

МИНИСТЕРСТВО ОБРАЗОВАНИЯ, НАУКИ И МОЛОДЕЖНОЙ ПОЛИТИКИ  
НИЖЕГОРОДСКОЙ ОБЛАСТИ

Государственное бюджетное профессиональное образовательное учреждение  
«АРЗАМАССКИЙ ТЕХНИКУМ СТРОИТЕЛЬСТВА И ПРЕДПРИНИМАТЕЛЬСТВА»  
Центр цифрового образования детей «IT-куб»

СОГЛАСОВАНО:

Начальник ЦЦОД «IT-Куб»

Г.А. Волгунова

(подпись)

(Ф.И.О.)

« 10 » 07 20 20 г.

УТВЕРЖДАЮ:

Директор ГБПОУ АТСП

О.Н. Бабушкин

(Ф.И.О.)

(подпись)

Приказ № 200 от « 10 » 07 2020 г.



ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ

ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

«Мобильная разработка. IT школа Samsung»

*Направленность – техническая*

**Возраст обучающихся: 14-17 лет**

**Срок реализации: 1 год**

**Объем: 132 часа**

**Авторы-составители:**

Волгина Ирина Александровна,  
педагог дополнительного образования,  
Морозова Надежда Ивановна,  
методист

Арзамас

2020

## Содержание

1. Комплекс основных характеристик программы.....	2
1.1. Пояснительная записка .....	2
1.2. Цель и задачи программы .....	11
1.3. Содержание общеразвивающей программы .....	13
1.3.1. Содержание учебного плана.....	21
1.4. Планируемые результаты .....	31
2. Комплекс организационно – педагогических условий реализации общеразвивающей программы.....	33
2.1. Календарный учебный график на 2020-2021 учебный год .....	33
2.2. Условия реализации программы .....	34
2.3. Формы аттестации и оценочные материалы.....	36
2.4. Методические материалы .....	37
Список литературы .....	40
Приложение 1 .....	43
Приложение 2 .....	46
Приложение 3 .....	48
Приложение 4 .....	52
Приложение 5 .....	55
Приложение 6 .....	59
Приложение 7 .....	61
Приложение 8 .....	63

## **1. Комплекс основных характеристик программы**

### **1.1. Пояснительная записка**

На сегодня разработка программного обеспечения является наиболее востребованным направлением в любых сферах применения. Кроме того, большое развитие мобильных платформ дает более широкий выбор направлений разработки.

Мобильные устройства работают на различных операционных системах, но самыми распространенными и открытыми для программирования являются устройства на популярной платформе Android.

В современном мире Java как платформа является наиболее популярной, так как не имеет требований к операционной системе для запуска своих приложений. Кроме того, мобильные устройства на самой популярной ОС Android в большинстве случаев используют приложения, написанные именно на этой платформе. Изучение языка программирования Java по данной программе дает возможность пользователю мобильного устройства с ОС Android вести разработку очень широкого круга программных продуктов, начиная от игровых приложений для мобильных телефонов, и заканчивая профессиональными системами, использующими самые современные технологии.

Дополнительная общеобразовательная общеразвивающая программа «Мобильная разработка. ИТ школа Samsung» была составлена на основе программы «ИТ ШКОЛА SAMSUNG» компании Samsung Electronics, которая была разработана Samsung R&D InstituteRus (Исследовательский Центр Samsung) при участии Московского физико-технического института (МФТИ).

Программа «ИТ ШКОЛА SAMSUNG» компании — Samsung Electronics - это долгосрочная инициатива, которая реализуется при поддержке Министерства Образования и Науки РФ. В рамках данной программы уже

более 2800 школьников получили сертификат Samsung, успешно завершив программу дополнительного образования в области ИТ и программирования за 5 лет.

Дополнительная общеобразовательная общеразвивающая программа «Мобильная разработка. ИТ школа Samsung» имеет **техническую направленность**, ориентирована на развитие навыков программирования и проектирования программ под платформу Android.

Основанием для проектирования и реализации данной общеобразовательной программы служит *перечень следующих нормативных правовых актов и государственных программных документов:*

- Федеральный закон «Об образовании в Российской Федерации» от 29.12.2012 N 273-ФЗ; Стратегия развития воспитания в Российской Федерации на период до 2025 года. Распоряжение Правительства Российской Федерации от 29 мая 2015 г. № 996-р;
- Письмо Министерства образования и науки РФ от 18.11.2015г. № 09-3242. «О направлении Методических рекомендаций по проектированию дополнительных общеобразовывающих программ (включая разноуровневые)»;
- Распоряжение правительства РФ от 04.09. 2014 № 1726-р «Об утверждении Концепции развития дополнительного образования детей»;
- «Основы законодательства РФ об охране здоровья граждан», утвержденные Верховным советом РФ от 22.07.1993 № 5487 - (ред. от 25.11.2009);
- Федеральный закон от 24.07.1998 № 124-ФЗ «Об основных гарантиях прав ребенка в РФ»; Федеральный закон «Об основах охраны здоровья граждан в Российской Федерации», 2011г.;
- Приказ Министерства просвещения России от 09.11.2018 г. № 196 «Об утверждении Порядка организации и осуществления образовательной

деятельности по дополнительным общеобразовательным программам» (Приказ №1008 отменен).

**Актуальность** программы обусловлена потребностью общества в технически грамотных специалистах и полностью отвечает социальному заказу по подготовке квалифицированных кадров в области мобильной разработки. Данная программа является единственным в своем роде экспериментом в связи с востребованностью на рынке и отсутствием программ образования в данном направлении для школьников, не имеет аналогов на рынке общеобразовательных услуг и является своего рода уникальным образовательным продуктом в области информационных технологий.

**Прогностичность** дополнительной общеобразовательной общеразвивающей программы «Мобильная разработка. IT школа Samsung» заключается в том, что она отражает требования и актуальные тенденции не только сегодняшнего, но и завтрашнего дня, а также имеет междисциплинарный характер, что полностью отражает современные тенденции построения как дополнительных общеобразовательных программ, так и образования в целом. Также данная программа является базой для перехода на более сложные программы обучения. Обучающиеся приобретают знания по основам IT, которые будут востребованы для дальнейшего обучения в профильных средних специальных и высших учебных заведениях, а также, по окончанию программы, получают сертификат от компании «Samsung», признанного лидера в сфере IT инноваций.

**Отличительная особенность программы** «Мобильная разработка. IT школа Samsung» заключается в комплексном подходе в построении программы: охвачены минимально необходимые базовые знания из разных областей IT и программирования с учетом уровня подготовки обучающихся, а также в ориентации на различный уровень подготовки. В программе предусмотрено проектное обучение: разработка обучающимися

индивидуального проекта в виде приложения для мобильных платформ на ОС Android.

