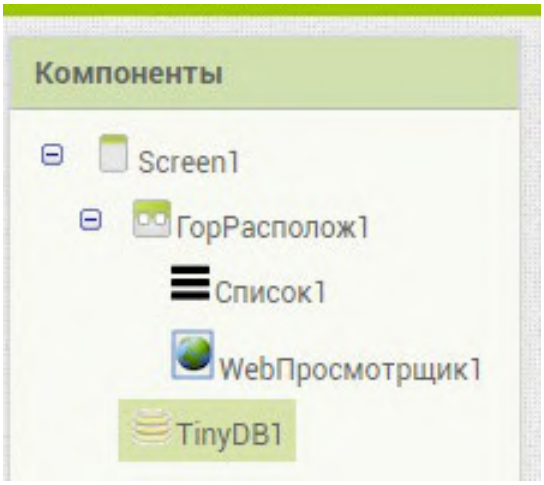


Рисунок 299. Схема компонент экрана

6. Для компоненты TinyDB1 установить свойства:
   a. Namespace — ChtoGdeKogda
7. Перейти в режим Блоки.
8. В режиме блоки создать три основные группы блоков:
   a. процедуру проверки есть ли в хранилище TinyDB1 группа записей с ключом zapisi
   b. обработчик события инициализации экрана для заполнения компоненты Список1
   c. реализация поиска элементов компоненты Список1 в сети Интернет
9. Для реализации процедуры проверки хранилища добавить блок процедуры из раздела Процедуры. Воспользоваться мутатором и создать параметр «tag», в который будет передаваться ключ проверяемой записи:

Рисунок 300. Блок проверки существует ли ключ в базе.

Процедура возвращает логическое значение Истина или Ложь в зависимости от того, существует или нет запись с ключом tag в базе данных. Проверка базы данных происходит за сч`т блока «в списке? элемент» из раздела Массивы.

10. Реализовать обработчик события инициализации экрана следующим образом:

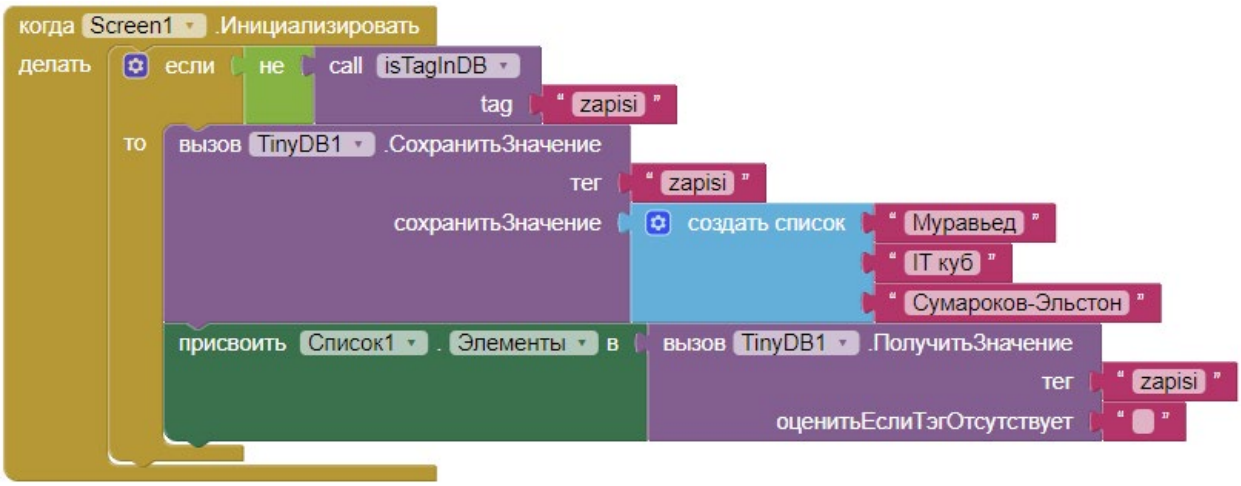


Рисунок 301. Обработчик инициализации экрана

В обработчике события инициализации экрана приложения происходит проверка: если база данных не содержит ключ «zapisi», тогда такой ключ создается и под ним в базе сохраняется массив значений «Муравьед», «IT куб», «Сумароков-Эльстон».

После проверки элементам компоненты Список1 присваиваются значения из базы данных TinyDB1 по ключу «zapisi».

11. Реализовать обработчик выбора элемента компоненты Список1:

Рисунок 302. Обработчик события выбора элемента в компоненте Список1

В данном случае обработчик полностью идентичен обработчику из лабораторной работы №10.

12. Запустить эмулятор, проверить работу приложения.

Дополнительные задания:

1. Добавить возможность редактирования списка. Для этого необходимо извлекать данные из БД в массив, редактировать массив, затем сохранять данные заново. Удобно сразу создать для этого процедуру и затем использовать её в обработчике.

Проверить работу приложения.

Выводы: в данной лабораторной работе реализуется запись и считывание данных в локальную базу посредством компоненты TinyDB.

Контрольные вопросы:

1. Для чего нужны локальные хранилища?
2. С помощью какого блока можно проверить есть ли заданный ключ в хранилище TinyDB?
3. Что такое пространство имен TinyDB?

Примеры конспектов уроков

Тема урока «Работа с компонентами интерфейса и программными блоками в среде АИ»

Тип урока: систематизации знаний.

Цель урока: систематизация и закрепление знаний учащихся по работе со базовыми компонентами интерфейса пользователя, а также базовыми блоками, их особенностями. Закрепление основных приёмов по работе с ними.

Планируемые результаты

Предметные: получение информации о базовых компонентах интерфейса пользователя и их основных свойствах; настройка компонент; выбор и настройка базовых программных блоков; закрепление основных навыков комбинирования блоков.

Метапредметные: умение контролировать и корректировать учебную деятельность, способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные).

Личностные: готовность и способность обучающихся к саморазвитию и личностному самоопределению, сформированность навыков сотрудничества со сверстниками; готовность и способность к образованию, в том числе самообразованию.

Время реализации: 1 академический час

Оборудование и материалы: компьютер, проектор, интерактивная доска.

I. Этап постановки цели и задач урока, мотивации к учебной деятельности — 5 мин.

Деятельность учителя: предлагает учащимся вспомнить работу на предыдущих уроках, с какими компонентами и блоками велась работа? (Напоминает про первый проект, где были задействованы кнопка и уведомитель, блок обработки события нажатия на кнопку и вызов уведомителя).

Деятельность учеников: отвечают на вопросы учителя.

Далее учитель сообщает, что на сегодняшнем уроке будет продолжена работа базовыми компонентами и блоками среды АИ.

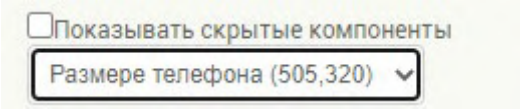
II. Этап актуализации знаний и пробного учебного действия — 8 мин.

Деятельность учителя: предлагает учащимся создать новый проект АИ под названием BasicComponents. Объясняет, что в основе всего находится экран (Screen1) и что все компоненты располагаются на экране. У экрана есть свойства. Чтобы задать свойство любой компоненты и экрана в том числе, необходимо в окне Свойства найти нужное свойство, вписать в текстовое поле нужное значение (или отметить галочку, или выбрать нужное значение из списка) и нажать клавишу Enter, либо кликнуть ЛКМ в любом другом месте. Основные свойства экрана, которые будут необходимы при работе, следующие:

Заголовок — с помощью него можно задать имя приложения, высвечиваемое наверху экрана;

Theme — выбрать одну из предустановленных тем приложения;

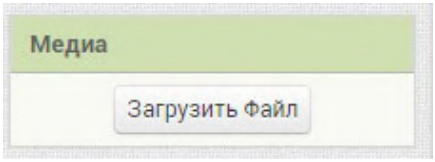
Sizing — Fixed (только для сматрфонов), Responsive (можно выбрать формат экрана — планшет, монитор, смартфон:



(Объясняет, что на большом экране поместится больше компонент. По умолчанию стоит размер телефона (505, 320) и лучше эту размерность не менять).

ВыровнятьПоГоризонтали — выравнивание компонент на экране слева, по центру или справа.

ФоновыйРисунок — задать фон экрана (для этого необходимо подгрузить Рисунок через кнопку «Загрузить файл» в окне Медиа):



Всего у экрана в АИ 24 свойства. Большинство из них используются редко и могут быть освоены самостоятельно в процессе углубленного изучения среды.

Объясняет ученикам, что кроме экрана есть компоненты, посредством которых пользователь наиболее часто взаимодействует с приложением. Большинство из них находится в разделе Интерфейс пользователя. В рамках данного курса будут рассмотрены: Кнопка, Флажок, Изображение, Надпись, ВыборИзСписка, Список, ИндикаторОжидания, Уведомитель, Switch, WebПросмотрщик.

На данном уроке будут рассматриваться Кнопка, Флажок, Надпись, Текст. Кнопка используется для нажатия и вызова какого-либо действия. Надпись отображает текст, т.е. осуществляет вывод информации. Компонент Текст служит для ввода и вывода информации. Флажок является переключателем и может задавать значения свойствам и переменным типа Истина/Ложь.

Дает задание ученикам ЛКМ перетащить на экран текущего проекта эти компоненты. Примерная схема компонент будет следующей:

Акцентирует внимание, что если одна компонента содержит другую, то является родительской по отношению к дочерней компоненте. Дочерняя компонента не может выходить за рамки родительской.

