

Индивидуальный предприниматель


М.А. Иванова

«2» марта 2026 г.

**Дополнительная общеобразовательная общеразвивающая
программа «Основы создания мобильных приложений в App Inventor»**

Программа реализуется исключительно с применением электронного обучения и дистанционных образовательных технологий.

Направленность: техническая. Возраст обучающихся: от 10 лет.

Срок реализации: 28 академических часов в течение 14 недель

Автор программы: Иванова Марина Александровна

Содержание

1. Основные характеристики образовательной программы.....	3
2. Учебный план.....	5
3. Календарный учебный график	6
4. Рабочая программа.....	7
5. Оценка качества освоения программы.....	24
6. Условия реализации образовательной программы	25
7. Фонд оценочных материалов	26
8. Электронная библиотека	28

1. Основные характеристики образовательной программы

1.1. Актуальность программы

Мобильные приложения стали частью повседневной жизни: дети пользуются смартфоном для учёбы, общения и игр с раннего возраста. Умение не только пользоваться приложениями, но и создавать их, переводит ребёнка из позиции потребителя цифровых продуктов в позицию их создателя и закладывает основу для дальнейшего изучения программирования и выбора IT-профессии.

MIT App Inventor — визуальная среда разработки мобильных приложений, созданная Массачусетским технологическим институтом для обучения программированию. Приложение собирается из готовых компонентов (кнопки, надписи, изображения, датчики), а его поведение программируется блоками, как в Scratch. Это позволяет детям от 10 лет уже на первом занятии создать работающее приложение и запустить его на собственном телефоне.

Программа отвечает на запрос современного образования в развитии у детей вычислительного мышления, инженерного подхода к решению задач и проектной деятельности. Освоение App Inventor формирует понимание того, как устроены реальные мобильные приложения: интерфейс, события, данные, датчики устройства, и готовит обучающихся к изучению текстовых языков программирования.

1.2. Направленность и уровень освоения программы

Программа имеет техническую направленность и ориентирована на развитие алгоритмического и логического мышления, формирование навыков программирования и разработки мобильных приложений.

Уровень освоения — базовый.

1.3. Новизна программы

Новизна программы заключается в том, что обучающиеся с первого занятия создают настоящие мобильные приложения и сразу проверяют их на своём телефоне. Программа интегрирует:

- поэтапное освоение ключевых концепций программирования от простого к сложному;
- практико-ориентированный подход: на каждом занятии создаётся законченное работающее приложение;
- использование возможностей реального мобильного устройства: сенсорного экрана, звука, синтеза речи, вибрации, акселерометра;
- освоение основ проектирования интерфейса и хранения данных;
- развитие творческих способностей через создание собственных игр и полезных приложений;
- подготовку к публичной презентации результатов своей работы.

Реализация программы в электронном формате с применением дистанционных технологий позволяет обучающимся осваивать материал в комфортном темпе, неоднократно возвращаться к записям занятий и получать индивидуальную поддержку преподавателя.

1.4. Обучающиеся

Программа рассчитана на обучающихся в возрасте от 10 лет без требований к предварительной подготовке в области программирования. Опыт работы в среде Scratch желателен, но не обязателен.

1.5. Объем и срок реализации программы

- Срок реализации: 14 недель
- Общий объем программы: 28 академических часов (академический час равен 45 астрономическим минутам)
- Режим занятий: 1 онлайн-занятие в неделю продолжительностью 2 академических часа (90 минут, с 10-минутным перерывом после первого часа) в формате прямого эфира с преподавателем. На занятии преподаватель демонстрирует каждое действие, обучающиеся сразу повторяют за ним на своих компьютерах и телефонах, создавая проект синхронно с преподавателем.

1.6. Цели и задачи программы

Цель программы: развитие алгоритмического мышления и творческих способностей обучающихся путём освоения основ программирования через создание мобильных приложений и игр в визуальной среде MIT App Inventor.

Задачи программы:

Обучающие:

- сформировать представление об устройстве мобильного приложения: интерфейс, компоненты, события, данные;
- обучить работе в среде App Inventor: дизайнер интерфейса, редактор блоков, подключение мобильного устройства, сборка приложения;
- сформировать навыки проектирования интерфейса с использованием компонентов и их расположения;
- обучить использованию основных алгоритмических конструкций: последовательность, ветвление, событие, таймер;
- сформировать понимание работы с переменными, списками и случайными числами;
- обучить работе с мультимедиа (звук, синтез речи) и датчиками мобильного устройства;
- сформировать навыки работы с графикой и движением объектов на холсте;
- обучить сохранению данных и созданию многоэкранных приложений;

- обучить основам проектной деятельности: от замысла до реализации и презентации.

Развивающие:

- развить алгоритмическое и логическое мышление;
- развить пространственное мышление через работу с координатами на экране устройства;
- развить творческое мышление и способность к генерации идей;
- развить навыки отладки программ и поиска ошибок;
- развить навыки самостоятельного обучения и работы с технической документацией.

Воспитательные:

- воспитать интерес к техническому творчеству и программированию;
- сформировать ответственное отношение к результатам своей работы;
- сформировать культуру безопасного и осознанного использования мобильных устройств и уважение к интеллектуальной собственности.

1.7. Планируемые результаты обучения

По завершении обучения по программе обучающиеся будут:

Знать:

- основные понятия программирования: алгоритм, событие, условие, переменная, список, таймер;
- интерфейс и основные возможности среды App Inventor;
- назначение основных компонентов интерфейса, медиа, холста, датчиков и хранения данных;
- систему координат на холсте мобильного приложения;
- назначение переменных и списков и способы работы с ними;
- принцип событийного программирования;
- этапы разработки приложения от идеи до установки на телефон и презентации.

Уметь:

- создавать, сохранять и открывать проекты в App Inventor;
- подключать мобильное устройство или эмулятор для проверки приложения;
- проектировать интерфейс с помощью компонентов и расположений;
- программировать реакцию приложения на нажатия, касания и ввод текста;
- использовать условия, переменные, списки и случайные числа;
- работать со звуком, синтезом речи, холстом, спрайтами, таймером и датчиками;
- сохранять данные в TinyDB и создавать приложения из нескольких экранов;
- находить и исправлять ошибки в программах;
- собирать приложение и устанавливать его на телефон.

Владеть:

- навыками алгоритмического мышления;
- навыками работы в визуальной среде разработки мобильных приложений App Inventor;
- базовыми приёмами программирования: события, условия, переменные, списки, таймеры.

1.8. Итоговый документ

Обучающемуся, успешно прошедшему итоговую аттестацию (освоившему программу), выдаётся сертификат о прохождении обучения по дополнительной общеобразовательной общеразвивающей программе. Обучающемуся, не прошедшему итоговую аттестацию или освоившему часть программы, по заявлению выдаётся справка об обучении.