*Адресат общеразвивающей программы.* Дополнительная общеобразовательная общеразвивающая программа «Мобильная разработка. ИТ школа Samsung» предназначена для детей в возрасте 14-17 лет, без ограничений возможностей здоровья. Также программа ориентирована на обучающихся, которые:

- имеют склонность к алгоритмическому мышлению;
- увлекаются ИТ-технологиями;
- владеют хотя бы одним языком программирования на уровне знания условных и циклических конструкций;
- имеют устойчивые знания по школьному курсу математики за 1-8 класс.

Группы формируются *по возрасту*: 14-17 лет. Формы занятий групповые, количество обучающихся в группе – 8-12 человек. Состав групп постоянный.

*Место проведения* занятий: 607220, г. Арзамас, ул. Мира, д.1.

*Возрастные особенности группы.* Содержание программы учитывает возрастные и психологические особенности обучающихся в возрасте 14-17 лет, которые определяют выбор форм проведения занятий с ними. Выделенные нами возрастные периоды при формировании групп 14-17 лет базируются на психологических особенностях развития старшего подросткового возраста.

Обучающиеся этого возраста отличаются внутренней уравновешенностью, стремлением к активной практической деятельности, поэтому основной формой проведения занятий выбраны практические занятия. Их также увлекает совместная, коллективная деятельность, так как резко возрастает значение коллектива, общественного мнения, отношений со сверстниками, оценки поступков и действий не только старших, но и

сверстников. Они стремятся завоевать авторитет, занять достойное место в коллективе. Поэтому в программу включены практические занятия соревновательного характера, которые позволяют каждому проявить себя и найти свое место в коллективе.

Также следует отметить, что обучающиеся данной возрастной группы характеризуются такими психическими процессами, как изменение структуры личности и возникновение интереса к ней, развитие абстрактных форм мышления, становление более осознанного и целенаправленного характера деятельности, проявление стремления к самостоятельности и независимости, формирование самооценки. Эти процессы позволяют положить начало формированию начального профессионального самоопределения.

***Режим занятий, объем общеразвивающей программы:*** длительность одного занятия составляет 2 академических часа (по 45 минут) с перерывом (переменой) в 10 минут, периодичность занятий – 2 раза в неделю.

***Срок освоения общеразвивающей программы*** определяется содержанием программы и составляет 1 год (**132 часа**).

***Формы обучения и виды занятий:*** сочетание очной формы образования с применением дистанционных образовательных технологий (Закон №273-ФЗ, гл.2, ст.17, п.2.).

Учебный процесс строится таким образом, чтобы экспериментальная и практическая работы преобладали над теоретической подготовкой. Необходимые для работы теоретические сведения находятся на каждом персональном компьютере в специальной папке, даются педагогом перед началом практических занятий. Индивидуальная работа проводится во время практических занятий – при выполнении задания у каждого обучающегося возникают свои вопросы. Групповая работа проводится во время теоретических занятий. Каждая тема по программированию сопровождается наглядной демонстрацией работы алгоритма для того, чтобы учащиеся представляли работоспособность алгоритма, а также к чему им нужно

стремиться при выполнении поставленной задачи. Учебный процесс организуется на основе постепенного усложнения учебного материала, как теоретического, так и практического.

Виды деятельности обучающегося в процессе обучения:

- освоение теоретического и практического материала на занятиях;
- самостоятельная практическая работа: выполнение домашних заданий, мини-проектов (небольшие приложения, которые реализуются учениками преимущественно на занятиях совместно с учителем с небольшими самостоятельными доработками в качестве домашнего задания);
- разработка индивидуального проекта;
- участие в вебинарах;
- промежуточная аттестация по модулям в форме электронного тестирования.

Примерные варианты тестовых заданий по каждому модулю представлены в приложениях.

По типу организации взаимодействия педагогов с обучающимися при реализации дополнительной общеобразовательной общеразвивающей программы используются личностно-ориентированные технологии, технологии сотрудничества.

Реализация программы предполагает использование здоровьесберегающих технологий.

Здоровьесберегающая деятельность реализуется:

- через создание безопасных материально-технических условий;
- включением в занятие динамических пауз, периодической смены деятельности слушателей;
- контролем соблюдения обучающимися правил работы на ПК;
- через создание благоприятного психологического климата в учебной группе в целом.

***Объем общеразвивающей программы:*** 132 часа.

Форма организации образовательной деятельности – групповая.

***Программа является одногодичной, одноуровневой, модульной.***

Каждый модуль направлен на формирование определенных компетенций.

Зачисление детей на обучение по данной программе производится по рейтингу. Рейтинг определяется путем сдачи предварительного тестирования, которое проходит на платформе «IT школа Samsung». Примерные задания тестирования представлены в Приложении 1.

Модуль – структурная единица образовательной программы, имеющая определенную логическую завершенность по отношению к результатам обучения. Каждый модуль состоит из кейсов (не менее двух), направленных на формирование определенных компетенций (hard и soft). Результатом каждого кейса является «продукт» (групповой, индивидуальный), демонстрирующий сформированность компетенций.

Кейс – история, описывающая реальную ситуацию, которая требует проведения анализа, выработки и принятия обоснованных решений. Кейс включает в себя набор специально разработанных учебно-методических материалов. Кейсовые «продукты» могут быть самостоятельным проектом по результатам освоения модуля или общего проекта по результатам всей образовательной программы.

Модули и кейсы различаются по сложности и реализуются по принципу «от простого к сложному».

***Программа обучения состоит из модулей:***

1. Основы программирования на языке Java;
2. Введение в объектно-ориентированное программирование;
3. Основы программирования Android приложений;
4. Алгоритмы и структуры данных на языке;
5. Основы разработки серверной части мобильных приложений.

***Педагогическая целесообразность программы.*** Программа реализует профориентационные задачи, обеспечивает знакомство с современными

профессиями в сфере ИТ. Программа предполагает использование форм организации материала, обеспечивающих доступ к сложным и нетривиальным разделам в рамках содержательно-тематического направления дополнительной общеобразовательной общеразвивающей программы. Также предполагает углубленное изучение содержания программы и доступ к околопрофессиональным и профессиональным знаниям в рамках содержательно-тематического направления.

Осваивая данную дополнительную общеобразовательную общеразвивающую программу, обучающиеся будут овладевать навыками, востребованными на рынке труда. Практически для каждой перспективной профессии будут полезны знания и навыки, рассматриваемые в данной программе. Программа помогает решать проблемы личностного и профессионального самоопределения, самореализации обучающихся.