Чтобы удалить любой из компонент надо кликнуть по нему в окне компоненты и нажать кнопку Удалить внизу окна. Кнопка Переименовать позволяет переименовать компоненты. Предлагает несколько раз удалить и заново перетащить компоненты на окно.

III. Этап закрепления нового материала — 16 мин

Деятельность учителя: рассказывает учащимся кратко об основных свойствах выбранных компонент. У них всех есть следующие общие свойства:

Текст — задает текст, который компонента. У кнопки по умолчанию текст «Текст для Кнопка1», у надписи «Текст для Надпись1» и т.д. Если его не поменять в окне Свойства, то именно он будет отображаться в приложении.

Предлагает учащимся поменять тексты для всех компонент на более осмысленные.

Высота и Ширина — одни из самых главных свойств, с помощью них можно регулировать положение компонент на экране. Есть 4 варианта этих свойств: Автоматический (система сама регулирует размеры, в зависимости от других компонент и прочих факторов), «Наполнить родительский» - в этом случае размеры компонента заполняют родительский компонент в котором содержатся. Например, если таким образом указать ширину, то компоненты расширятся до ширины родительского. Варианты Pixels, Percents позволяют задать размеры компоненты в пикселях или процентах от родительского компонента.

Предлагает учащимся поменять ширину до родительского у надписи и текста. Также попробовать поменять размеры у других компонент, затем вернуть значения по умолчанию.

Свойства РазмерШрифта, ЖирныйШрифт, КурсивныйШрифт, ЦветФона, ЦветТекста также являются общими.

Кроме Надписи у всех остальных компонент имеется свойство Включено, которое обозначает, что компонент доступен и с ним можно работать.

У всех компонент кроме Флажка имеется также свойство ВыравниваниеТекста с вариантами: Слева, Центр, Справа.

Особым свойством именно компоненты Текст является ТолькоЦифры. Если отмечена галочка с этим свойством, то в Текст можно вводить только цифры.

Дает задание применить различные значения свойств для компонент в окне Свойства. Проанализировать изменения внешнего вида компонент.

Далее объясняет, что все свойства можно задавать как вручную в окне Свойства, так и через блоки в режиме Блоки. Такие блоки называется сеттеры. Блоки, с помощью которых можно считать свойства называются сеттеры.

Кроме блоков, устанавливающих и считывающих различные свойства компонент, имеются блоки, вызывающие процедуры, вызывающие обработчики событий, связанных с действиями над компонентами.

Рассказывает про блоки обработчики событий:

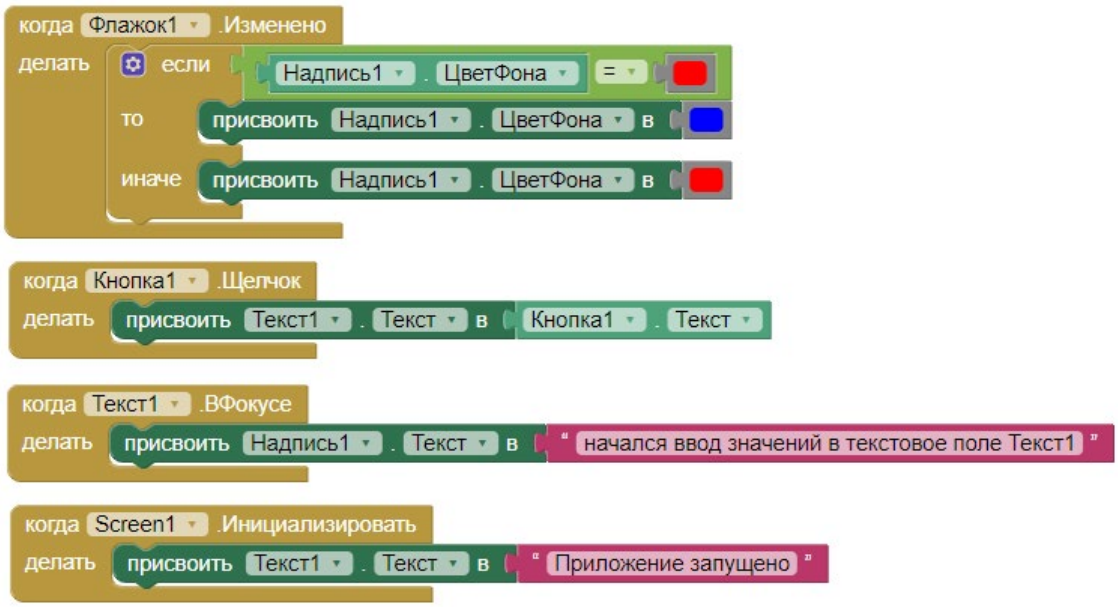
Блок вида «когда Кнопка1.Щелчок» обрабатывающий событие нажатия на кнопку;

Блок вида «когда Флажок1.Изменено» обрабатывающий событие изменения состояния флажка (галочка помечается или снимается);

Блок вида «когда Текст1.ВФокусе» обрабатывающий событие начала ввода текста в текстовом поле;

Блок инициализации вида «когда Screen1.Инициализировать», выполняющий команды при запуске приложения.

Предлагает учащимся реализовать следующую схему программных блоков для созданных в начале урока компонент. Затем запустить эмулятор и проверить работу программы:



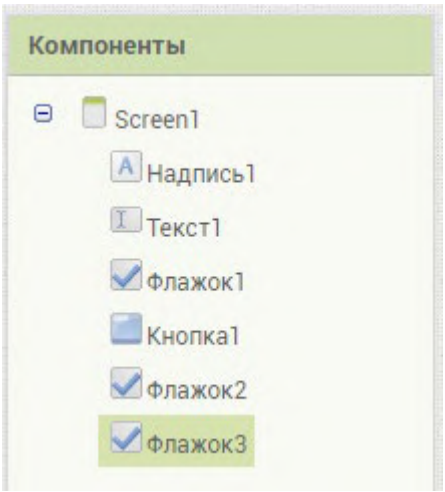
Предлагает обсудить блоки и проанализировать, что происходит. Рассказывает про коричневый условный блок «если .. то иначе» из раздела Управление, используемый для создания ветвления. Акцентирует внимание на зелёном блоке проверки равенства из раздела Логика.

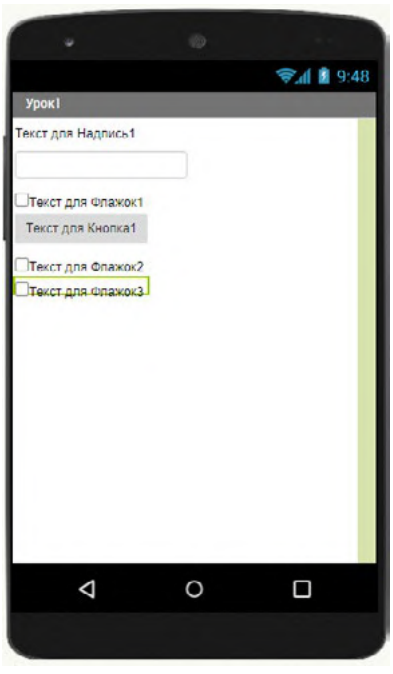
Объясняет, что блоки с цветами и текстом перетаскиваются ЛКМ из разделов Цвета и Текст соответственно.

Деятельность учащихся: разбирают логику работы приложения.

IV. Этап проверки понимания и первичного закрепления — 8 мин

Деятельность учителя: предлагает учащимся дополнить имеющуюся схему компонент и добавить компоненты Флажок2 и Флажок3 внизу экрана.

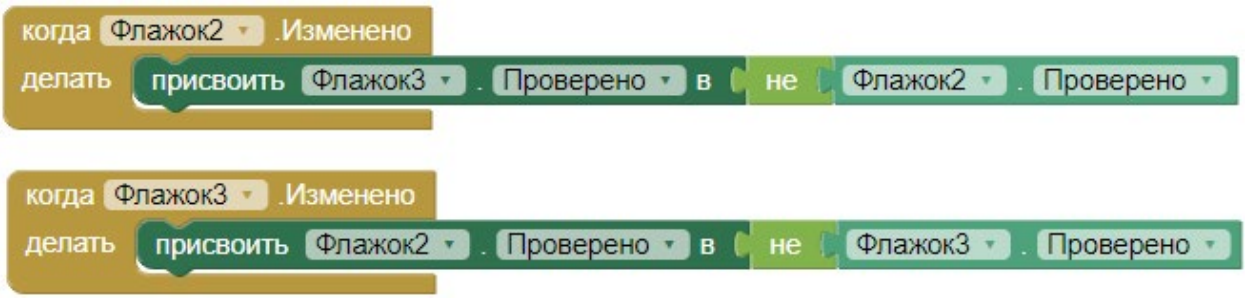




Тексты Флажков (как и прочих компонент) учащиеся меняют и настраивают самостоятельно под свои предпочтения и вкусы.

Далее необходимо реализовать взаимное переключение флажков друг другом. Если один помечается, то с другого пометка снимается и наоборот.

Для этого необходимо добавить блоки:



Деятельность учащихся: разбирают использование логического НЕ при изменении состоянии Флажков.

Деятельность учеников: создают комбинацию блоков, анализируют, запускают приложение в эмуляторе, делают выводы.

V. Этап контроля усвоения и коррекции ошибок — 4 мин

Деятельность учителя: предлагает учащимся ответить на следующие вопросы:

1) Какой блок надо перетащить в окно Просмотр чтобы реализовать обработку события нажатия Кнопки?
2) Для чего нужна компонента Флажок?
3) Какие есть общие свойства у компонент Надпись, Текст и Кнопка?
4) Как свойство надо задать, чтобы в текстовое поле Текст можно было вводить только цифры?