2. Учебный план

№ п/п	Наименование разделов и тем занятий	Количество ак. часов		Форма контроля
		Всего	Практика	
1.	Знакомство с App Inventor. Создание первого приложения	2	2	практическое задание
2.	Компоненты интерфейса и их расположение	2	2	практическое задание
3.	Работа со звуком и медиафайлами	2	2	практическое задание
4.	Ввод текста и синтез речи	2	2	практическое задание
5.	Переменные и арифметические операции	2	2	практическое задание
6.	Условия. Ветвление в программе	2	2	практическое задание
7.	Холст и рисование. События касания	2	2	практическое задание
8.	Спрайты и движение объектов	2	2	практическое задание
9.	Таймер. Создание игры	2	2	практическое задание
10.	Датчики мобильного устройства	2	2	практическое задание
11.	Списки и случайный выбор	2	2	практическое задание
12.	Несколько экранов и сохранение данных	2	2	практическое задание
13.	Подготовка проекта. Создание приложения-викторины	2	2	практическое задание
14.	Итоговый контроль	2	2	защита проекта
	Итого часов	28	28	

3. Календарный учебный график

№	Наименование разделов и тем	Неделя и день обучения
1.	Знакомство с App Inventor. Создание первого приложения	1 неделя, среда
2.	Компоненты интерфейса и их расположение	2 неделя, среда
3.	Работа со звуком и медиафайлами	3 неделя, среда
4.	Ввод текста и синтез речи	4 неделя, среда
5.	Переменные и арифметические операции	5 неделя, среда

№	Наименование разделов и тем	Неделя и день обучения
6.	Условия. Ветвление в программе	6 неделя, среда
7.	Холст и рисование. События касания	7 неделя, среда
8.	Спрайты и движение объектов	8 неделя, среда
9.	Таймер. Создание игры	9 неделя, среда
10.	Датчики мобильного устройства	10 неделя, среда
11.	Списки и случайный выбор	11 неделя, среда
12.	Несколько экранов и сохранение данных	12 неделя, среда
13.	Подготовка проекта. Создание приложения-викторины	13 неделя, среда
14.	Итоговый контроль	14 неделя, среда

Конкретное расписание составляется для каждой учебной группы по мере набора учебных групп. Численность учебных групп — 5–8 человек, осваивающих образовательную программу одновременно.

Занятия проводятся по средам в 15:20 МСК.

4. Рабочая программа

УРОК 1. ЗНАКОМСТВО С APP INVENTOR. СОЗДАНИЕ ПЕРВОГО ПРИЛОЖЕНИЯ

Длительность: 2 академических часа (90 минут)

Формат: синхронная работа – преподаватель показывает, ученики сразу повторяют

ХОД ЗАНЯТИЯ:

1. Введение в мобильные приложения и App Inventor (5 минут)

Преподаватель рассказывает:

- Что такое мобильное приложение и как оно попадает на телефон
- Что такое App Inventor и какие приложения в нём можно создать
- Чем App Inventor похож на Scratch: программа собирается из блоков

2. Вход в App Inventor (10 минут)

Преподаватель показывает → ученики повторяют:

- Открыть браузер и перейти на сайт code.appinventor.mit.edu
- Нажать кнопку «Continue Without An Account» (продолжить без аккаунта)
- Записать полученный код доступа в тетрадь – по нему можно вернуться к своим проектам
- Закрывать приветственное окно
- Нажать «Projects» → «Start new project» (создать новый проект)
- Ввести имя проекта Privet (только латинские буквы, без пробелов) и нажать «OK»

3. Знакомство с интерфейсом (10 минут)

Преподаватель показывает каждый элемент → ученики находят его на своём экране:

- Палитра компонентов «Palette» (слева)
- Экран телефона «Viewer» (в центре)
- Список компонентов «Components» (справа)
- Свойства выбранного компонента «Properties» (справа)
- Кнопки переключения «Designer» (дизайнер) и «Blocks» (блоки) – справа вверху

Преподаватель объясняет:

- В дизайнера мы рисуем, как выглядит приложение
- В блоках программируем, что приложение делает

4. Подключение телефона (15 минут)

Преподаватель показывает → ученики повторяют:

- Установить на телефон или планшет приложение MIT AI2 Companion
- На компьютере нажать «Connect» → «AI Companion»
- На экране появится QR-код и шестизначный код

- В приложении на телефоне нажать «scan QR code» и навести камеру на экран компьютера
- Дождаться, пока на телефоне откроется пустой экран приложения

Преподаватель помогает тем, у кого телефон не подключился:

- Проверить подключение телефона к интернету
- Ввести шестизначный код вручную
- При отсутствии телефона – использовать эмулятор («Connect» → «Emulator»)

Перерыв 10 минут

5. Создание первого приложения «Привет!» (25 минут)

Преподаватель делает каждый шаг → ученики повторяют:

Шаг 1: Дизайн

- Из палитры «User Interface» перетащить на экран компонент Label (надпись)
- В свойствах изменить Text на «Здесь появится приветствие»
- Изменить FontSize (размер шрифта) на 20
- Перетащить на экран компонент Button (кнопка)
- Изменить Text кнопки на «Нажми меня»
- Проверить на телефоне – надпись и кнопка уже видны

Шаг 2: Программирование

- Перейти в редактор «Blocks»
- Нажать на Button1 – открыть его блоки
- Перетащить блок «when Button1.Click» (когда кнопка нажата)
- Нажать на Label1 и найти блок «set Label1.Text to» (установить текст надписи)
- Вставить его внутрь блока «when Button1.Click»
- Из раздела «Text» взять пустой текстовый блок, присоединить и написать «Привет! Это моё первое приложение!»
- Проверить – нажать кнопку на телефоне

6. Свободная практика под наблюдением (10 минут)

Преподаватель предлагает:

- Изменить цвет кнопки (свойство BackgroundColor)
- Изменить цвет надписи (свойство TextColor)
- Поменять текст приветствия на свой
- Преподаватель отвечает на вопросы и помогает тем, кто отстал

7. Подведение итогов (5 минут)

- Преподаватель спрашивает, что узнали на уроке
- Напоминает: дизайнер – внешний вид, блоки – поведение
- Анонсирует тему следующего урока

Результат урока:

- Все обучающиеся вошли в App Inventor и создали первый проект
- Подключили телефон и увидели своё приложение вживую
- Научились связывать нажатие кнопки с действием

УРОК 2. КОМПОНЕНТЫ ИНТЕРФЕЙСА И ИХ РАСПОЛОЖЕНИЕ

Длительность: 2 академических часа (90 минут)

Формат: синхронная работа – преподаватель показывает, ученики сразу повторяют

ХОД ЗАНЯТИЯ:

1. Повторение и вход в App Inventor (5 минут)

- Ученики входят по своему коду и подключают телефон
- Преподаватель спрашивает: где мы задаём внешний вид, а где – поведение?