### ***Аннотация***

Программа «Мобильная разработка» была составлена при поддержке © Samsung R&D Institute Rus (Исследовательский Центр Samsung) и при участии Московского физико-технического института (МФТИ).

Проект «IT-школа Samsung» компании Samsung Electronics – это долгосрочная инициатива, которая реализуется при поддержке Министерство просвещения РФ. В ее рамках запланировано бесплатное обучение более 5 тысяч школьников в 40 регионах России по программе дополнительного образования технической направленности в области ИТ и программирования на языке Java.

Данная программа является единственным в своем роде экспериментом в связи с востребованностью на рынке и отсутствием программ образования в данном направлении для школьников. Особенность программы «Мобильная разработка» заключается в изучении основ языка программирования Java и структуры приложения под ОС Android и строится в доступной и понятной для обучающихся среде, т. е. программирование

ведется в текстово-графическом режиме, что позволяет сразу задавать необходимый функционал для элементной базы приложения.

Программа рассчитана на обучающихся 14-17 лет.

## **1.2. Цель и задачи программы**

**Цель программы:** формирование технической грамотности средствами приобщения обучающихся к разработке программ под современную платформу Android.

Для успешной реализации поставленной цели необходимо решить ряд педагогических, развивающих и воспитательных задач:

***Обучающие:***

- расширение знаний о современных и популярных платформах;
- формирование представления о процессе проектирования информационных систем;
- формирование начальных навыков программирования;
- обучение объектно-ориентированному подходу в проектировании и разработке программного обеспечения;
- знакомство с архитектурой приложения под ОС Android;
- обучение программированию технических устройств.

***Развивающие:***

- формирование алгоритмического мышления, навыков работы с информацией;
- формирование умения самостоятельно решать поставленную задачу, излагать мысли в четкой логической последовательности, отстаивать свою точку зрения, анализировать ситуацию и самостоятельно находить ответы на вопросы путем логических рассуждений;
- развитие логического и технического мышления обучающихся.

***Воспитательные:***

- формирование активной жизненной позиции, гражданско-патриотической ответственности;
- воспитание этики групповой работы, отношений делового сотрудничества, взаимоуважения, развитие основ коммуникативных отношений внутри проектных групп и в коллективе в целом;

- создание условий для развития устойчивой потребности в самообразовании; воспитание упорства в достижении результата целеустремленности, организованности;
- формирование, ответственного отношения к труду, толерантности и уважительного отношения к окружающим.

### 1.3. Содержание общеразвивающей программы

Таблица 1

№ п/п	Наименование разделов и тем	Краткое содержание темы	Количество часов		
			теория	практика	всего
<b>Модуль 1. Основы программирования на языке Java</b>			<b>8</b>	<b>12</b>	<b>20</b>
1-2	Вводное занятие Знакомство со средой разработки	Введение. Техника безопасности. Порядок создания, компиляции, сборки и запуска в IDE приложения Java. Порядок установки среды разработки на домашнем компьютере. Разбор Java- проекта.	1	1	2
3-4	Типы данных и операции	Понятия переменные, примитивные типы данных, арифметические операторы, выражения и присваивания. Написание простейших программ, объявляющих переменные целого, вещественного типа, присваивающих им значения.	1	1	2
5-6	Операции отношения и логические выражения	Тип Boolean. Операции отношения, логические и тернарные операции. Разбор программы, которая демонстрирует выполнение логических операций и операций отношений.	1	1	2
7-8	Условные конструкции	Понятие блока, область действия блоков. Условные конструкции: if-else, switch. Закрепление навыков написания всех ранее изученных операторов путем написания и вычисления выражений.	1	1	2
9-10	Итеративные конструкции while	Общие понятия. Конструкции while, do-while. Безусловные операторы перехода break. Разбор примеров, илюстрирующих использование операторов цикла. Написание собственного примера на использование операторов цикла и операторов безусловного перехода.	1	1	2

11-12	Итеративные конструкции for Массивы	Вложенные циклы. Безусловные операторы перехода break с меткой, continue. Одномерные массивы, цикл for each в Java. Разбор примеров нахождения максимума и минимума, поиска на числовых массивах.	1	1	2
13-14	Методы Видимость переменных	Изучение понятия функций на примере методов Java. Передача параметров, возвращение результата. Видимость переменных. Обзорная классификация переменных по области видимости: область класса, область метода, область блока. Разбор примера функции println. Реализация собственной функции для считывания и вывода массива с использованием уже существующих функций Java	1	1	2
15-16	Многомерные и неровные массивы	Многомерные массивы на примере двумерных. Матрицы. Особенность массивов в Java. Разбор примеров с матрицами: объявление, инициализация, обращение к элементам. Использование вложенного цикла for each.	1	1	2
17-20	Практикум	Решение практических задач по пройденным темам модуля 1. Подготовка к тестированию.		2	2
21-22	Контрольное тестирование по модулю 1	Тестовые задания по темам модуля 1.		2	2
<b>Модуль 2. Введение в объектно-ориентированное программирование</b>			<b>10</b>	<b>18</b>	<b>28</b>
23-24	Понятие класса и объекта	Цели и задачи ОО-подхода к проектированию и разработке ПО. Объект, сообщение, класс, экземпляр объекта, метод. Общее понятие о парадигмах ООП: абстракция, инкапсуляция, наследование, полиморфизм. Описание протокола класса. Практика: Проектирование и реализация простейшего класса. Обзор классов,	1	1	2

		соответствующих примитивным типам.			
25-26	Конструкторы Статические методы	Конструкторы и деструкторы. Статические методы. Открытые и закрытые поля. Доступ к полям объекта. Реализация конструкторов.	1	1	2
27-28	Работа с экземплярами класса Инициализация различных типов данных	Инициализация переменных, массивов, полей классов в конструкторе.		2	2
29-30	Перегрузка методов	Конструктор по умолчанию. Перегрузка методов на примере конструкторов.	1	1	2
31-32	Начальные приемы тестирования и отладки	Виды тестирования и отладки. Начальные приемы тестирования и отладки, сценарии тестирования на примерах.	1	1	2
33-34	Знакомство с Android разработкой	ОС Android. Среда разработки. Принципиальная архитектура Android-приложения. Создание первого Android приложения в IDE. Изучение жизненного цикла Activity.	1	1	2
35-36	Интерфейс Android приложения	Построение простейшего интерфейса пользователя. Язык разметки XML. Описание ресурсов Android с помощью XML. Разметки (Layouts) и их применение. Построение простейшего интерфейса пользователя. Создание собственной разметки.	1	1	2
37-38	Представления	Представления (Views) Создание приложения с простейшим интерфейсом и обработкой событий.	1	1	2
39-40	Наследование, инкапсуляция	Производные классы и наследование. Защищенные части классов и правила доступа для классов и объектов в Java. Сравнение иерархии классов и контейнеризации классов. Разбор примера, где одно из наследований заменено вложенностю классов.	1	1	2
41-42	Понятие контекста Намерения	Понятия контекста (Context) и намерения (Intent) в Android. Намерения (Intents) в Android.	1	1	2