VI. Информация о домашнем задании, инструктаж по его выполнению — 2 мин.

Деятельность учителя: сообщает домашнее задание:

1) Самостоятельно изучить компоненту Switch (переключатель) из раздела Интерфейс пользователя, которая имеет точно такое же поведение, как у флажка. Только если у флажка, когда он отмечен, меняется свойство «Проверено», то у переключателя меняется свойство «on».
2) Модифицировать созданное на уроке приложение, добавив еще одно текстовое поле Текст2 (в любое место экрана) и компонент Флажок2 или Switch1. Реализовать следующую логику: если флажок (или Switch) отмечен, то при нажатии на кнопку текст из текстового поля Текст1 копируется в поле Текст2, иначе ничего не происходит.

VII. Этап рефлексии деятельности на уроке — 2 мин.

Деятельность учителя: предлагает учащимся оценить свою деятельность на уроке, ответив на следующие вопросы:

1) Что нового вы узнали сегодня на уроке?
2) Считаю ли я свою сегодняшнюю работу на уроке успешной?
3) Поняли ли они многослойную концепцию организации приложения в АИ, в которой есть слой представления в виде компонент и слой логики приложения в виде комбинаций блоков, причем часть блоков обрабатывают взаимодействие с пользователем, а другие задают прочую логику работы приложения?

Тема урока «Работа с компонентами интерфейса и программными блоками в среде АИ»

Тип урока: систематизации знаний.

Цель урока: систематизация и закрепление знаний учащихся по работе со компонентами типа расположение, а также базовыми блоками, их особенностями. Закрепление основных приемов по работе с ними.

Планируемые результаты

Предметные: получение информации о компонентах из раздела Расположения и их основных свойствах; настройка компонент; выбор и настройка базовых программных блоков; закрепление основных навыков комбинирования блоков.

Метапредметные: умение контролировать и корректировать учебную деятельность, способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные).

Личностные: готовность и способность обучающихся к саморазвитию и личностному самоопределению, сформированность навыков сотрудничества со сверстниками; готовность и способность к образованию, в том числе самообразованию.

Время реализации: 1 академический час.

Оборудование и материалы: компьютер, проектор, интерактивная доска.

Этап постановки цели и задач урока, мотивации к учебной деятельности — 5 мин.

Деятельность учителя: предлагает учащимся вспомнить работу на предыдущих уроках, с какими компонентами данных велась работа?

Деятельность учеников: отвечают на вопросы учителя.

Далее учитель сообщает, что раньше они просто добавляли компоненты на экран друг за другом, а с помощью новых компонент ГоризонтальноеРасположение и ВертикальноеРасположение их можно группировать вместе и создавать более удобный интерфейс приложения. При этом управление расположением и группировкой компонент на экране происходит автоматически, что очень удобно.

II. Этап актуализации знаний и пробного учебного действия — 8 мин.

Деятельность учителя: предлагает учащимся открыть проект BasicComponents прошлого урока, сохранить его как новый проект BasicComponents2 через пункт меню Проекты — Сохранить как ... и в режиме Дизайнер посмотреть расположение компонент на экране. И ответить на следующие вопросы:

Красиво ли выглядит взаимное расположение компонент?

Какие компоненты можно поместить не друг под другом, а на одной строке?

Деятельность учеников: отвечают на вопросы учителя.

Предлагает ученикам на листе бумаги нарисовать примерный дизайн компонент.

Совместно обсуждают полученные варианты.

Объясняет, что обычные компоненты, например, из раздела Интерфейс пользователя перетаскиваются ЛКМ в компоненты типа Расположение. Тем самым они группируются внутри расположений. Если это горизонтальное расположение, то это будет группировка в строку, если вертикальное, то в столбец. В режиме Блоки у этих компонент нет обработчиков действий, только сеттеры и геттеры. Без лишней необходимости нет смысла их модифицировать. Предполагается, что в большинстве случаев достаточно их настроить вручную посредством окна Свойства.

III. Этап закрепления нового материала — 16 мин.

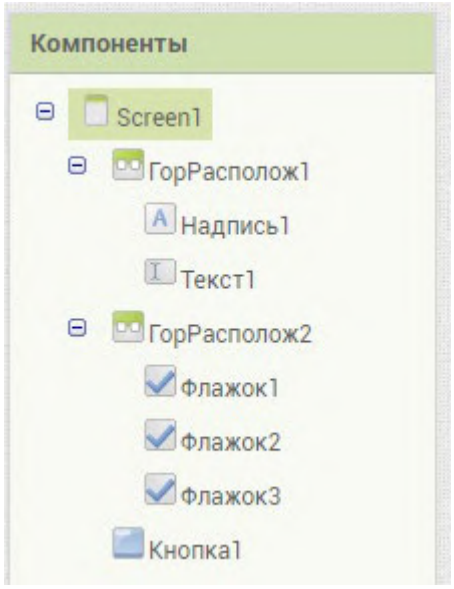
Рассказывает про свойства компонент ГоризонтальноеРасположение и ВертикальноеРасположение, которые у них абсолютно одинаковые:

ВыровнятьПоГоризонтали, ВыровнятьПоВертикали, то же самое, что и у экрана Screen1. И отвечает за выравнивание компонент, находящихся внутри расположения.

Высота и Ширина — точно такие же и с такими же вариантами, как и у всех компонент, рассмотренных на предыдущем уроке.

Обычно, для красивого внешнего вида имеет смысл оба эти свойства установить как «Наполнить родительский», чтобы расположения точно вписывались в экран. Соответственно и компоненты внутри них тоже (если у них так же установлены эти же свойства) будут вписываться в родительские компоненты типа Расположение, а, следовательно, и в экран приложения.

Предлагает модифицировать интерфейс экрана в текущем приложении и добавить расположения на экран. Затем в эти расположения перетащить нужные компоненты экрана. Для этого надо ЛКМ аккуратно перетащить нужную компоненту и наложить на компоненту типа Расположение. Следующие компоненты так же. Затем настроить свойства расположений и компонент внутри них. Примерное решение этой задачи может иметь следующий вид:



Компоненты расположения переименованы в ГорРасполож1 и 2, так как иначе не видны в окне Компоненты.

При этом установлены следующие свойства:

У всех компонент, кроме Надпись1 и Текст1 свойство Ширина — «Наполнить родительский». А у компоненты Кнопка1, так же и Высота — «Наполнить родительский». У компоненты Экран (Screen1) свойство Заголовок — «Урок2».

Внешний вид интерфейса приобретает следующий вид:

Деятельность учеников: реализуют схему компонент, настраивают свойства и создают похожий интерфейс. Запускают эмулятор. Оценивают интерфейс приложения в эмуляторе.

Деятельность учителя: кратко рассказывает о компоненте HorizontalScrollArrangement. Объясняет, что она такая же, как и ГоризонтальноеРасположение, но еще и обладает полосой прокрутки. За счёт этого компоненты внутри неё не сжимаются, чтобы поместиться внутри, а могут иметь любую нужную ширину. Если общая ширина компонент внутри превышает ширину этого вида расположения, то тогда с помощью полосы прокрутки можно добраться до любого нужного компонента.

Предлагает поэкспериментировать с данным видом расположения.

Деятельность учеников: пробуют заменить одно из обычных горизонтальных расположений компонентой HorizontalScrollArrangement. Оценивают интерфейс приложения в эмуляторе.

IV. Этап проверки понимания и первичного закрепления — 8 мин.

Деятельность учителя: предлагает учащимся подумать, а что будет если в одно расположение добавить другое расположение, например, в вертикальное добавить несколько горизонтальных или наоборот?

Деятельность учеников: обсуждают подобную схему.

Деятельность учителя: объясняет, что в этом случае внутренние расположения будут сгруппированы внутри внешнего точно так же, как если бы это были бы обычные компоненты. С помощью такого способа можно также добиться интересных решений в расстановке компонент на экране приложения.

Деятельность учеников: выполняют решение примеров, сравнивают скорость решения, делают выводы.

V. Этап контроля усвоения и коррекции ошибок — 4 мин.

Деятельность учителя: обращает внимание, что в данной схеме названия Флажков слегка прикрываются снизу границей Расположения. Предлагает ученикам самостоятельно поэкспериментировать и установить для расположения ГорРасполож2 нужное значение свойства Высота в процентах.

Деятельность учеников: выполняют задание. Оценивают интерфейс приложения в эмуляторе.

VI. Информация о домашнем задании, инструктаж по его выполнению — 2 мин

Деятельность учителя: сообщает домашнее задание:

Самостоятельно изучить компоненту Табличное расположение.

Добавить ее на экран текущего проекта и поэкспериментировать с компонентами и расположениями.

VII. Этап рефлексии деятельности на уроке — 2 мин.

Деятельность учителя: предлагает учащимся оценить свою деятельность на уроке, ответив на следующие вопросы:

Что нового вы узнали сегодня на уроке?

Считаю ли я свою сегодняшнюю работу на уроке успешной?

Позволят ли новые компоненты создавать более удобные интерфейсы приложений? Или лучше указывать свойства компонент вручную в пикселях и настраивать их внешний вид?

Тема урока «Анимация»

Тип урока: систематизации знаний.

Цель урока: систематизация и закрепление знаний учащихся по работе со компонентами анимации, а также их базовыми блоками, их особенностями. Закрепление основных приемов по работе с ними.