2. Новый проект и свойства экрана (10 минут)

Преподаватель показывает → ученики повторяют:

- Создать новый проект Vizitka
- Выбрать Screen1 в списке компонентов
- Изменить Title (заголовок экрана) на «Моя визитка»
- Изменить BackgroundColor экрана на любой светлый цвет
- Свойство AlignHorizontal – выбрать «Center» (по центру)

3. Изображения (15 минут)

Преподаватель показывает → ученики повторяют:

- Перетащить на экран компонент Image (изображение)
- В свойстве Picture нажать «Upload File» и загрузить картинку (аватар, любимое животное)
- Изменить Width и Height на 150 пикселей
- Проверить на телефоне

4. Надписи с информацией (10 минут)

Преподаватель делает → ученики повторяют:

- Добавить Label с именем – FontSize 24, FontBold включить
- Добавить Label «Мне ... лет» и Label «Я люблю ...»
- Заполнить текст своими данными

Перерыв 10 минут

5. Расположение компонентов (15 минут)

Преподаватель объясняет и показывает:

- Компоненты на экране выстраиваются сверху вниз
- Чтобы поставить их в ряд, нужно расположение HorizontalArrangement

Преподаватель делает → ученики повторяют:

- Из палитры «Layout» перетащить HorizontalArrangement под надписи

- Перетащить внутрь него две кнопки
- Подписать кнопки «Хобби» и «Мечта»
- Свойство Width у расположения – «Fill parent» (по ширине экрана)

6. Программирование кнопок (20 минут)

Преподаватель делает пошагово → ученики повторяют:

Шаг 1: Скрытая надпись

- Добавить под расположением Label3 с текстом о хобби
- Снять галочку Visible – надпись скрытана

Шаг 2: Показать надпись

- В блоках взять «when Button1.Click»
- Вставить «set Label3.Visible to» и логический блок «true»
- Проверить – по нажатию «Хобби» появляется текст

Шаг 3: Вторая кнопка

- Для кнопки «Мечта» менять текст той же надписи: «set Label3.Text to»
«Моя мечта – ...»
- Не забыть сделать надпись видимой
- Проверить обе кнопки

7. Подведение итогов (5 минут)

- Обучающиеся по очереди показывают визитки на камеру
- Преподаватель отмечает удачные решения

Результат урока:

- Все обучающиеся создали приложение-визитку
- Освоили компоненты Image, Label, Button и расположения
- Научились показывать и скрывать компоненты

УРОК 3. РАБОТА СО ЗВУКОМ И МЕДИАФАЙЛАМИ

Длительность: 2 академических часа (90 минут)

Формат: синхронная работа – преподаватель показывает, ученики сразу повторяют

ХОД ЗАНЯТИЯ:

1. Введение (5 минут)

Преподаватель рассказывает:

- Какие звуки есть в приложениях на телефоне
- Чем отличаются компоненты Sound (короткие звуки) и Player (музыка)

2. Подготовка медиафайлов (10 минут)

Преподаватель показывает → ученики повторяют:

- Создать проект GovoryashchiyKot
- Скачать с материалов урока картинку кота и звук мяуканья
- В разделе «Media» (под списком компонентов) нажать «Upload File»

- Загрузить картинку и звук

3. Кот-кнопка (15 минут)

Преподаватель делает → ученики повторяют:

- Добавить Label «Погладь кота!» – FontSize 24
- Добавить Button, в свойстве Image выбрать картинку кота
- Стереть текст кнопки (свойство Text)
- Из палитры «Media» добавить компонент Sound
- Свойство Source – выбрать звук мяуканья
- Компонент Sound – невидимый, он появляется под экраном

4. Программирование звука (10 минут)

Преподаватель делает → ученики повторяют:

- «when Button1.Click» → «call Sound1.Play»
- Проверить – кот мяукает
- Добавить «call Sound1.Vibrate» с числом 500 (полсекунды)
- Проверить – телефон вибрирует

Перерыв 10 минут

5. Музыкальный проигрыватель (20 минут)

Преподаватель показывает → ученики повторяют:

- Загрузить в «Media» короткую мелодию
- Добавить компонент Player, свойство Source – мелодия
- Добавить HorizontalArrangement с кнопками «Музыка» и «Стоп»
- «when Button2.Click» → «call Player1.Start»
- «when Button3.Click» → «call Player1.Stop»
- Проверить обе кнопки

6. Громкость (10 минут)

Преподаватель показывает → ученики повторяют:

- Добавить компонент Slider (ползунок) из «User Interface»
- Свойства: MinValue 0, MaxValue 100, ThumbPosition 50
- «when Slider1.PositionChanged» → «set Player1.Volume to» «thumbPosition»
- Проверить – двигать ползунок во время музыки

7. Подведение итогов (5 минут)

- Преподаватель спрашивает, чем Sound отличается от Player
- Обсуждают, в каких приложениях нужен звук

Результат урока:

- Все обучающиеся создали приложение «Говорящий кот»
- Научились загружать картинки и звуки
- Освоили компоненты Sound, Player и Slider

УРОК 4. ВВОД ТЕКСТА И СИНТЕЗ РЕЧИ

Длительность: 2 академических часа (90 минут)

Формат: синхронная работа – преподаватель показывает, ученики сразу повторяют

ХОД ЗАНЯТИЯ:

1. Введение (5 минут)

Преподаватель рассказывает:

- Как телефон умеет читать текст вслух
- Где используется синтез речи: навигаторы, голосовые помощники

2. Поле для ввода текста (15 минут)

Преподаватель показывает → ученики повторяют:

- Создать проект Govorilka
- Добавить Label «Напиши, что сказать»
- Добавить TextBox (текстовое поле), свойство Hint – «Введите текст»
- Свойство Width – «Fill parent»
- Добавить Button «Сказать»
- Из палитры «Media» добавить TextToSpeech
- Свойство Language у TextToSpeech – ru

3. Программирование речи (15 минут)

Преподаватель делает → ученики повторяют:

- «when Button1.Click» → «call TextToSpeech1.Speak message»
- В message вставить блок «TextBox1.Text»
- Проверить – ввести «Привет!» и нажать кнопку
- Если телефон молчит – проверить громкость и установленный голосовой модуль

Перерыв 10 минут

4. Склеивание текста (15 минут)

Преподаватель показывает → ученики повторяют:

- Добавить второй TextBox для имени
- Из раздела «Text» взять блок «join» (соединить)
- Собрать фразу: «join» «Привет, » + TextBox2.Text + «! » + TextBox1.Text
- Подставить её в TextToSpeech и в Label1.Text
- Проверить – телефон произносит фразу с именем

5. Настройки голоса (15 минут)

Преподаватель делает → ученики повторяют:

- Добавить два ползунка Slider: «Высота голоса» и «Скорость»
- Свойства: MinValue 0, MaxValue 2, ThumbPosition 1
- Перед «call TextToSpeech1.Speak» вставить «set TextToSpeech1.Pitch to» Slider1.ThumbPosition