		Разбор примеров и реализация собственного класса, аналог которого общеупотребителен в окружающем мире или в математике и имеет хорошо знакомое поведение.			
43-44	Полиморфизм	Полиморфизм. Полиморфные методы и позднее связывание в Java. Абстрактные методы и классы, интерфейсы. Создание класса, хранящего информацию о человеке (возраст, имя). Разбор примера реализация иерархии классов.	1	1	2
45-48	Практикум	Решение практических задач по пройденным темам модуля 2. Подготовка к тестированию.		4	4
49-50	Контрольное тестирование по модулю 2	Тестовые задания по темам модуля 2.		2	2
<b>Модуль 3. Основы программирования Android приложений</b>			<b>10</b>	<b>16</b>	<b>26</b>
51-52	Практикум ООП проектирования	Разбор кейсов проектирования архитектуры классов приложения.		2	2
53-54	Диаграммы UML	Принципы SOLID. Изучение простейших диаграмм UML. Построение собственной диаграммы классов.	1	1	2
55-56	Ввод, вывод Исключения	Библиотечные классы ввода-вывода. Обработка исключений и классы исключений. Стандартные исключения Java. Работа с файлами в Android. Работа с классом File, как пример необходимости обработки исключений.	1	1	2
57-58	Внутренние (вложенные) классы	Понятие внутреннего класса. Отличие от наследования. Назначение. Доступ к состоянию объекта с помощью внутреннего класса. Разбор примера с доступом к объекту с помощью внутреннего класса.	1	1	2
59-60	Анонимные классы	Локальные и анонимные внутренние классы. Сущность, синтаксис, назначение. Обработка событий пользовательского интерфейса. Краткий обзор классов и интерфейсов для обработки	1	1	2

		событий. Классы Listeners.			
61-62	Параллелизм и синхронизация, потоки	Процессы и потоки в Android. Классы AsyncTask и Thread. Реализация логики потоков. Синхронизация потоков. Разбор примера использования AsyncTask.	1	1	2
63-64	Фрагменты Сенсоры	Создание и управление фрагментами. Класс Fragment и его методы. Взаимодействие фрагментов и активностей. Типы сенсоров и обработка событий.	1	1	2
65-66	Двумерная графика в Android-приложениях	Двумерная графика в Android приложениях. Класс Canvas. Реализация простейшего приложения на Canvas.	1	1	2
67-68	Разработка игровых приложений	Этапы проектирования и реализации. Профессии в мире индустрии игр. Знакомство с классом SurfaceView. Обзор. Отличие View от SurfaceView. Особенность класса SurfaceView.	2		2
69-70	Реализация графики на основе SurfaceView	Понятие игрового движка. Реализация графики на основе SurfaceView.	1	1	2
71-74	Практикум	Решение практических задач по пройденным темам модуля 3. Подготовка к тестированию.		4	4
75-76	Контрольное тестирование по модулю 3	Тестовые задания по темам модуля 3.		2	2
<b>Модуль 4. Алгоритмы и структуры данных</b>			<b>14</b>	<b>18</b>	<b>32</b>
77-78	Массив Алгоритм двоичного поиска	Структуры данных. Сложность алгоритмов. Массив, как базовая структура данных. Алгоритм двоичного поиска: идея, применения, реализация.	1	1	2
79-80	Классы Arrays и ArrayList	Классы Arrays (массивы), ArrayList (неограниченный массив). Изучение класса Arrays: заполнение, копирование, сравнение, печать, методы поиска и др. общие методы. Изучение класса ArrayList. Создание итераторов, навигация.	1	1	2
81-82	Список	Список, как базовая структура данных: стеки, очереди,	2		2

		односвязные и двусвязные списки.			
83-84	Список LinkedList	Изучение класса LinkedList, реализующему связные списки Сравнение ArrayList и LinkedList. Разбор примера применения классов LinkedList, Queue и Stack.	1	1	2
85-86	Адаптеры в Андроид	Назначение и применение. Стандартные адаптеры. Использование готовых адаптеров ArrayAdapter и SimpleAdapter для реализации ListView.	1	1	2
87-88	Реляционная модель данных СУБД	Реляционная модель данных и реляционные схемы. Представление данных в виде таблиц. Типы связей: один к одному, один ко многим, многие-ко-многим. Проектирование простейшей БД Необходимость возникновения и история развития СУБД. Обзор и классификация современных СУБД. Проектирование схемы БД.	1	1	2
89-90	Локальные СУБД Введение в SQL	Локальная СУБД на примере SQLite. Введение в SQL. Создание и наполнение таблиц: команды CREATE, INSERT. Команда SELECT для выборки данных, UPDATE для изменения, DELETE для удаления записей из таблицы. Дополнения к запросам SELECT. Ключевые слова ORDER BY, DISTINCT. Агрегация в SELECT в запросах: COUNT, SUM, AVG, MAX, MIN.	2		2
91-92	Знакомство с SQLite	Знакомство с SQLite. Знакомство с SQLite. Разбор заготовки Android приложения. Реализация изученных команд SQL		2	2
93-94	Рекурсия	Линейная и ветвящаяся рекурсия. Стек вызовы. Пример использования при обходе дерева директорий.	1	1	2
95-96	Деревья	Дерево, как базовая структура	1	1	2

		данных. Сбалансированные деревья. Двоичные деревья. Практическое занятие по библиотечному классу TreeSet.			
97-98	Алгоритмы сортировки	Обзор алгоритмов сортировок: пузырьковая, вставкой и быстрая в сравнении трудоемкости. Компараторы. Компаратор, интерфейс Comparable. Практическое занятие по использованию методов класса Arrays, реализующих сортировку.	1	1	2
99-100	Хэш-таблица и функция хэширования	Хэш-таблица и функция хэширования. Поддержка хэширования в Java, метод hashCode. Семейства контейнеров Collections и Map. Практическое занятие по библиотечным классам HashSet и TreeSet.	1	1	2
101-102	Ассоциативные массивы	Ассоциативные массивы. Класс Map, контейнеры HashMap, TreeMap. Хранение данных в SharedPreferences. Практическое занятие по библиотечным классам, реализующим ассоциативные контейнеры.	1	1	2
103-107	Практикум	Решение практических задач по пройденным темам модуля 4. Подготовка к тестированию.		4	4
107 - 108	Контрольное тестирование по модулю 4	Тестовые задания по темам модуля 4.		2	2
<b>Модуль 5. Основы разработки серверной части мобильных приложений</b>			7	<b>19</b>	<b>26</b>
109-110	IP-сети	Адресация в IP-сетях. IPv4. Автоматизация назначения IP-адресов (DHCP). Доменные имена (DNS), URLссылки. Несколько IP адресов для одного сайта. Популярные сетевые команды ping, tracert, ipconfig. Сервисы работы с IP-адресами. Работа с сетевыми командами и сервисами.	1	1	2
111-112	Web сервер	Протокол HTTP. Понятие web-сервера.	1	1	2
113-	HTTP запросы и	Запросы клиента POST и GET,	1	1	2