Планируемые результаты

Предметные: получение информации о компонентах из раздела Анимация и их основных свойствах; настройка компонент; выбор и настройка базовых программных блоков; закрепление основных навыков комбинирования блоков.

МетаПредметные: умение контролировать и корректировать учебную деятельность, способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные).

Личностные: готовность и способность обучающихся к саморазвитию и личностному самоопределению, сформированность навыков сотрудничества со сверстниками; готовность и способность к образованию, в том числе самообразованию.

Время реализации: 1 академический час.

Оборудование и материалы: компьютер, проектор, интерактивная доска.

Этап постановки цели и задач урока, мотивации к учебной деятельности — 5 мин.

Деятельность учителя: предлагает учащимся создать новый проект под названием Animation. Затем совместно просмотреть и изучить содержимое раздела «Рисование и Анимация».

Деятельность учеников: создают проект.

Далее учитель сообщает, что с помощью компонент этого раздела можно создавать движущиеся объекты, наделять их текстурой, свойствами, как типовыми, так и такими, как скорость, направление. Можно добавлять обработчики поведения, например, такого как наложение объектов друг на друга, отскакивание от границ области, бросание, вращение и т.д.

II. Этап актуализации знаний и пробного учебного действия — 8 мин.

Деятельность учителя: предлагает ЛКМ перетащить на экран приложения компоненту Холст из раздела «Рисование и анимация». Затем установить для компоненты следующие свойства:

Высота, Ширина — Наполнить родительский;

Фоновый Рисунок — при желании посредством окна медиа ученики могут загрузить понравившийся Рисунок и установить его фоном Холста.

Объясняет ученикам, что компонента Холст является основой, на которой могут «существовать» и двигаться компоненты типа Шар и Спрайт.

Далее перетаскивают компоненту Шар на Холст. Обращает внимание на следующие свойства компоненты Шар:

Радиус — 5 (задает размер);

ЦветКраски — чёрный (задает цвет);

Курс — 0;

Интервал — 100.

Свойство Курс задает Направление в котором движется Шар, значение по умолчанию 0 означает вправо. Если вверх, то это будет 90, влево 180, вниз 270. Итого, варианты от 0 до 360, соответственно градусам окружности.

Свойство Интервал задает такт движения Шара в миллисекундах. Чем меньше значение, тем быстрее он движется.

III. Этап закрепления нового материала — 16 миню

Деятельность учителя: предлагает учащимся добавить ещё один Шар на Холст, затем добавить компоненту Спрайт. Один Шар сделать зелёным, второй синим.

Задать у зелёного Шара свойства:

Скорость — 25;

Курс — 95.

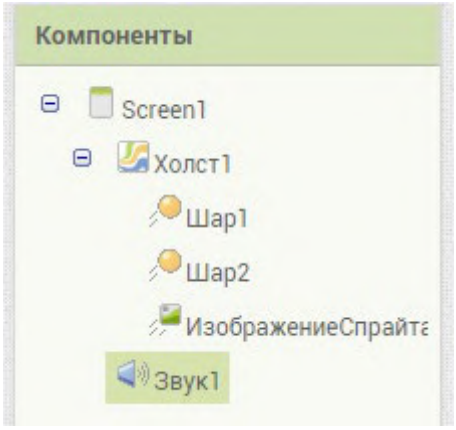
Задать у синего Шара свойства:

Скорость 15;

Курс 30.

Далее добавить компоненту Звук из раздела Медиа. С помощью звука будет производиться эффект вибрации при определенных событиях.

Объясняет, что Спрайт ничем не отличается от Шара, кроме того, что у него есть обычные для всех компонент свойства Высота и Ширина (которых нет у Шара, потому что у него есть радиус). Также у Спрайта есть свойство Изображение, посредством которого Спрайт можно превратить в картинку, например, викинга с мечом или футболиста с мячом. Примерная схема интерфейса приложения должна выглядеть следующим образом:



Далее предлагает перейти в режим Блоки и посмотреть какие блоки имеются у компонент Шар и Спрайт.

Акцентирует внимание, что блоки у Шаров и Спрайта одинаковые и делятся на три группы: обработчики событий (коричневые), процедуры (фиолетовые) и сеттеры с геттерами (зелёные).

Упоминает некоторые базовые обработчики и процедуры Шаров и Спрайтов:

Таблица 1.

Обработчики событий для компонент типа Шар и Спрайт

Обработчик	Событие
когда Шар1.Касание	когда пользователь касается пальцем изображения шара
когда Шар1.Перетащенный	перетаскивание шара его по экрану (здесь 3 пары координат: начальные означают точку где он был перед перетаскиванием, текущие — точку в данный момент, а предыдущие — точку в прошлый такт (интервал))
когда Шар1.ДостигнутКрай	когда шар достиг края экран
когда Шар1.НаложениеС-Объектом	когда шар пересёкся с другим шаром или спрайтом

Предлагает реализовать следующую схему программных блоков:

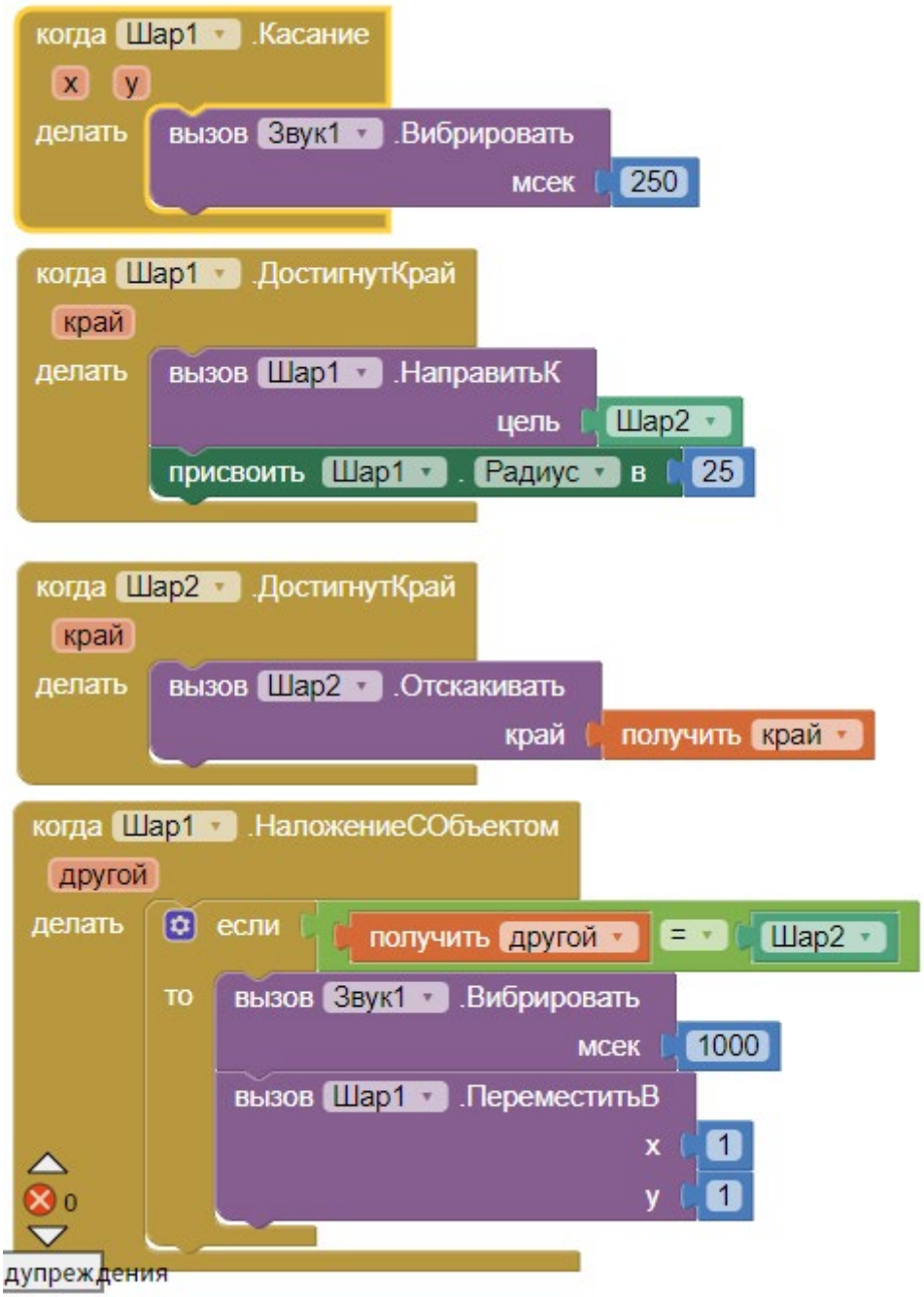
Таблица 2.

Процедуры компонент типа Шар и Спрайт

Процедура	Действие
вызов Шар1.НаправитьК	направляет Шар в сторону другой цели (Шара или Спрайта)
вызов Шар1.ТочкаВНаправлении	направляет Шар в точку с координатами (x,y)

Блоки геттеров и сеттеров типовые, как у многих компонент и позволяют задавать и считывать свойства, те же, что и в режиме Дизайн в окне Свойства.

Предлагает построить и изучить следующую схему программных блоков.



Чтобы получить геттеры вида:

в обработчике события «когда Шар2.ДостигнутКрай» необходимо ЛКМ кликнуть на слове «край»:

И в появившемся контекстном окне выбрать блок «получить край» и вставить его в блок процедуры «вызов Шар2.Отскакивать край»:

Тем самым обозначается, что Шар должен отскочить от того самого края, который вызвал событие ДостигнутКрай.