- И «set TextToSpeech1.SpeechRate to» Slider2.ThumbPosition
- Проверить – робот говорит тонким или низким голосом, быстро или медленно

6. Очистка поля (5 минут)

- Добавить кнопку «Очистить»: «set TextBox1.Text to» пустой текст

7. Подведение итогов (5 минут)

- Преподаватель спрашивает, что делает блок «join»
- Обучающиеся демонстрируют свои «говорилки»

Результат урока:

- Все обучающиеся создали приложение «Говорилка»
- Научились принимать текст от пользователя
- Освоили склеивание строк и синтез речи

УРОК 5. ПЕРЕМЕННЫЕ И АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ

Длительность: 2 академических часа (90 минут)

Формат: синхронная работа – преподаватель показывает, ученики сразу повторяют

ХОД ЗАНЯТИЯ:

1. Введение: что такое переменная (5 минут)

Преподаватель рассказывает:

- Переменная – «коробочка» с именем, в которой хранится значение
- Чем переменные в App Inventor похожи на переменные в Scratch

2. Дизайн калькулятора (15 минут)

Преподаватель показывает → ученики повторяют:

- Создать проект Kalkulyator
- Добавить два TextBox, у обоих включить NumbersOnly (только числа)
- Добавить HorizontalArrangement с четырьмя кнопками: «+», «-», «×», «÷»
- Добавить Label «Результат: » – FontSize 24

3. Сложение (10 минут)

Преподаватель делает → ученики повторяют:

- «when Button1.Click» → «set Label1.Text to»
- Из раздела «Math» взять блок «+»
- Вставить в него TextBox1.Text и TextBox2.Text
- Проверить: $5 + 7 = 12$

Перерыв 10 минут

4. Остальные действия (15 минут)

Преподаватель показывает → ученики повторяют:

- Скопировать блок сложения (Ctrl+C, Ctrl+V)

- Поменять кнопку на Button2 и операцию на «—»
- Так же сделать умножение «×» и деление «/»
- Проверить все четыре кнопки

5. Переменная-счётчик (15 минут)

Преподаватель делает пошагово → ученики повторяют:

Шаг 1: Создать переменную

- Из раздела «Variables» взять «initialize global name to» и назвать «schetchik»
- Присоединить число 0

Шаг 2: Увеличивать при каждом вычислении

- В каждую кнопку добавить «set global schetchik to» «get global schetchik» + 1

Шаг 3: Показывать значение

- Добавить Label2 и выводить в неё «join» «Вычислений: » и значение переменной
- Проверить – счётчик растёт

6. Красивый результат (10 минут)

- Использовать «join», чтобы выводить «5 + 7 = 12», а не только число
- Преподаватель помогает собрать длинную склейку из пяти частей

7. Подведение итогов (5 минут)

- Преподаватель спрашивает: зачем нужна переменная, если есть надпись?
- Анонсирует: на следующем уроке научим калькулятор принимать решения

Результат урока:

- Все обучающиеся создали приложение «Калькулятор»
- Освоили математические блоки
- Научились создавать и изменять глобальные переменные

УРОК 6. УСЛОВИЯ. ВЕТВЛЕНИЕ В ПРОГРАММЕ

Длительность: 2 академических часа (90 минут)

Формат: синхронная работа – преподаватель показывает, ученики сразу повторяют

ХОД ЗАНЯТИЯ:

1. Введение: условия в жизни и в программе (5 минут)

Преподаватель рассказывает:

- «Если идёт дождь – возьму зонт, иначе – кепку»
- Блок «if – then – else» (если – то – иначе)

2. Дизайн игры «Угадай число» (10 минут)

Преподаватель показывает → ученики повторяют:

- Создать проект UgadaiChislo
- Добавить Label «Я загадал число от 1 до 100. Угадай!»
- Добавить TextBox (NumbersOnly) и Button «Проверить»

- Добавить Label2 для подсказок – FontSize 22

3. Загадываем число (10 минут)

Преподаватель делает → ученики повторяют:

- Создать переменную «zagadka» со значением 0
- Взять событие «when Screen1.Initialize» (при запуске экрана)
- Внутрь: «set global zagadka to» «random integer from 1 to 100»

Перерыв 10 минут

4. Проверка ответа (25 минут)

Преподаватель делает пошагово → ученики повторяют каждый шаг:

Шаг 1: Блок «if»

- «when Button1.Click» → блок «if» из раздела «Control»
- Нажать на синюю шестерёнку и добавить два «else if» и один «else»

Шаг 2: Сравнения

- Первое условие: TextBox1.Text < zagadka → Label2.Text = «Больше!»
- Второе условие: TextBox1.Text > zagadka → Label2.Text = «Меньше!»
- Иначе: Label2.Text = «Угадал!»

Шаг 3: Проверка

- Сыграть несколько раз, проверить все подсказки

5. Счётчик попыток и новая игра (15 минут)

Преподаватель показывает → ученики повторяют:

- Создать переменную «popytki», увеличивать её при каждой проверке
- При угадывании выводить «Угадал за N попыток!»
- Добавить кнопку «Новая игра»: новое случайное число, попытки = 0, очистить поле

6. Защита от пустого ввода (5 минут)

- Преподаватель показывает блок «is empty»: если поле пустое – подсказать «Введите число»

7. Подведение итогов (5 минут)

- Обучающиеся играют в игры друг друга (называют числа вслух)
- Преподаватель спрашивает, чем отличаются «if» и «if – else»

Результат урока:

- Все обучающиеся создали игру «Угадай число»
- Освоили условия, сравнения и случайные числа
- Научились использовать событие запуска экрана

УРОК 7. ХОЛСТ И РИСОВАНИЕ. СОБЫТИЯ КАСАНИЯ

Длительность: 2 академических часа (90 минут)

Формат: синхронная работа – преподаватель показывает, ученики сразу повторяют

ХОД ЗАНЯТИЯ:

1. Введение: холст и координаты (5 минут)

Преподаватель объясняет и показывает:

- Canvas (холст) – область, на которой можно рисовать
- Координаты на холсте: точка (0, 0) в левом верхнем углу, X растёт вправо, Y – вниз
- Сравнение с Scratch, где (0, 0) в центре

2. Дизайн «Рисовалки» (10 минут)

Преподаватель показывает → ученики повторяют:

- Создать проект Risovalka
- Из палитры «Drawing and Animation» добавить Canvas
- Width – «Fill parent», Height – 300 пикселей
- Под холстом – HorizontalArrangement с кнопками «Красный», «Синий», «Зелёный», «Очистить»
- Покрасить кнопки в соответствующие цвета

3. Рисование пальцем (15 минут)

Преподаватель делает → ученики повторяют:

- Взять событие «when Canvas1.Dragged» (когда по холсту ведут пальцем)
- Внутрь: «call Canvas1.DrawLine»
- $x1 = \text{prevX}$, $y1 = \text{prevY}$, $x2 = \text{currentX}$, $y2 = \text{currentY}$
- Проверить – провести пальцем по холсту