114	ответы	коды ответов сервера, заголовки запросов и ответов. Разбор запросов и ответов сервера. Отправка запроса на сервер с помощью формы из браузера. Обработка методов на сервере средствами Java, *PHP.			
115-116	Облачные платформы	Облачные платформы. REST взаимодействие Облачные сервисы для хостинга серверной части приложений.	2		2
117-118	REST-взаимодействие	Реализация серверной части средствами Java, *PHP. Стиль взаимодействия REST. Реализация Android приложения REST.		2	2
119-120	Серверные СУБД	Серверные СУБД Хранение данных на сервере в виде БД. Индексы.	2		2
121-122	Серверные СУБД PostgreSQL/MySQL	Знакомство с серверной СУБД на примере PostgreSQL/MySQL. Реализация back-end части web сервера средствами Java, PHP.		2	2
123-124	Практикум	Решение практических задач по пройденным темам модуля 5. Подготовка к тестированию.		2	2
125-128	Практикум	Работа над индивидуальным проектом		4	4
129-130	Контрольное тестирование по модулю 5	Тестовые задания по темам модуля 5.		2	2
131-132	<b>Защита проектов</b>			2	2
<b>Итого:</b>			<b>49</b>	<b>83</b>	<b>132</b>

### **1.3.1. Содержание учебного плана**

#### **Модуль 1. Основы программирования**

##### **1.1. Вводное занятие. Знакомство со средой разработки.**

**Теория:** Введение. Техника безопасности. Порядок создания, компиляции, сборки и запуска в IDE приложения Java. Порядок установки среды разработки на домашнем компьютере.

**Практика:** Разбор Java-проекта.

##### **1.2. Типы данных и операции**

**Теория:** Понятия переменные, примитивные типы данных, арифметические операторы, выражения и присваивания.

**Практика:** Написание простейших программ, объявляющих переменные целого, вещественного типа, присваивающих им значения.

##### **1.3. Операции отношения и логические выражения**

**Теория:** Тип Boolean. Операции отношения, логические и тернарные операции.

**Практика:** Разбор программы, которая демонстрирует выполнение логических операций и операций отношений.

##### **1.4. Условные конструкции**

**Теория:** Понятие блока, область действия блоков. Условные конструкции: if-else, switch.

**Практика:** Закрепление навыков написания всех ранее изученных операторов путем написания и вычисления выражений.

##### **1.5. Итеративные конструкции while**

**Теория:** Общие понятия. Конструкции while, do-while. Безусловные операторы перехода break.

**Практика:** Разбор примеров, иллюстрирующих использование операторов цикла. Написание собственного примера на использование операторов цикла и операторов безусловного перехода.

##### **1.6. Итеративные конструкции for. Массивы**

**Теория:** Вложенные циклы. Безусловные операторы перехода break с меткой, continue. Одномерные массивы, цикл for each в Java.

**Практика:** Разбор примеров нахождения максимума и минимума, поиска на числовых массивах

### **1.7. Методы. Видимость переменных**

**Теория:** Изучение понятия функций на примере методов Java. Передача параметров, возвращение результата. Видимость переменных. Обзорная классификация переменных по области видимости: область класса, область метода, область блока.

**Практика:** Разбор примера функции println. Реализация собственной функции для считывания и вывода массива с использованием уже существующих функций Java.

### **1.8. Многомерные и неровные массивы**

**Теория:** Многомерные массивы на примере двумерных. Матрицы. Особенность массивов в Java.

**Практика:** Разбор примеров с матрицами: объявление, инициализация, обращение к элементам. Использование вложенного цикла for each.

### **Практикум**

Решение практических задач по пройденным темам модуля 1. Подготовка к тестированию.

### **Контрольное тестирование по модулю 1**

Электронный тест охватывает все темы модуля и в большей части направлен на оценку практических знаний и навыков обучающихся, полученных в ходе изучения модуля. Тест представлен на платформе IT школы Samsung. Примерные задания представлены в Приложении 2.

**Модуль 2. Введение в объектно-ориентированное программирование**

#### **2.1. Понятие класса и объекта**

**Теория:** Цели и задачи ОО-подхода к проектированию и разработке ПО. Объект, сообщение, класс, экземпляр объекта, метод. Общее понятие о парадигмах ООП: абстракция, инкапсуляция, наследование, полиморфизм. Описание протокола класса.

**Практика:** Проектирование и реализация простейшего класса. Обзор классов, соответствующих примитивным типам

## **2.2. Конструкторы. Статические методы**

**Теория:** Конструкторы и деструкторы. Разновидности конструкторов. Статические методы. Открытые и закрытые поля. Доступ к полям объекта.

**Практика:** Реализация конструкторов.

## **2.3 Работа с экземплярами класса. Инициализация различных типов данных**

**Теория:** Инициализация переменных, массивов, полей классов в конструкторе.

## **2.4 Перегрузка методов**

**Теория:** Конструктор по умолчанию.

**Практика:** Перегрузка методов на примере конструкторов.

## **2.5 Начальные приемы тестирования и отладки**

**Теория:** Виды тестирования и отладки.

**Практика:** Начальные приемы тестирования и отладки, сценарии тестирования на примерах

## **2.6. Знакомство с Android разработкой**

**Теория:** ОС Android. Среда разработки. Принципиальная архитектура Android-приложения. Изучение жизненного цикла Activity

**Практика:** Создание первого Android приложения в IDE.

## **2.7. Интерфейс Android приложения**

**Теория:** Простейший интерфейс пользователя. Язык разметки XML. Описание ресурсов Android с помощью XML. Разметки (Layouts) и их применение.

**Практика:** Построение простейшего интерфейса пользователя. Создание собственной разметки.

## **2.8. Представления**

**Теория:** Представления (Views).

**Практика:** Создание приложения с простейшим интерфейсом и обработкой событий.

## **2.9. Наследование, инкапсуляция**

**Теория:** Производные классы и наследование. Защищенные части классов и правила доступа для классов и объектов в Java. Сравнение иерархии классов и контейнеризации классов.