Деятельность учеников: в режиме Блоки строят аналогичную схему и изучают.

IV. Этап проверки понимания и первичного закрепления — 8 мин.

Деятельность учителя: предлагает учащимся запустить эмулятор и проверить работу приложения.

Деятельность учеников: запускают эмулятор и проверяют работу приложения.

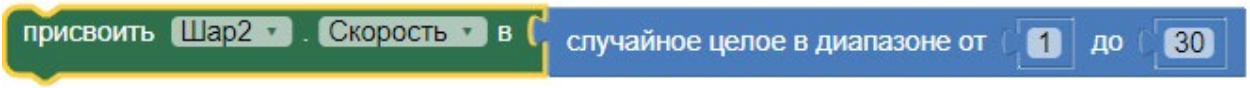
Деятельность учителя: предлагает учащимся поменять свойства шаров (скорость и радиус) в режиме Дизайн в окне Свойства.

Деятельность учеников: меняют свойства и проверяют работу приложения.

Деятельность учителя: предлагает учащимся поменять в режиме Блоки для Шара1 свойство Радиус и координаты x,y процедуры ПереместитьВ.

Деятельность учеников: меняют свойства и проверяют работу приложения.

Деятельность учителя: предлагает в блок «когда Шар2.ДостигнутКрай» добавить следующий сеттер скорости Шара2 (сеттер появляется при нажатии на компоненту Шар2 в разделе Screen1 окна Блоки, а синий блок и цифры перетаскиваются из раздела Математика):



Деятельность учеников: добавляют сеттер и проверяют работу приложения, делают выводы.

V. Этап контроля усвоения и коррекции ошибок — 4 мин.

Деятельность учителя: предлагает учащимся ответить на следующие вопросы:

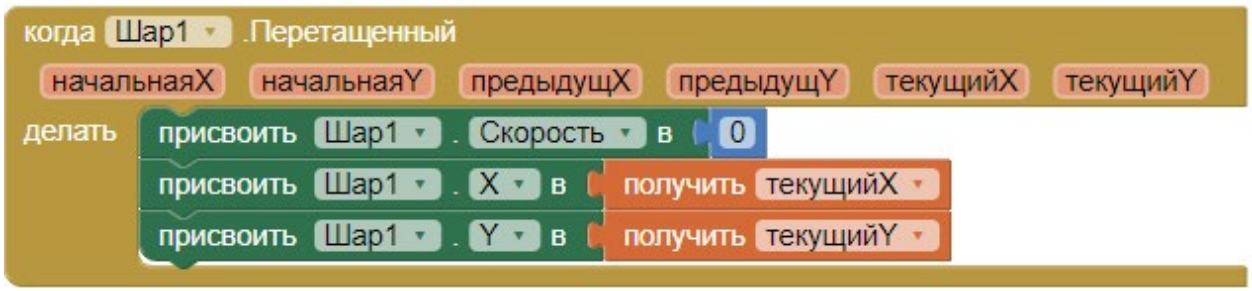
1) Что делают обработчики событий?

2) Чем сеттер свойства отличается от процедуры?

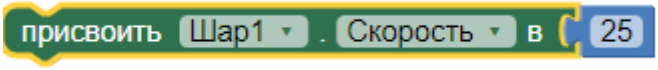
VI. Информация о домашнем задании, инструктаж по его выполнению — 2 мин.

Деятельность учителя: сообщает домашнее задание: поработать со свойствами в блоках, меняя их и наблюдая за поведением шаров в эмуляторе.

Добавить возможность перетаскивать Шар1 по экрану, удерживая его пальцем (в режиме эмулятора это будет ЛКМ):

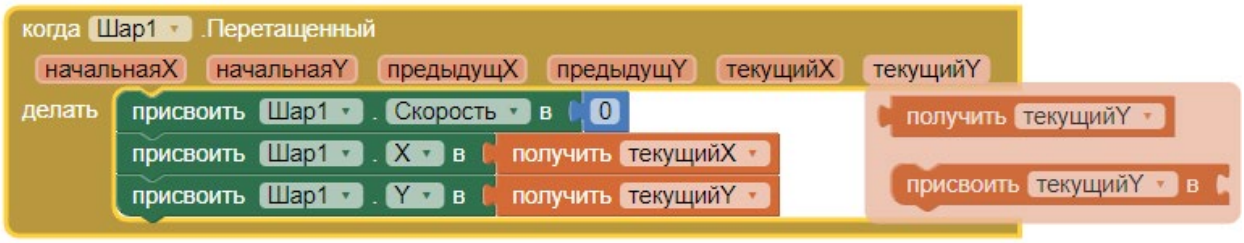


Объясняет, что для нормального перетаскивания скорость Шара обнуляется. Чтобы ее вернуть необходимо добавить сеттер для скорости Шара1 в блок «когда Шар1.НаложениеСОбъектом делать»:

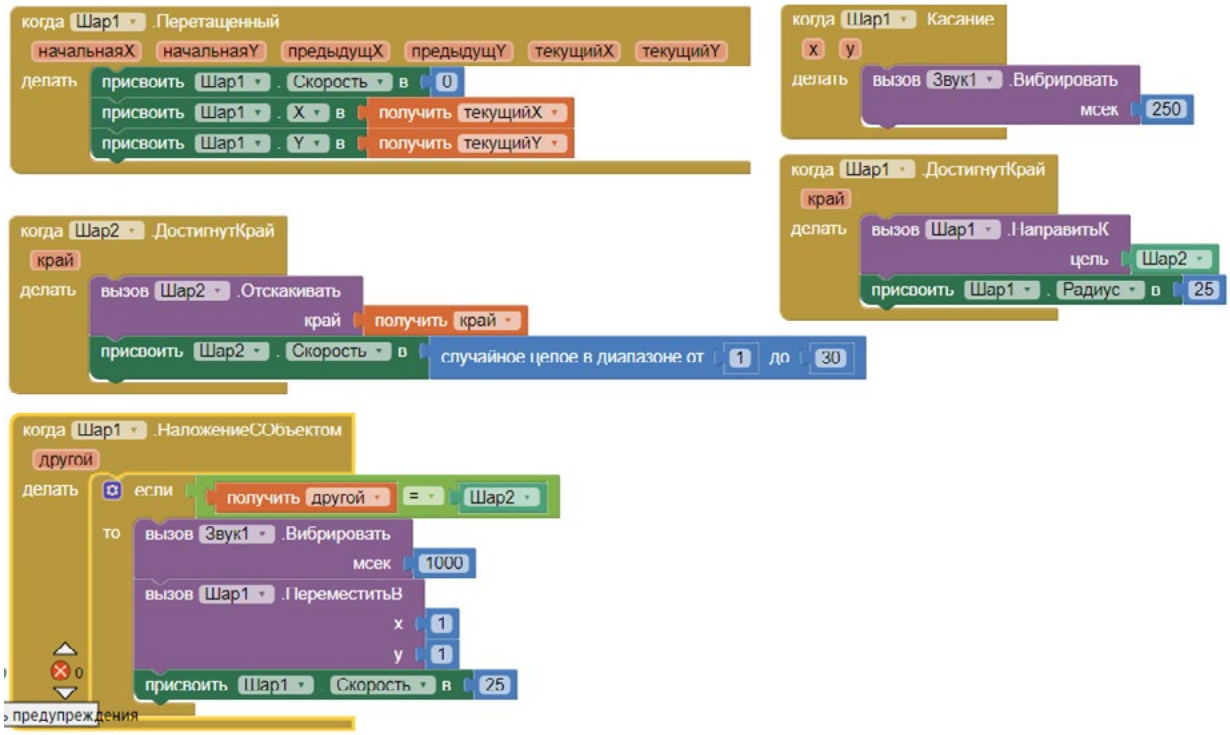


В этом случае, скорость вернётся, как только шары пересекутся на экране.

Геттеры «получить текущий X» и Y добавляются аналогично, как выше геттер для края экрана. Для этого необходимо нажать на оранжевое выражение «текущийY» в блоке. Сами геттеры означают, что Шару1 присваиваются координаты той точки, где пользователь поднял палец (в режиме эмулятора это будет точка, где пользователь отпустил ЛКМ):