Перерыв 10 минут

4. Цвета и очистка (15 минут)

Преподаватель делает → ученики повторяют:

- «when Button1.Click» → «set Canvas1.PaintColor to» красный цвет из раздела «Colors»
- Так же для синей и зелёной кнопки
- «when Button4.Click» → «call Canvas1.Clear»
- Проверить – нарисовать картинку тремя цветами

5. Точки по касанию и толщина линии (20 минут)

Преподаватель показывает → ученики повторяют:

- «when Canvas1.Touched» → «call Canvas1.DrawCircle» с x, y и радиусом 10
- Добавить Slider «Толщина»: MinValue 1, MaxValue 30
- «when Slider1.PositionChanged» → «set Canvas1.LineWidth to» thumbPosition
- Проверить – рисовать тонкой и толстой кистью

6. Фон холста (5 минут)

- Загрузить картинку-раскраску и выбрать её в свойстве BackgroundImage холста

7. Подведение итогов (5 минут)

- Обучающиеся показывают рисунки на камеру
- Преподаватель спрашивает, где на холсте точка (0, 0)

Результат урока:

- Все обучающиеся создали приложение «Рисовалка»
- Освоили компонент Canvas и события касания
- Научились работать с цветами и координатами на экране телефона

УРОК 8. СПРАЙТЫ И ДВИЖЕНИЕ ОБЪЕКТОВ

Длительность: 2 академических часа (90 минут)

Формат: синхронная работа – преподаватель показывает, ученики сразу повторяют

ХОД ЗАНЯТИЯ:

1. Введение (5 минут)

Преподаватель рассказывает:

- Спрайты в App Inventor: Ball (мяч) и ImageSprite (картинка)
- Спрайт может двигаться сам: у него есть скорость (Speed) и направление (Heading)

2. Мяч на холсте (15 минут)

Преподаватель показывает → ученики повторяют:

- Создать проект PrygayushchiyMyach
- Добавить Canvas: Width – «Fill parent», Height – «Fill parent»
- Добавить на холст Ball: Radius 20, PaintColor – красный
- Свойства Ball: Speed 5, Heading 45, Interval 50
- Проверить на телефоне – мяч улетает за край

3. Отскок от стен (10 минут)

Преподаватель делает → ученики повторяют:

- Взять событие «when Ball1.EdgeReached» (мяч коснулся края)
- Внутрь: «call Ball1.Bounce» с параметром «edge»
- Проверить – мяч прыгает по экрану

Перерыв 10 минут

4. Управление мячом (20 минут)

Преподаватель делает пошагово → ученики повторяют каждый шаг:

Шаг 1: Кнопки

- Под холстом добавить кнопки «Быстрее», «Медленнее», «Стоп»
- Уменьшить высоту холста, чтобы кнопки поместились

Шаг 2: Скорость

- «Быстрее»: «set Ball1.Speed to» Ball1.Speed + 2
- «Медленнее»: «set Ball1.Speed to» Ball1.Speed – 2

- «Стоп»: «set Ball1.Enabled to» false, повторное нажатие – true

Шаг 3: Проверка

- Проверить все кнопки

5. Касание мяча (15 минут)

Преподаватель показывает → ученики повторяют:

- «when Ball1.Touched» → менять цвет мяча на случайный: «make color» с тремя «random integer from 0 to 255»
- Добавить ImageSprite (картинка-препятствие) в центр холста
- «when Ball1.CollidedWith» → «call Ball1.Bounce» и звук удара (компонент Sound)

6. Подведение итогов (5 минут)

- Преподаватель спрашивает, что задают Speed, Heading и Interval
- Анонсирует: на следующем уроке сделаем настоящую игру

Результат урока:

- Все обучающиеся создали приложение «Прыгающий мяч»
- Освоили спрайты и их свойства движения
- Научились обрабатывать касания и столкновения объектов

УРОК 9. ТАЙМЕР. СОЗДАНИЕ ИГРЫ

Длительность: 2 академических часа (90 минут)

Формат: синхронная работа – преподаватель показывает, ученики сразу повторяют

ХОД ЗАНЯТИЯ:

1. Введение: игра «Поймай крота» (5 минут)

Преподаватель рассказывает правила:

- Крот появляется в случайном месте каждую секунду
- Нужно успеть коснуться крота – за это даётся очко
- На игру даётся 30 секунд

2. Дизайн игры (10 минут)

Преподаватель показывает → ученики повторяют:

- Создать проект RoymayKrota
- Добавить Label «Очки: 0» и Label «Время: 30»
- Добавить Canvas («Fill parent» по ширине, 350 пикселей по высоте) с фоном-полянкой
- Добавить на холст ImageSprite с картинкой крота (Width и Height – 60 пикселей)
- Добавить Button «Начать игру»

3. Компонент Clock – таймер (15 минут)

Преподаватель объясняет и показывает:

- Clock (часы) из палитры «Sensors» выполняет действие через заданный интервал
- Добавить Clock1: TimerInterval 1000 (1 секунда), TimerEnabled – выключить

Преподаватель делает → ученики повторяют:

- «when Clock1.Timer» → «call ImageSprite1.MoveTo»
- x – «random integer from 0 to» Canvas1.Width – ImageSprite1.Width
- y – «random integer from 0 to» Canvas1.Height – ImageSprite1.Height

Перерыв 10 минут

4. Очки (15 минут)

Преподаватель делает → ученики повторяют:

- Создать переменную «ochki» = 0
- «when ImageSprite1.Touched» → увеличить «ochki» на 1
- Обновить Label1: «join» «Очки: » и ochki
- Добавить вибрацию при попадании (компонент Sound, метод Vibrate 100)

5. Обратный отсчёт и конец игры (20 минут)

Преподаватель делает пошагово → ученики повторяют каждый шаг:

Шаг 1: Второй таймер

- Добавить Clock2: TimerInterval 1000, TimerEnabled – выключить
- Создать переменную «vremya» = 30

Шаг 2: Старт

- «when Button1.Click»: ochki = 0, vremya = 30, включить оба таймера

Шаг 3: Отсчёт

- «when Clock2.Timer»: уменьшить vremya на 1, обновить Label2
- «if» vremya = 0 → выключить оба таймера
- Добавить Notifier и вызвать «ShowAlert» с текстом «Игра окончена! Очки: ...»