**Практика:** Разбор примера, где одно из наследований заменено вложенностью классов.

## **2.10 Понятие контекста. Намерения**

**Теория:** Понятия контекста (Context) и намерения (Intent) в Android Намерения (Intents) в Android.

**Практика:** Разбор примеров и реализация собственного класса, аналог которого общеупотребителен в окружающем мире или в математике и имеет хорошо знакомое поведение.

## **2.11. Полиморфизм**

**Теория:** Полиморфизм. Полиморфные методы и позднее связывание в Java. Абстрактные методы и классы, интерфейсы.

**Практика:** Создание класса, хранящего информацию о человеке (возраст, имя). Разбор примера реализация иерархии классов.

## **Практикум**

Решение практических задач по пройденным темам модуля 2. Подготовка к тестированию.

## **Контрольное тестирование по модулю 2**

Электронный тест охватывает все темы модуля и в большей части направлен на оценку практических знаний и навыков обучающихся,

полученных в ходе изучения модуля. Тест представлен на платформе IT школы Samsung. Примерные задания представлены в Приложении 3.

### **Модуль 3. Основы программирования Android приложений**

#### **3.1 Практикум ООП проектирования**

**Практика:** Разбор кейсов проектирования архитектуры классов приложения.

#### **3.2 Диаграммы UML**

**Теория:** Принципы SOLID. Изучение простейших диаграмм UML.

**Практика:** Построение собственной диаграммы классов.

#### **3.3 Ввод, вывод. Исключения**

**Теория:** Библиотечные классы ввода-вывода. Обработка исключений и классы исключений. Стандартные исключения Java. Работа с файлами в Android.

**Практика:** Работа с классом File, как пример необходимости обработки исключений.

#### **3.4 Внутренние (вложенные) классы**

**Теория:** Понятие внутреннего класса. Отличие от наследования. Назначение. Доступ к состоянию объекта с помощью внутреннего класса.

**Практика:** Разбор примера с доступом к объекту с помощью внутреннего класса.

#### **3.5 Анонимные классы**

**Теория:** Локальные и анонимные внутренние классы. Сущность, синтаксис, назначение. Краткий обзор классов и интерфейсов для обработки событий. Классы Listeners.

**Практика:** Обработка событий пользовательского интерфейса.

#### **3.6 Параллелизм и синхронизация, потоки**

**Теория:** Процессы и потоки в Android. Классы AsyncTask и Thread. Реализация логики потоков. Синхронизация потоков.

**Практика:** Разбор примера использования AsyncTask.

### **3.7 Фрагменты. Сенсоры**

**Теория:** Создание и управление фрагментами. Класс Fragment и его методы. Взаимодействие фрагментов и активностей. Типы сенсоров и обработка событий

**Практика:** Разбор примера приложения с использованием фрагмента.

### **3.8 Двумерная графика в Android-приложениях**

**Теория:** Двумерная графика в Android приложениях. Класс Canvas

**Практика:** Реализация простейшего приложения на Canvas

### **3.9 Разработка игровых приложений**

**Теория:** Этапы проектирования и реализации. Профессии в мире индустрии игр. Знакомство с классом SurfaceView. Обзор. Отличие View от SurfaceView. Особенность класса SurfaceView

### **3.10 Реализация графики на основе SurfaceView**

**Теория:** Понятие игрового движка.

**Практика:** Реализация графики на основе SurfaceView

### **Практикум**

Решение практических задач по пройденным темам модуля 3. Подготовка к тестированию.

### **Контрольное тестирование по модулю 3**

Электронный тест охватывает все темы модуля и в большей части направлен на оценку практических знаний и навыков обучающихся, полученных в ходе изучения модуля. Тест представлен на платформе IT школы Samsung. Примерные задания представлены в Приложении 3.

## **Модуль 4. Алгоритмы и структуры данных**

### **4.1 Массив. Алгоритм двоичного поиска**

**Теория:** Структуры данных. Сложность алгоритмов Массив, как базовая структура данных.

**Практика:** Алгоритм двоичного поиска: идея, применения, реализация.

## **4.2 Классы Arrays и ArrayList**

**Теория:** Классы Arrays (массивы), ArrayList (неограниченный массив).

**Практика:** Изучение класса Arrays: заполнение, копирование, сравнение, печать, методы поиска и др. общие методы. Изучение класса ArrayList. Создание итераторов, навигация.

## **4.3 Список**

**Теория:** Список, как базовая структура данных: стеки, очереди, односвязные и двусвязные списки.

## **4.5 Список LinkedList**

**Теория:** Изучение класса LinkedList, реализующему связные списки

**Практика:** Сравнение ArrayList и LinkedList. Разбор примера применения классов LinkedList, Queue и Stack.

## **4.6 Адаптеры в Андроид**

**Теория:** Назначение и применение. Стандартные адаптеры

**Практика:** Использование готовых адаптеров ArrayAdapter и SimpleAdapter для реализации ListView.

## **4.7 Реляционная модель данных. СУБД**

**Теория:** Реляционная модель данных и реляционные схемы. Представление данных в виде таблиц. Типы связей: один к одному, один ко многим, многие-ко-многим. Проектирование простейшей БД. Необходимость возникновения и история развития СУБД. Обзор и классификация современных СУБД.

**Практика:** Проектирование схемы БД.

## **4.8 Локальные СУБД. Введение в SQL**

**Теория:** Локальная СУБД на примере SQLite. Введение в SQL. Создание и наполнение таблиц: команды CREATE, INSERT. Команда SELECT для выборки данных, UPDATE для изменения, DELETE для удаления записей из таблицы. Дополнения к запросам SELECT. Ключевые слова ORDER BY, DISTINCT. Агрегация в SELECT в запросах: COUNT, SUM, AVG, MAX, MIN.

## **4.9 Знакомство с SQLite**

**Практика:** Знакомство с SQLite. Знакомство с SQLite. Разбор заготовки Android приложения. Реализация изученных команд SQL.

## **4.10 Рекурсия**

**Теория:** Линейная и ветвящаяся рекурсия. Стек вызовов.

**Практика:** Пример использования при обходе дерева директорий.

## **4.11 Деревья**

**Теория:** Дерево, как базовая структура данных. Сбалансированные деревья. Двоичные деревья.

**Практика:** Практическое занятие по библиотечному классу TreeSet.

## **4.12 Алгоритмы сортировки**

**Теория:** Обзор алгоритмов сортировок: пузырьковая, вставкой и быстрая в сравнении трудоемкости. Компараторы

**Практика:** Компаратор, интерфейс Comparable. Практическое занятие по использованию методов класса Arrays, реализующих сортировку.