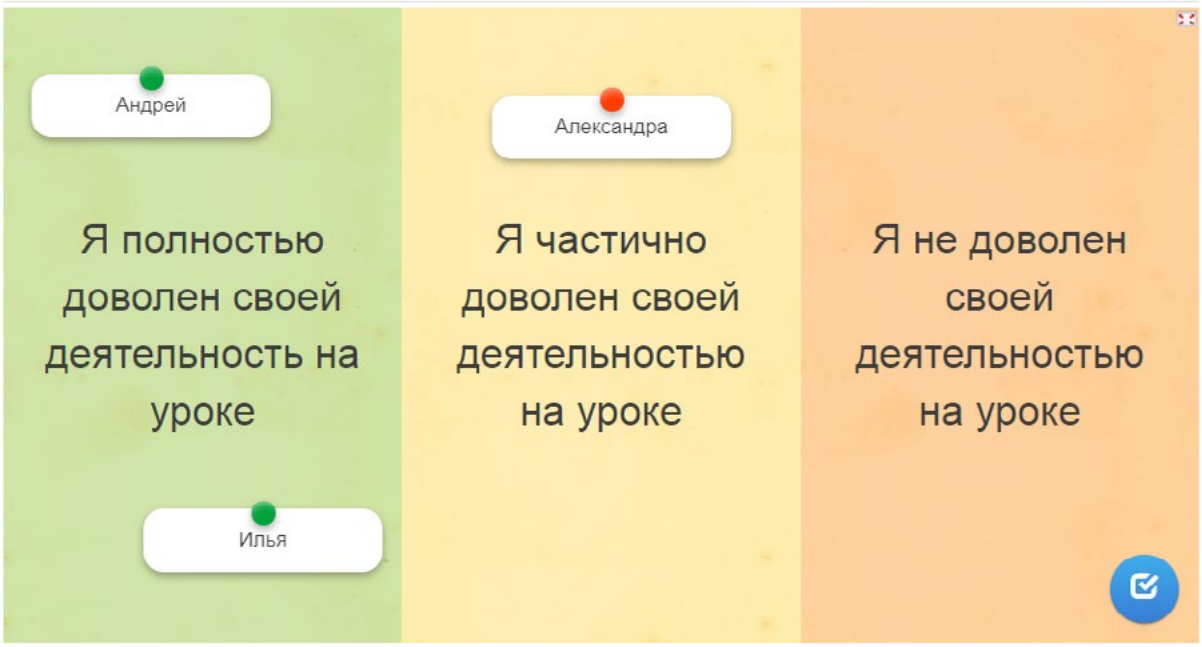


Финальная схема блоков имеет примерно следующий вид:



VII. Этап рефлексии деятельности на уроке — 2 мин.

Деятельность учителя: предлагает учащимся оценить свою деятельность на уроке. Для проведения данного этапа можно создать интерактивный ресурс, например, в сервисе https://learningapps.org/.



Форма аттестации, примеры контрольно-оценочных материалов

Во время проведения курса предполагается текущий, промежуточный и итоговый контроль.

Текущий контроль осуществляется регулярно во время проведения каждого лабораторного занятия, заключается в ответе учащихся на контрольные вопросы, демонстрации разработанных приложений, фронтальных опросов учителем.

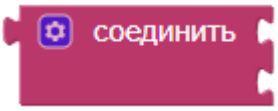
Также в тематическом планировании предполагается один промежуточный тест и одна творческая работа.

Тест для проверки полученных навыков по теме «Работа с компонентами интерфейса и программными блоками в среде АИ»

1. В режиме Дизайнер раздел с компонентами типа кнопка, надпись, выбор даты, список, текстовое поле, флажок называется:
    1. Интерфейс пользователя +
    2. Медиа
    3. Хранилище
    4. Рисование и анимация

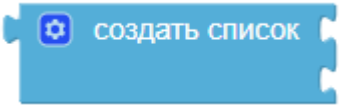
2. Для присвоения переменным и свойствам числовых значений (в режиме Блоки) можно использовать встроенные блоки из раздела:
    1. Математика +
    2. Цвета
    3. Логика
    4. Переменные

3. Можно ли в АИ соединить вместе текст и число при помощи следующего блока:



    1. Да+
    2. Нет
    3. Если число целое
    4. Если число вещественное

4. Для чего нужен следующий блок:



    1. Чтобы создать массив элементов+
    2. Чтобы создать словарь
    3. Чтобы создать компоненту Список из раздела «Интерфейс Пользователя» в режиме Дизайнера
    4. Чтобы создать список экранов приложения

Творческая работа для проверки полученных навыков по теме «Компоненты сенсоров и общения», «Хранилища данных»

Варианты заданий

1. Создать приложение для экстренной отправки пожилыми людьми СМС-сообщений по нескольким адресам с указанием координат местоположения.

Для реализации приложения достаточно немного модифицировать лабораторную работу № 13.

2. Реализовать переводчик с возможностью перевода с русского языка на два языка. Использовать хранилище TinyDB.

Для реализации приложения достаточно модифицировать лабораторных работ № 12 и № 14.

Материалы для организации и проведения учебно-исследовательской и проектной деятельности школьников

Проекты по программированию представляют собой проекты, результатами которых является программа для решения той или иной задачи. Особенностью является то, что одна и та же задача в зависимости от уровня проработки, может быть решена как начинающим, так и опытным программистом.

При выполнении проекта по программированию учащиеся имеют следующие возможности: получить умения самостоятельно формулировать цели и задачи проекта, планировать свою деятельность, получить умение представления результатов своей деятельности.

Проект может разрабатываться индивидуально или группой учащихся. Если задача достаточно сложная, то проекта может быть разбит на подзадачи, подпроекты. Каждую подзадачу будут выполнять различные группы участников проекта. Например, одна группа занимается разработкой алгоритма, другая группа — непосредственно написанием и отладкой кода, третья — подготовкой к презентации проекта.

План работы над проектом по программированию может совпадать с этапами разработки программы, представленной на следующем рисунке.

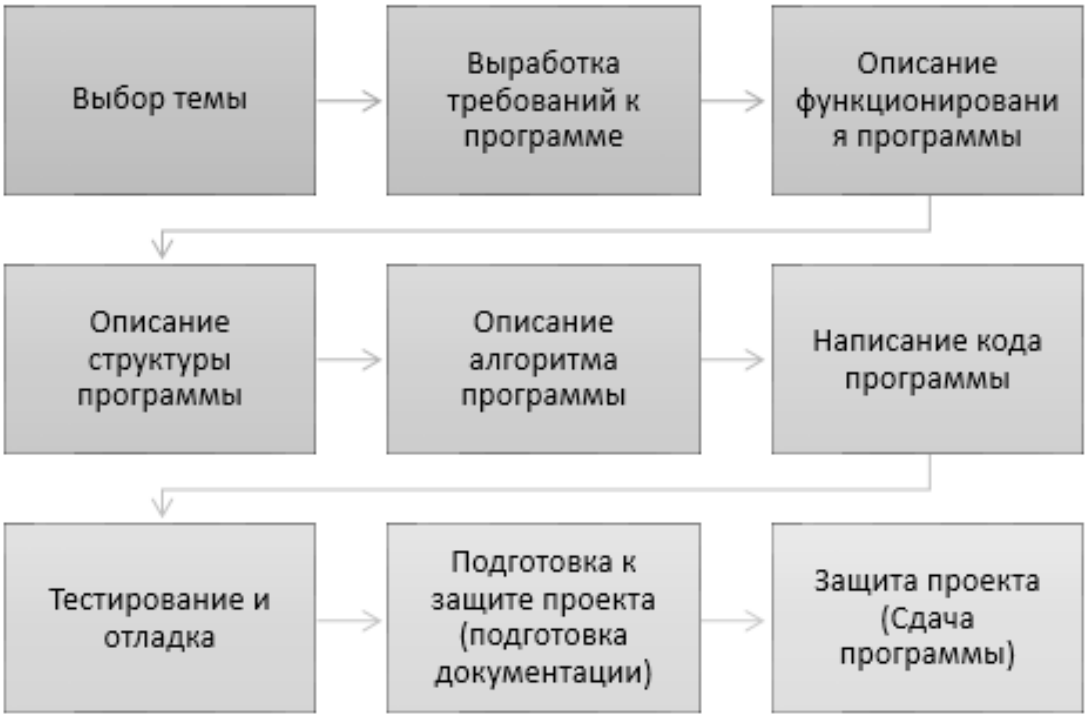


Рисунок 303. Этапы проекта

В помощь участникам проекта можно предложить заполнить следующий учётный лист.

Проект по мобильной разработке

Тема проекта:
Творческое название (при наличии):
Основополагающий вопрос:
Авторы:
1.
2.
3.
...
Предметная область:
Краткая аннотация:
Этапы выполнения проекта:

При подготовке к защите проекта учащимся необходимо подготовить презентацию и доклад, в котором отражаются основные этапы разработки программы, представлен алгоритм решения задачи, листинг программы, основные результаты работы. Можно предложить в помощь учащимся заполнить следующий чек-лист:

1. Аннотация.
2. Содержание.
3. Постановка задачи:
    a. Возможности использования программы
    b. Описание интерфейса
4. Формализация алгоритма:
    a. Перечень подпрограмм (при наличии)
    b. Описание алгоритма (блок-схема или подробное словесное описание алгоритма)
5. Листинг программы (текст программы).
6. Тестовые примеры
    a. Результаты работы
    b. Скриншоты результатов работы
7. Описание размещения.
8. Требования к программным и аппаратным средствам.
9. Для оценивания проекта могут быть разработаны специальные оценочные листы.

Ниже представлен пример оценочного листа:

Лист оценивания проекта

Критерий оценивания	1 группа	2 группа	...
Актуальность темы			
Соответствие содержания проекта заявленной теме			
Техническая сложность разработанной программы			
Оригинальность алгоритма			
Дизайн интерфейса			
Степень разработанности программы			
Применение программы для решения аналогичных задач			
Итоговое количество баллов			

Ниже приведены возможные темы исследовательских проектов учащихся:

Проект 1. Создать приложение Шагомер. Можно использовать компоненту Pedometer из раздела Сенсоры режима Дизайнер.