6. Подведение итогов (5 минут)

- Ученики сравнивают свои рекорды
- Преподаватель спрашивает, зачем в игре два таймера

Результат урока:

- Все обучающиеся создали игру «Поймай крота»
- Освоили компонент Clock и случайное перемещение
- Научились программировать счёт очков и окончание игры

УРОК 10. ДАТЧИКИ МОБИЛЬНОГО УСТРОЙСТВА

Длительность: 2 академических часа (90 минут)

Формат: синхронная работа – преподаватель показывает, ученики сразу повторяют
ХОД ЗАНЯТИЯ:

1. Введение: какие датчики есть в телефоне (5 минут)

Преподаватель рассказывает:

- Акселерометр, компас, датчик освещённости, GPS
- Акселерометр сообщает, куда наклонён телефон
- Датчики работают только на настоящем телефоне, а не в эмуляторе

2. Что показывает акселерометр (15 минут)

Преподаватель показывает → ученики повторяют:

- Создать проект SharikNaklon
- Добавить три Label для значений X, Y, Z
- Из палитры «Sensors» добавить AccelerometerSensor
- «when AccelerometerSensor1.AccelerationChanged» → вывести xAccel, yAccel, zAccel в надписи
- Наклонять телефон и наблюдать, как меняются числа

Перерыв 10 минут

3. Шарик, которым управляет наклон (20 минут)

Преподаватель делает → ученики повторяют:

- Спрятать надписи (Visible – выключить)
- Добавить Canvas («Fill parent» по ширине и высоте) и Ball (Radius 20)
- В событие AccelerationChanged добавить «call Ball1.MoveTo»
- $x = \text{Ball1.X} - x\text{Accel} \times 2$, $y = \text{Ball1.Y} + y\text{Accel} \times 2$
- Проверить – шарик катится при наклоне; при необходимости поменять знак

4. Цель-лунка (15 минут)

Преподаватель показывает → ученики повторяют:

- Добавить ImageSprite «лунка» и поставить его в угол холста
- Создать переменную «ochki»
- «when Ball1.CollidedWith» → ochki + 1, звук и перенос лунки в случайное место
- Вывести очки в заголовок экрана: «set Screen1.Title to»

5. Встряхивание (10 минут)

Преподаватель показывает → ученики повторяют:

- Событие «when AccelerometerSensor1.Shaking» (телефон встряхнули)
- Внутри: вернуть шарик в центр и обнулить очки

6. Подведение итогов (5 минут)

- Преподаватель спрашивает, какие ещё приложения используют датчики

Результат урока:

- Все обучающиеся создали игру «Шарик-наклон»
- Познакомились с датчиками мобильного устройства
- Научились управлять объектом с помощью наклона телефона

УРОК 11. СПИСКИ И СЛУЧАЙНЫЙ ВЫБОР

Длительность: 2 академических часа (90 минут)

Формат: синхронная работа – преподаватель показывает, ученики сразу повторяют

ХОД ЗАНЯТИЯ:

1. Введение: что такое список (5 минут)

Преподаватель рассказывает:

- Список – переменная, в которой хранится много значений по порядку
- Примеры: список покупок, список учеников, плейлист

2. Дизайн «Шара предсказаний» (10 минут)

Преподаватель показывает → ученики повторяют:

- Создать проект SharPredskazaniy
- Добавить Image с картинкой волшебного шара
- Добавить Label «Задай вопрос и встряхни телефон» и Label2 для ответа
- Добавить Button «Получить ответ»

3. Создание списка ответов (15 минут)

Преподаватель делает → ученики повторяют:

- Создать переменную «otvety»
- Из раздела «Lists» взять «make a list» и присоединить к переменной
- Шестерёнкой добавить слоты и вписать 6 ответов: «Да», «Нет», «Скорее всего», «Спроси позже» и т.д.

Перерыв 10 минут

4. Случайный ответ (15 минут)

Преподаватель делает → ученики повторяют:

- «when Button1.Click» → «set Label2.Text to» «pick a random item list» otvety
- Добавить TextToSpeech, чтобы ответ звучал вслух
- Добавить AccelerometerSensor: «when Shaking» – то же самое действие
- Проверить кнопкой и встряхиванием

5. Пополнение списка (20 минут)

Преподаватель делает пошагово → ученики повторяют каждый шаг:

Шаг 1: Поле ввода

- Добавить TextBox и кнопку «Добавить ответ»

Шаг 2: Добавление

- «call add items to list» otvety, item – TextBox1.Text

- Очистить поле после добавления

Шаг 3: Количество

- Показывать «length of list» – сколько ответов знает шар

6. Подведение итогов (5 минут)

- Обучающиеся задают вопросы шарам друг друга
- Преподаватель спрашивает, чем список отличается от обычной переменной

Результат урока:

- Все обучающиеся создали приложение «Шар предсказаний»
- Освоили создание списков и выбор случайного элемента
- Научились добавлять элементы в список во время работы приложения

УРОК 12. НЕСКОЛЬКО ЭКРАНОВ И СОХРАНЕНИЕ ДАННЫХ

Длительность: 2 академических часа (90 минут)

Формат: синхронная работа – преподаватель показывает, ученики сразу повторяют

ХОД ЗАНЯТИЯ:

1. Введение (5 минут)

Преподаватель рассказывает:

- Почему данные пропадают, когда закрываешь приложение
- TinyDB – маленькая база данных внутри телефона
- Зачем приложениям несколько экранов

2. Дизайн «Моих заметок» (10 минут)

Преподаватель показывает → ученики повторяют:

- Создать проект MoiZametki
- Добавить TextBox (MultiLine включить) и Button «Сохранить»
- Добавить ListView – список заметок на экране
- Из палитры «Storage» добавить TinyDB

3. Сохранение заметок (20 минут)

Преподаватель делает пошагово → ученики повторяют каждый шаг:

Шаг 1: Список заметок

- Создать переменную «zametki» = «create empty list»

Шаг 2: Кнопка «Сохранить»

- «add items to list» zametki – TextBox1.Text
- «call TinyDB1.StoreValue» tag «zametki», valueToStore – переменная zametki
- «set ListView1.Elements to» zametki, очистить поле

Шаг 3: Загрузка при запуске

- «when Screen1.Initialize» → «set global zametki to» «call TinyDB1.GetValue» tag «zametki», valueIfTagNotThere – «create empty list»
- Показать заметки в ListView

- Проверить: добавить заметки, закрыть и снова открыть приложение

Перерыв 10 минут

4. Второй экран (20 минут)

Преподаватель показывает → ученики повторяют:

- Нажать «Add Screen» и назвать экран Screen2
- На Screen2 – Label «О приложении», текст об авторе, кнопка «Назад»
- «when Button1.Click» на Screen2 → «close screen»
- На Screen1 добавить кнопку «О приложении»: «open another screen screenName» «Screen2»
- Проверить переходы между экранами

5. Удаление заметки (10 минут)

- «when ListView1.AfterPicking» → «remove list item» zametki, index – ListView1.SelectionIndex
- Снова сохранить список в TinyDB и обновить ListView

6. Подведение итогов (5 минут)

- Преподаватель спрашивает, где хранятся заметки и почему они не пропадают
- Анонсирует: на следующем уроке начинаем итоговый проект

Результат урока:

- Все обучающиеся создали приложение «Мои заметки»
- Научились сохранять данные в TinyDB
- Освоили работу с несколькими экранами

УРОК 13. ПОДГОТОВКА ПРОЕКТА. СОЗДАНИЕ ПРИЛОЖЕНИЯ-ВИКТОРИНЫ

Длительность: 2 академических часа (90 минут)

Формат: синхронная работа – преподаватель показывает, ученики повторяют и дополняют проект своими идеями

ХОД ЗАНЯТИЯ:

1. Выбор темы викторины (5 минут)

Преподаватель рассказывает:

- Итоговый проект – приложение-викторина на любую тему: животные, космос, игры, мультфильмы
- Каждый выбирает тему и придумывает 5 вопросов с короткими ответами

2. Стартовый экран (10 минут)

Преподаватель показывает → ученики повторяют:

- Создать проект Viktorina
- На Screen1: название викторины, картинка по теме, кнопка «Начать»

- Добавить экран Screen2, по кнопке «Начать» открыть его

3. Списки вопросов и ответов (15 минут)

Преподаватель делает → ученики повторяют:

- На Screen2 создать переменные «voprosy» и «otvety» – два списка по 5 элементов
- Создать переменные «nomer» = 1 и «ochki» = 0
- Добавить Label для вопроса, TextBox для ответа, кнопку «Ответить», Label для результата
- «when Screen2.Initialize» → показать «select list item» voprosy, index – nomer

Перерыв 10 минут

4. Проверка ответа (25 минут)

Преподаватель делает пошагово → ученики повторяют каждый шаг:

Шаг 1: Сравнение

- «when Button1.Click» → «if» «downcase» TextBox1.Text = «select list item» otvety, index nomer
- Блок «downcase» переводит ответ в маленькие буквы, чтобы «Кот» и «кот» совпадали

Шаг 2: Реакция

- Правильно: ochki + 1, надпись «Верно!», звук или TextToSpeech
- Неправильно: надпись «Неверно, правильный ответ: ...»

Шаг 3: Следующий вопрос

- nomer + 1; если nomer > «length of list» voprosy – показать итог через Notifier: «Твой результат: ... из 5»
- Иначе показать следующий вопрос и очистить поле

5. Оформление и сборка приложения (10 минут)

Преподаватель показывает → ученики повторяют:

- Настроить цвета, шрифты и картинки по теме викторины
- Нажать «Build» → «Android App (.apk)» и отсканировать QR-код телефоном
- Установить своё приложение на телефон

6. Подведение итогов (5 минут)

- Преподаватель проверяет, у всех ли работает викторина
- Напоминает требования к защите проекта
- Анонсирует следующий урок – защиту проекта

Результат урока:

- Все обучающиеся создали приложение-викторину
- Применили списки, переменные, условия и несколько экранов в одном проекте
- Установили собственное приложение на телефон

УРОК 14. ИТОГОВЫЙ КОНТРОЛЬ

Длительность: 2 академических часа (90 минут)

Формат: презентация и обсуждение проектов

ХОД ЗАНЯТИЯ:

1. Объяснение процедуры защиты

Преподаватель рассказывает:

- Каждый покажет свою викторину на камеру или через демонстрацию экрана
- Расскажет, что сделал
- Ответит на 1–2 вопроса
- Время на одного – до 10 минут

2. Защита проектов

Каждый обучающийся по очереди:

- Запускает свою викторину на телефоне
- Проходит 2–3 вопроса, показывает правильный и неправильный ответ
- Показывает итоговый результат и переход между экранами

Рассказ:

- Какую тему выбрал
- Что было сложно
- Чем гордится в проекте

Вопросы преподавателя:

- Где хранятся вопросы и ответы?
- Как программа понимает, что ответ правильный?
- Зачем нужна переменная `nomer`?

Обратная связь от преподавателя:

- Что получилось хорошо
- Одна рекомендация для улучшения

3. Подведение итогов курса

Преподаватель:

- Поздравляет всех с завершением курса
- Перечисляет, чему научились:
 - Интерфейс App Inventor и подключение телефона
 - Компоненты и их расположение
 - Звук, речь и датчики
 - Переменные, условия и списки
 - Холст, спрайты и таймеры
 - Несколько экранов и сохранение данных
 - Сборка и установка собственного приложения

- Рассказывает, как продолжать: галерея проектов и уроки на сайте App Inventor
- Отвечает на финальные вопросы

Результат урока:

- Все презентовали свои проекты
- Получили обратную связь
- Подвели итоги обучения

5. Оценка качества освоения программы

5.1. Формы аттестации

Оценка качества освоения программы включает текущий контроль и итоговую аттестацию и проводится в соответствии с Положением о формах, периодичности и порядке текущего контроля успеваемости, промежуточной и итоговой аттестации обучающихся.

Текущий контроль проводится на каждом занятии в формате онлайн-практики с преподавателем без выставления отметок (безотметочная система) в форме:

- наблюдения преподавателя за работой обучающихся в режиме реального времени;
- проверки правильности выполнения практических заданий непосредственно на занятии;
- оценки работоспособности созданных приложений;
- оценки активности и самостоятельности обучающихся.

По результатам текущего контроля преподаватель даёт обучающемуся устную обратную связь: что получилось, над чем стоит поработать.

Итоговая аттестация проводится по завершении обучения в форме защиты итогового проекта на онлайн-занятии.

5.2. Форма итоговой аттестации

Защита итогового проекта — создание и презентация приложения-викторины в App Inventor с демонстрацией его работы на мобильном устройстве в прямом эфире, объяснением функционала и особенностей реализации, ответами на вопросы преподавателя.

Результат итоговой аттестации определяется по уровню освоения программы (высокий, средний, начальный) в соответствии с критериями, приведёнными в разделе 7.2. Обучающийся, показавший высокий, средний или начальный уровень, считается освоившим программу («освоил»). Если итоговый проект не выполнен или не работает, результат — «не освоил».

6. Условия реализации образовательной программы

6.1. Требования к минимальному материально-техническому обеспечению

Технические требования:

- компьютер или ноутбук;
- смартфон или планшет на Android с установленным бесплатным приложением MIT AI2 Companion (при отсутствии мобильного устройства — эмулятор на компьютере; работа с датчиками возможна только на мобильном устройстве);
- стабильное интернет-соединение (скорость не менее 5 Мбит/с);
- веб-камера и микрофон для участия в онлайн-занятиях.

Программное обеспечение:

- онлайн-среда разработки MIT App Inventor 2 (<https://appinventor.mit.edu>), вход без регистрации по коду доступа (<https://code.appinventor.mit.edu>);
- приложение MIT AI2 Companion для мобильного устройства;
- программа для видеосвязи «Яндекс Телемост».