## **4.13 Хэш-таблица и функция хэширования**

**Теория:** Хэш-таблица и функция хэширования. Поддержка хэширования в Java, метод hashCode. Семейства контейнеров Collections и Map.

**Практика:** Практическое занятие по библиотечным классам HashSet и TreeSet.

## **4.14 Ассоциативные массивы**

**Теория:** Ассоциативные массивы. Класс Map, контейнеры HashMap, TreeMap. Хранение данных в AndroidPreferences.

**Практика:** Практическое занятие по библиотечным классам, реализующим ассоциативные контейнеры.

## **Практикум**

Решение практических задач по пройденным темам модуля 4. Подготовка к тестированию.

## **Контрольное тестирование по модулю 4**

Электронный тест охватывает все темы модуля и в большей части направлен на оценку практических знаний и навыков обучающихся, полученных в ходе изучения модуля. Тест представлен на платформе IT школы Samsung. Примерные задания представлены в Приложении 5.

### **5. Основы разработки серверной части мобильных приложений**

#### **5.1 IP-сети**

**Теория:** Адресация в IP-сетях. IPv4. Автоматизация назначения IP-адресов (DHCP). Доменные имена (DNS), URLссылки. Несколько IP адресов для одного сайта. Популярные сетевые команды ping, tracert, ipconfig. Сервисы работы с IP-адресами.

**Практика:** Работа с сетевыми командами и сервисами

#### **5.2 Web сервер**

**Теория:** Протокол HTTP. Понятие web-сервера.

**Практика:** Понятие web-сервера, разбор на примере одной из наиболее популярных и распространенных программ, созданных для веб-разработчиков.

#### **5.3 HTTP запросы и ответы**

**Теория:** Запросы клиента POST и GET, коды ответов сервера, заголовки запросов и ответов. Разбор запросов и ответов сервера

**Практика:** Отправка запроса на сервер с помощью формы из браузера. Обработка методов на сервере средствами Java, \*PHP.

#### **5.4 Облачные платформы**

**Теория:** Облачные платформы. REST взаимодействие Облачные сервисы для хостинга серверной части приложений.

#### **5.5 REST-взаимодействие**

**Практика:** Реализация серверной части средствами Java, \*PHP. Стиль взаимодействия REST. Реализация Android приложения REST

#### **5.6 Серверные СУБД**

**Теория:** Серверные СУБД Хранение данных на сервере в виде БД. Индексы.

### **5.7 Серверные СУБД PostgreSQL/MySQL**

**Практика:** Знакомство с серверной СУБД на примере PostgreSQL/MySQL. Реализация back-end части web сервера средствами Java, PHP.

#### **Практикум**

Решение практических задач по пройденным темам модуля 5. Подготовка к тестированию.

#### **Контрольное тестирование по модулю 5**

Электронный тест охватывает все темы модуля и в большей части направлен на оценку практических знаний и навыков обучающихся, полученных в ходе изучения модуля. Тест представлен на платформе IT школы Samsung. Примерные задания представлены в Приложении 6.

#### **Защита проектов**

Итоговый контроль реализуется в виде защиты индивидуального/группового проекта обучающимися на тему, выбранную ранее или предложенную педагогом. Примерные темы для создания индивидуального/группового проекта представлен в Приложении 8.

## 1.4. Планируемые результаты

### *Предметные результаты:*

- знание и соблюдение требований техники безопасности и санитарно-гигиенических норм;
- знание основ языка программирования Java;
- умение использовать разные алгоритмы в приемах программирования;
- умение использовать ПК и IDE-разработки для программирования устройства;
- умение читать готовую программу и находить ошибки в готовых программах.

### *Личностные результаты:*

- формирование ответственного отношения к обучению, готовности и способности, обучающихся к саморазвитию и самообразованию средствами информационных технологий;
- формирование универсальных способов мыслительной деятельности (абстрактно-логического мышления, памяти, внимания, творческого воображения, умения производить логические операции);
- развитие опыта участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности;
- формирование целостного мировоззрения, соответствующего современному уровню развития информационных технологий;
- формирование осознанного позитивного отношения к другому человеку, его мнению, результату его деятельности;
- формирование ценности здорового и безопасного образа жизни;
- усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

***Метапредметные результаты:***

- умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
- умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата;
- умение критически оценивать правильность решения учебно-исследовательской задачи;
- умение формулировать, аргументировать и отстаивать свое мнение;
- владение основами самоконтроля, способность к принятию решений;
- умение извлекать нужную информацию из открытых источников;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий (ИКТ-компетенция);
- умение организовывать учебное сотрудничество и совместную деятельность с педагогом и сверстниками в процессе проектной и учебно-исследовательской деятельности.

## 2. Комплекс организационно – педагогических условий реализации общеразвивающей программы

## 2.1. Календарный учебный график на 2020-2021 учебный год

Таблица 2

<b>Условные обозначения:</b>		
	Занятия по расписанию	Каникулярный период
	Итоговая аттестация обучающихся	

## **2.2. Условия реализации программы**

### ***Материально-техническое обеспечение***

#### ***Требования к помещению:***

- помещение для занятий, отвечающие требованиям СанПин для учреждений дополнительного образования;
- качественное освещение;
- столы, стулья по количеству обучающихся и 1 рабочим местом для педагога.

#### ***Оборудование:***

- компьютеры или ноутбуки на каждого обучающегося и преподавателя 13 шт.;
- проекционное оборудование 1 комплект;
- маркерная доска 1 шт.;
- манипулятор типа мышь 13 шт.;
- WEB-камера 1 шт.;
- МФУ (принтер, сканер, копир) 1 шт.;
- Наушники 12 шт.;
- Планшет (для отладки) 13 шт.

#### ***Расходные материалы:***

- whiteboard маркеры;
- бумага писчая;
- шариковые ручки;
- permanent маркеры.

#### ***Информационное обеспечение:***

- Программное обеспечение: пакет MS Office, JDK, IntelliJ IDEA IDE, Android Studio, Notepad++, эмуляторы;
- ПК для педагога, объединенный с функцией сервера;
- Планшет (для отладки) 13 шт.

***Кадровое обеспечение:***

Программа реализуется педагогом дополнительного образования – Волгиной Ириной Александровной.

При реализации программы стоит учитывать, что преподавателю необходимо познакомиться с технологией обучения основам мобильной разработки.

## **2.3. Формы аттестации и оценочные материалы**

Система контроля знаний и умений обучающихся представляется в виде учета результатов по итогам выполнения контрольных тестов по каждому модулю. Метод педагогического наблюдения помогает отслеживать динамику развития обучающегося. В конце освоения дополнительной общеобразовательной общеразвивающей программы обучающиеся проходят защиту индивидуальных/групповых проектов.