Например, в простом варианте дизайн и программные блоки приложения могут иметь примерно следующий вид:

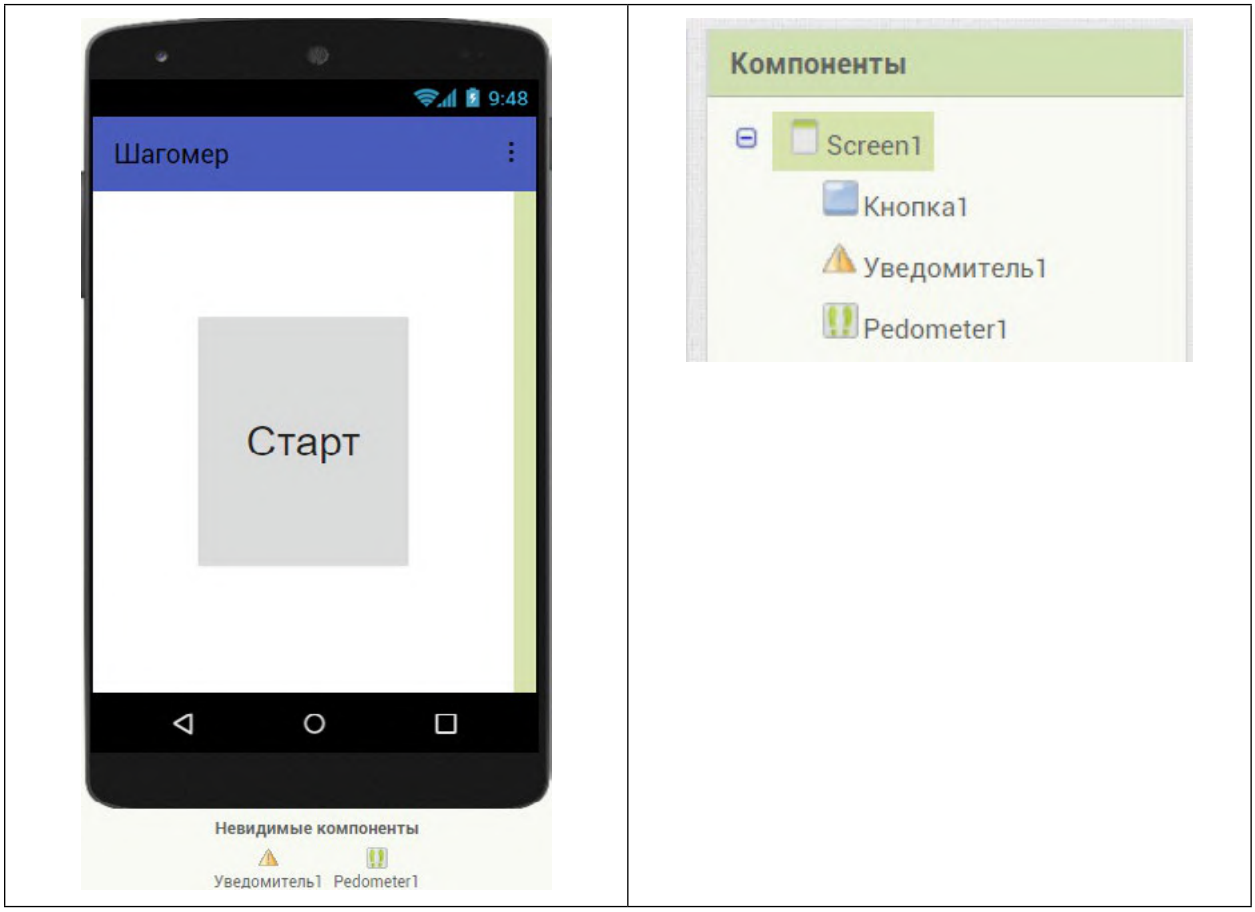


Рисунок 304. Дизайн простого варианта приложения Шагомер

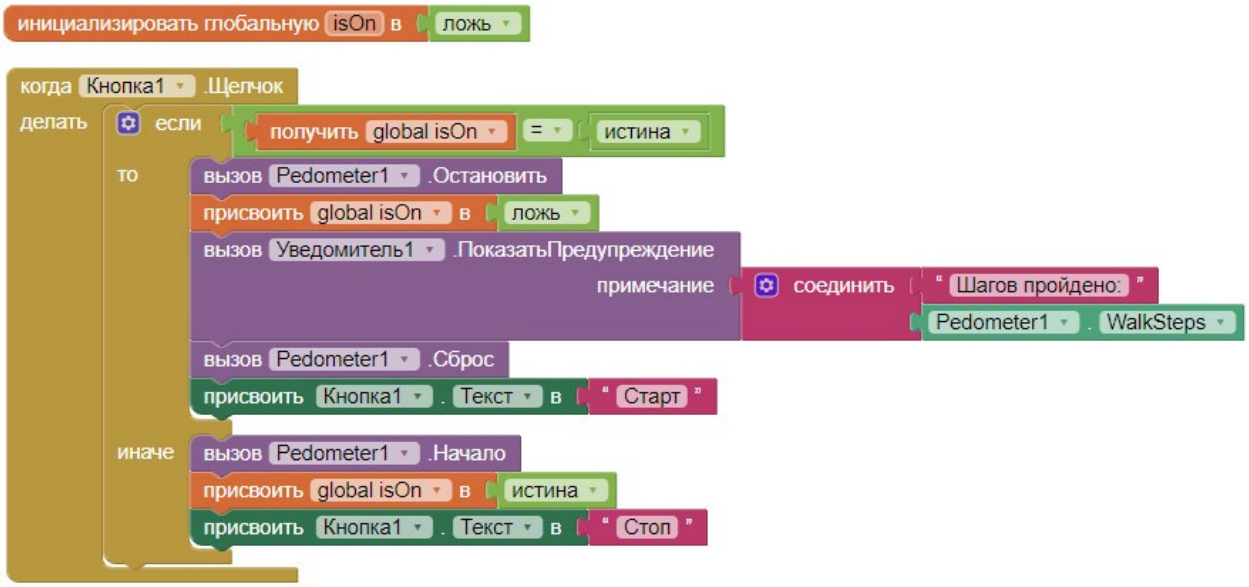


Рисунок 305. Программные блоки простого варианта приложения Шагомер

В этом варианте при нажатии на кнопку Старт начинается подсчет пройденных шагов. Надпись на кнопке меняется на «Стоп». При повторном нажатии подсчет останавливается, выдается уведомление о количестве пройденных шагов и происходит сброс шагомера. Булева переменная isOn используется для определения включен или выключен шагомер.

Можно сделать немного более сложный вариант: с записью и извлечением из базы TinyDB количества шагов, которые необходимо пройти. Также добавить компоненту Бегунок1, устанавливающую это количество. При закрытии приложения и повторном его открытии, положение бегунка считывается из базы. В этом случае дизайн и программные блоки приложения могут иметь примерно следующий вид:

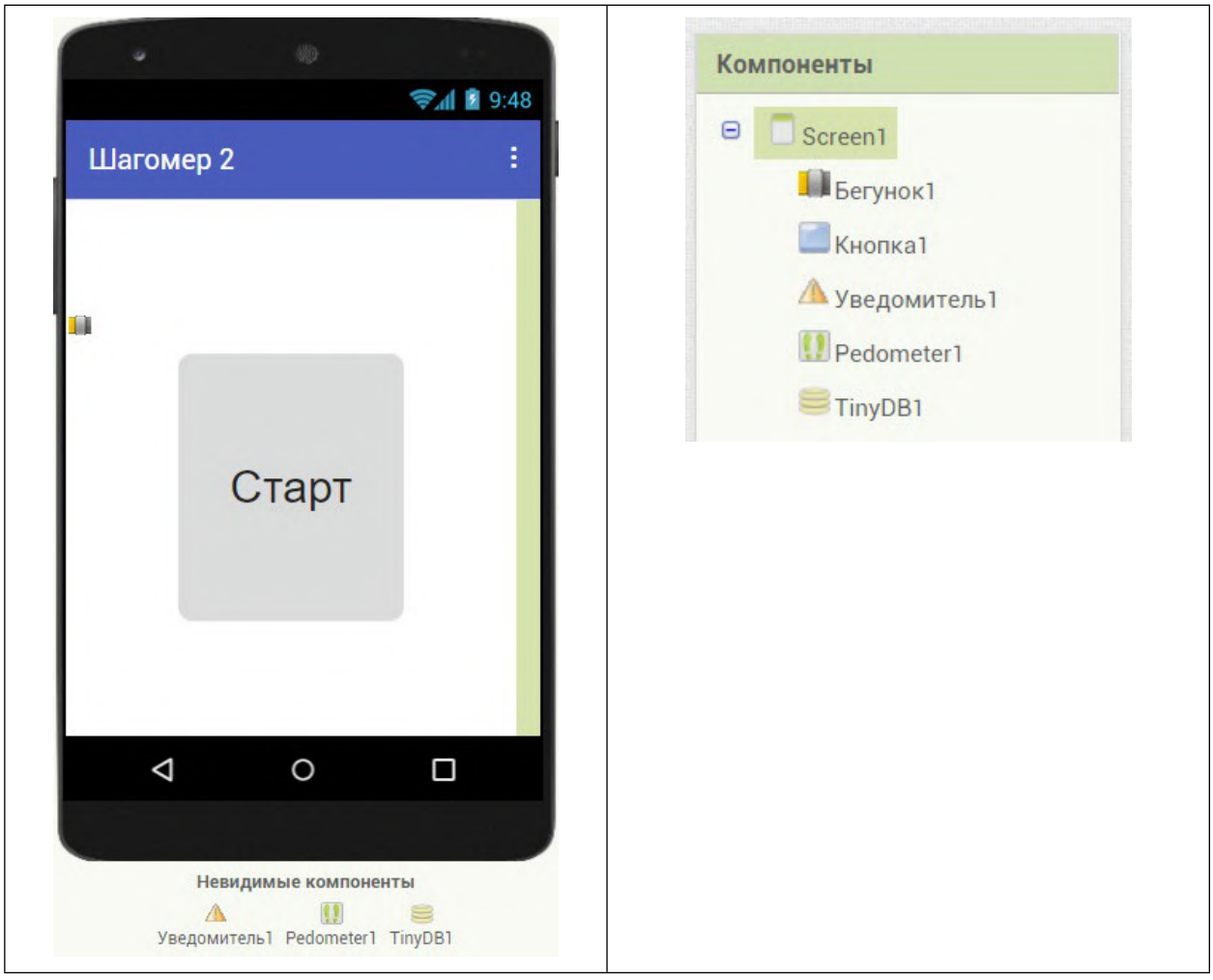


Рисунок 306. Дизайн второго варианта приложения Шагомер

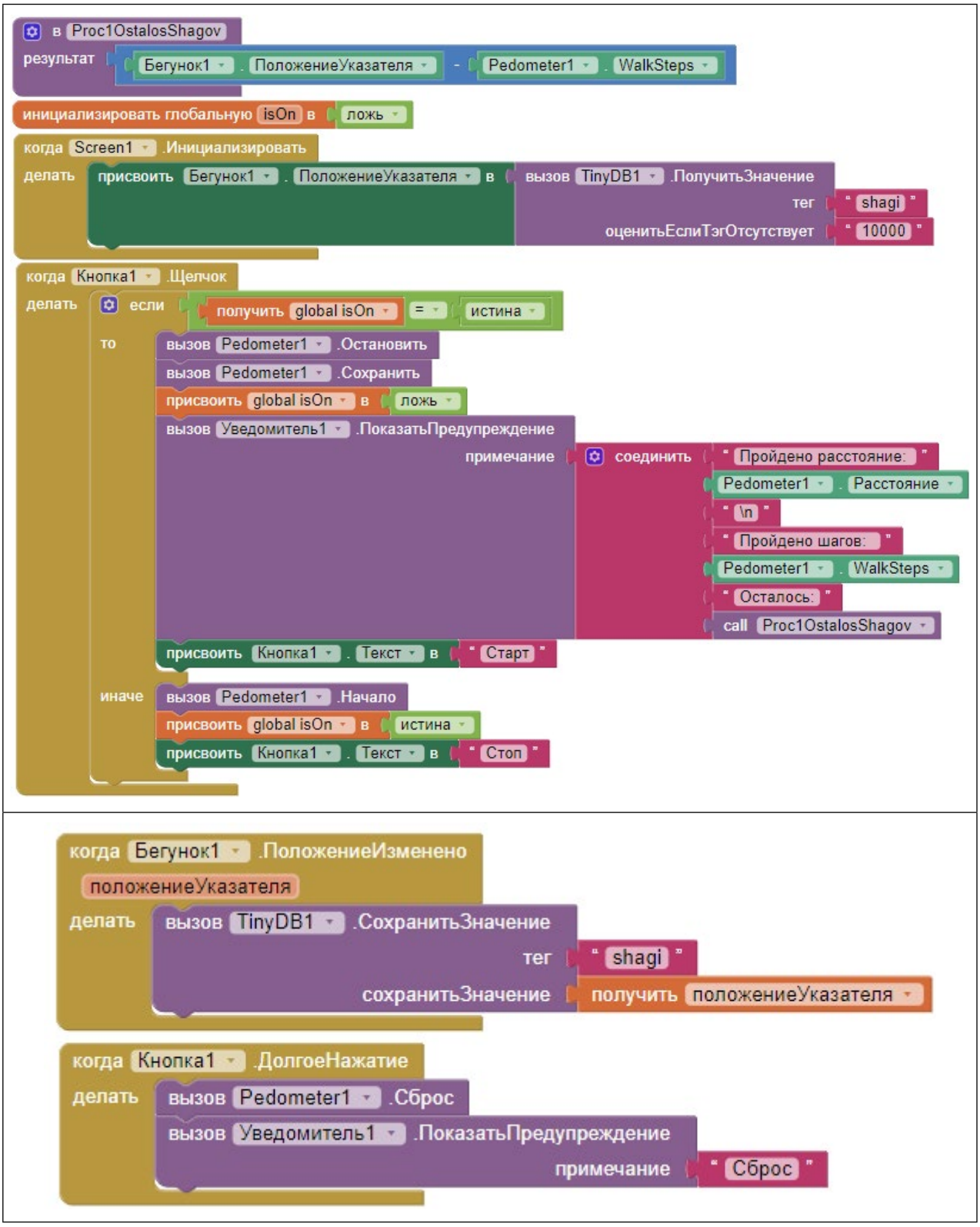


Рисунок 307. Программные блоки второго варианта приложения Шагомер

В этом варианте приложение должно восстанавливать пройденные шаги за счет вызова процедуры сенсора «вызов Pedometer1.Сохранить». Для сброса шагомера предусмотрено долгое нажатие кнопки.

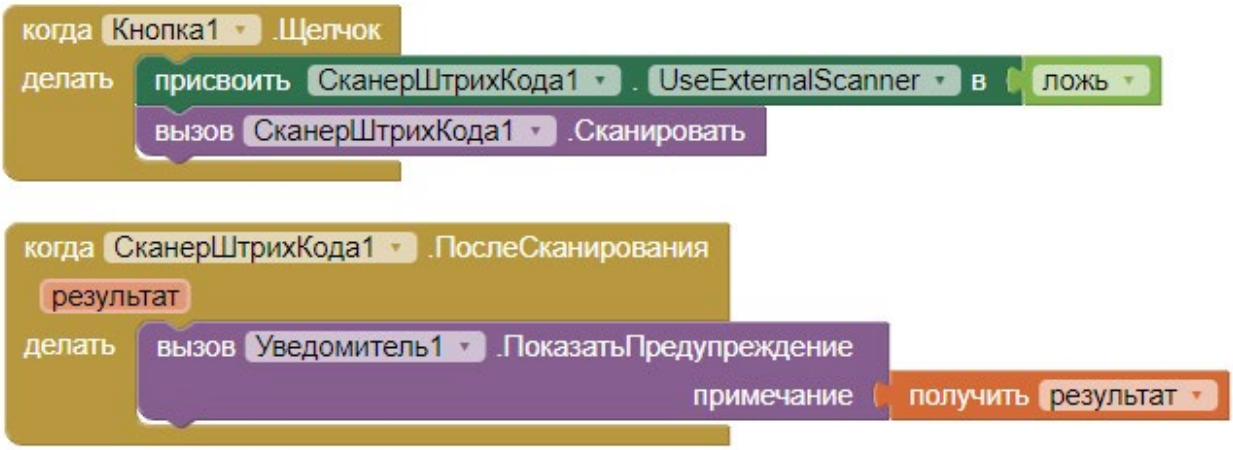


Рисунок 308. Блоки для работы с компонентой СканерШтрихКода1

Проект 3. Создать приложение «Пианино». См. [7] (Необходимо знание английского языка).

Проект 4. Создать приложение-игру «Найди золото» См. [8] (Необходимо знание английского языка).

Список литературы

1. Язык Kawa (на англ.языке) [Электронный ресурс] URL: https://www.gnu.org/software/kawa/index.html (дата обращения: 19.03.2021).
2. Установка эмулятора (на англ.языке) [Электронный ресурс] URL: http://appinventor.mit.edu/explore/ai2/setup-emulator (дата обращения: 19.03.2021).
3. Установка эмулятора в ОС Windows (на англ.языке) [Электронный ресурс] URL: http://appinventor.mit.edu/explore/ai2/windows (дата обращения: 19.03.2021).
4. AITech - Using Procedures and Any component blocks (на англ.языке) [Электронный ресурс] URL: https://appinventor.mit.edu/explore/blogs/karen/2016/07-0.html (дата обращения: 19.03.2021).
5. Процедуры в АИ (на англ.языке) [Электронный ресурс] URL: https://appinventor.mit.edu/explore/ai2/support/concepts/procedures (дата обращения: 19.03.2021).
6. База данных TinyDB (на англ.языке) [Электронный ресурс] URL: https://tinydb.readthedocs.io/en/latest/ (дата обращения: 19.03.2021).
7. Игра Пианино (на англ.языке) [Электронный ресурс] URL: https://drive.google.com/drive/folders/1f9D_bQPy-G17EmdPCpY3-KoKAfH1E7qE (дата обращения: 19.03.2021).
8. Игра «Найди золото» (на англ.языке) [Электронный ресурс] URL: https://drive.google.com/drive/folders/1xRSZGMLmtU7nJn22ToWCZIC92Z_bPaEF (дата обращения: 19.03.2021).
9. Инструкции по установке USB соединения (на англ.языке) [Электронный ресурс] URL: http://appinventor.mit.edu/explore/ai2/setup-device-usb (дата обращения: 19.03.2021).

Григорьев С. Г., Сабитов Р.А., Сабитов Ш.Р., Смирнова Г.С.

Мобильная разработка с использованием оборудования центра цифрового образования детей «IT-куб»

Методическое пособие

Под редакцией С. Г. Григорьева

Центр Естественно-научного и математического образования
Руководитель Центра *З. Г. Гапонюк*
Ответственный за выпуск *Е. С. Карауш*
Редактор *Е. С. Карауш*
Художественное оформление *С. И. Ситников*
Компьютерная вёрстка и техническое редактирование *О. С. Ивановой*
Корректор *Г. И. Мосякина*