6.2. Организационно-педагогические условия реализации программы

Кадровое обеспечение:

Для реализации образовательной программы привлекаются педагоги дополнительного образования, соответствующие требованиям профессионального стандарта «Педагог дополнительного образования детей и взрослых» и имеющие:

- высшее образование или среднее профессиональное образование в рамках укрупнённых групп специальностей и направлений подготовки по профилю «информатика», «программирование», «компьютерные науки» либо педагогическое образование с дополнительной подготовкой в области программирования;
- или высшее образование либо среднее профессиональное образование в рамках иных укрупнённых групп специальностей и направлений подготовки при условии его соответствия дополнительным общеобразовательным общеразвивающим программам, реализуемым организацией, и получение при необходимости после трудоустройства дополнительного профессионального образования педагогической направленности;
- или успешное прохождение промежуточной аттестации не менее чем за два года обучения по образовательным программам высшего образования по специальностям и направлениям подготовки, соответствующим направленности дополнительных общеобразовательных программ.

Требования к опыту практической работы педагогов: не менее одного года в должности педагога дополнительного образования, иной должности педагогического работника — для старшего педагога дополнительного образования.

Информационное обеспечение:

Обучающиеся получают доступ к:

- онлайн-занятиям в режиме реального времени;
- записям всех занятий для повторного просмотра;
- медиафайлам (картинкам и звукам) для проектов каждого урока.

7. Фонд оценочных материалов

7.1. Задания для текущего контроля

Текущий контроль осуществляется на каждом занятии в формате онлайн-практики без выставления отметок. Преподаватель наблюдает за работой обучающихся в режиме реального времени, оценивает правильность выполнения практических заданий, помогает с возникающими трудностями и даёт обратную связь.

Критерии текущего контроля:

- правильность повторения действий за преподавателем;
- темп работы (успевает ли обучающийся следовать за преподавателем);
- работоспособность создаваемого приложения;
- умение исправлять ошибки с подсказками преподавателя.

Ниже приведены описания проектов, которые создаются на каждом занятии в формате «преподаватель показывает → ученики повторяют».

Урок 1. Знакомство с App Inventor

Проект: «Привет!»

Обучающиеся создают первое приложение с кнопкой и надписью, подключают телефон и проверяют приложение.

Урок 2. Компоненты интерфейса

Проект: «Моя визитка»

Обучающиеся проектируют интерфейс с изображением, надписями и кнопками, используют расположения, показывают и скрывают компоненты.

Урок 3. Звук и медиафайлы

Проект: «Говорящий кот»

Обучающиеся загружают медиафайлы, программируют звук, вибрацию, проигрыватель музыки и регулятор громкости.

Урок 4. Ввод текста и синтез речи

Проект: «Говорилка»

Обучающиеся используют текстовые поля, склеивание строк и синтез речи с настройкой голоса.

Урок 5. Переменные и арифметика

Проект: «Калькулятор»

Обучающиеся программируют арифметические операции и счётчик вычислений с помощью глобальной переменной.

Урок 6. Условия

Проект: «Угадай число»

Обучающиеся программируют игру со случайным числом, сравнениями, ветвлением и счётчиком попыток.

Урок 7. Холст и рисование

Проект: «Рисовалка»

Обучающиеся программируют рисование пальцем на холсте, выбор цвета, толщины линии и очистку.

Урок 8. Спрайты и движение

Проект: «Прыгающий мяч»

Обучающиеся программируют движение мяча, отскок от краёв, управление скоростью и столкновения.

Урок 9. Таймер

Проект: «Поймай крота»

Обучающиеся создают игру с таймерами, случайным перемещением спрайта, подсчётом очков и окончанием игры.

Урок 10. Датчики

Проект: «Шарик-наклон»

Обучающиеся используют акселерометр для управления шариком наклоном телефона и встряхивание для сброса игры.

Урок 11. Списки

Проект: «Шар предсказаний»

Обучающиеся создают список ответов, выбирают случайный элемент и добавляют новые элементы в список.

Урок 12. Экраны и хранение данных

Проект: «Мои заметки»

Обучающиеся сохраняют заметки в TinyDB, выводят их списком и создают второй экран.

Урок 13. Приложение-викторина

Проект: «Викторина»

Обучающиеся создают комплексный проект со стартовым экраном, списками вопросов и ответов, проверкой ответов, подсчётом баллов и сборкой приложения.

7.2. Задание для итоговой аттестации

Урок 14. Итоговый контроль. Защита итогового проекта: приложение-викторина

Описание задания:

Обучающийся презентует созданное на уроке 13 приложение-викторину, демонстрируя его работу на мобильном устройстве (или в эмуляторе) и объясняя принципы реализации.

Обязательные элементы проекта:

1. Не менее 5 вопросов и ответов, хранящихся в списках.
2. Проверка ответа с помощью условия и подсчёт баллов в переменной.
3. Не менее двух экранов (стартовый экран и экран вопросов).
4. Вывод итогового результата после последнего вопроса.
5. Оформление интерфейса по теме викторины (картинки, цвета) и звуковая или речевая реакция на ответ.

Критерии оценивания по уровням освоения программы:

Уровень	Критерии
Высокий	Приложение полностью работает на мобильном устройстве; выполнены все обязательные элементы, большая часть — самостоятельно; добавлены собственные улучшения (дополнительные вопросы, картинки к вопросам, перезапуск викторины); обучающийся уверенно объясняет, как устроена программа, и отвечает на вопросы преподавателя.
Средний	Приложение работает; выполнены все или почти все обязательные элементы (не менее 4 из 5), допускаются небольшие недочёты; часть работы выполнена с помощью преподавателя; обучающийся объясняет основные блоки программы с помощью наводящих вопросов.
Начальный	Приложение запускается; выполнено не менее 3 из 5 обязательных элементов, в основном по образцу или с помощью преподавателя; обучающийся затрудняется в объяснении, но может показать, как работает приложение.
Не освоил	Приложение не выполнено, не запускается или выполнено менее 3 обязательных элементов.

Итог аттестации: высокий, средний или начальный уровень — «освоил» (выдаётся сертификат о прохождении обучения); проект не выполнен или не работает — «не освоил» (по заявлению выдаётся справка об обучении).

8. Электронная библиотека

1. MIT App Inventor — официальный сайт среды разработки. <https://appinventor.mit.edu>
2. MIT App Inventor. Учебные материалы и пошаговые уроки (Tutorials). <https://appinventor.mit.edu/explore/ai2/tutorials>
3. MIT App Inventor. Справочник компонентов (Component Reference). <https://ai2.appinventor.mit.edu/reference/components/>
4. MIT App Inventor. Справочник блоков (Built-in Blocks). <https://ai2.appinventor.mit.edu/reference/blocks/>
5. Сообщество MIT App Inventor (форум, примеры проектов). <https://community.appinventor.mit.edu>

Ссылки на материалы и медиафайлы к урокам размещаются в личном кабинете обучающегося на образовательной платформе. Материалы сайта App Inventor доступны на английском языке; преподаватель разбирает их на занятиях.