Индивидуальный проект оценивается формируемой комиссией по 10-балльной шкале, согласно таблице 3. Защита индивидуального проекта является обязательным этапом завершения данного курса. Для получения Сертификата выпускника. При положительных результатах защиты проекта обучающиеся получают сертификат, подтверждающий успешность прохождения данной программы.

Таблица 3

<b>Баллы, набранные обучающимся</b>	<b>Уровень освоения</b>
0–4 баллов	Незачтено
5–10 баллов	Зачтено

Состав комиссии (не менее 3 человек): в обязательном порядке входит педагог; приветствуется привлечение ИТ - профессионалов, представителей высших и других учебных заведений, администрации учебной организации.

Компонентами оценки индивидуального проекта являются (по мере убывания значимости): качество ИП, отзыв руководителя проекта, уровень презентации и защиты проекта. Если проект выполнен группой учащихся, то при оценивании учитывается не только уровень исполнения проекта в целом, но и личный вклад каждого из авторов. Решение принимается коллегиально. Для оценки индивидуального проекта членам комиссии рекомендуется использовать Бланк оценки индивидуального проекта, который приведен в Приложении 7.

Формы проведения итогов по каждому модулю дополнительной общеобразовательной общеразвивающей программы соответствуют целям и задачам Центра цифрового образования детей «IT-Куб».

## **2.4. Методические материалы**

В образовательном процессе используются следующие методы обучения:

- устные (беседы, объяснение);
- поисковые (изменение программы для приобретения устройством новых свойств);
- демонстрационные (демонстрация возможностей устройства);
- практические (написание программы, проведение минисоревнований).

Программой предусмотрены следующие виды деятельности обучающихся:

- работа с технической и справочной литературой;
- программирование;
- эксперимент, испытание.

Выбор методов обучения осуществляется исходя из анализа уровня готовности обучающихся к освоению содержания модуля, степени сложности материала, типа учебного занятия.

### ***Формы обучения:***

- ***фронтальная*** – предполагает работу педагога сразу со всеми обучающимися в едином темпе и с общими задачами. Для реализации обучения используется компьютер педагога с мультимедиа проектором, посредством которых учебный материал демонстрируется на общий экран. Активно используются Интернет-ресурсы;
- ***групповая*** – предполагает, что занятия проводятся с подгруппой. Для этого группа распределяется на подгруппы не более 6 человек, работа в которых регулируется педагогом;

- **индивидуальная** – подразумевает взаимодействие преподавателя с одним обучающимся. Как правило, данная форма используется в сочетании с фронтальной формой. Часть занятия (объяснение новой темы) проводится фронтально, затем обучающийся выполняет индивидуальные задания или общие задания в индивидуальном темпе;
  - **дистанционная** – взаимодействие педагога и обучающихся между собой на расстоянии, отражающее все присущие учебному процессу компоненты. Для реализации дистанционной формы обучения весь дидактический материал размещается в свободном доступе в сети Интернет, происходит свободное общение педагога и обучающихся в социальных сетях, по электронной почте, посредством видеоконференции или в общем чате. Кроме того, дистанционное обучение позволяет проводить консультации обучающихся при самостоятельной работе дома. Налаженная система сетевого взаимодействия подростка и педагога, позволяет не ограничивать процесс обучения нахождением в учебной аудитории, обеспечить возможность непрерывного обучения в том числе, для часто болеющих детей или всех детей в период сезонных карантинов (например, по гриппу) и температурных ограничениях посещения занятий.

### **Формы организации учебного занятия:**

В образовательном процессе помимо традиционного учебного занятия используются многообразные формы, которые несут учебную нагрузку и могут использоваться как активные способы освоения обучающимися дополнительной общеобразовательной общеразвивающей программы.

В соответствии с возрастом обучающихся, составом группы, содержанием учебного модуля используются следующие *педагогические технологии*:

- беседа;
  - лекция;

- мастер-класс;
- практическое занятие;
- защита проектов;
- лабораторная работа.

Некоторые формы проведения занятий могут объединять несколько учебных групп или весь состав объединения.

Методы воспитания: мотивация, убеждение, поощрение, упражнение, стимулирование, создание ситуации успеха и др.

*Дидактические материалы:*

Методические пособия, разрабатываемые преподавателем с учетом конкретных условий. Техническая библиотека объединения, содержащая справочный материал, учебную и техническую литературу. Индивидуальные задания.

Методическое обеспечение учебного процесса включает разработку преподавателем методических пособий, вариантов демонстрационных программ и справочного материала:

- демонстрационные программы;
- инструкции по настройке среды разработки;
- справочные материалы по терминам ПО;
- учебный материал по теме (портал обучения [myitschool.ru](http://myitschool.ru));
- инструкции по настройке среды разработки.

## Список литературы

### ***Нормативные документы:***

- Федеральный закон «Об образовании в Российской Федерации» от 29.12.2012 N 273-ФЗ;
- Стратегия развития воспитания в Российской Федерации на период до 2025 года. Распоряжение Правительства Российской Федерации от 29 мая 2015 г. № 996-р;
- Письмо Министерства образования и науки РФ от 18.11.2015г. № 09-3242. «О направлении Методических рекомендаций по проектированию дополнительных общеразвивающих программ (включая разноуровневые)»;
- Распоряжение правительства РФ от 04.09. 2014 № 1726-р «Об утверждении Концепции развития дополнительного образования детей»;
- «Основы законодательства РФ об охране здоровья граждан», утвержденные Верховным советом РФ от 22.07.1993 № 5487 - (ред. от 25.11.2009);
- Федеральный закон от 24.07.1998 № 124-ФЗ «Об основных гарантиях прав ребенка в РФ»;
- Федеральный закон «Об основах охраны здоровья граждан в Российской Федерации», 2011г.;
- Приказ Министерства просвещения России от 09.11.2018 г. № 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам» (Приказ №1008 отменен);
- Санитарно-эпидемиологические правила и нормативы СанПиН 2.4.4.3172-14.

## **Перечень учебной литературы, необходимой для освоения курса**

### ***Основные источники***

1. Н. М. Эдвардс, С. И. Осипова. Формирование компетентности ученого для международной научной проектной деятельности. – Инфра-М, Сибирский федеральный университет СФУ, 2018. – 240 с.
2. Янушевский В.Н. Учебное и социальное проектирование в основной и старшей школе: Методическое пособие / В.Н. Янушевский. – М.: Сентябрь, 2017 – 224 с.
3. Косярский Александр Алексеевич, Дорошкевич Татьяна Ивановна, Даниш Виктория Геннадьевна. Организация проектной деятельности. – БУК, 2020. – 64 с.
4. Алексей Корнилов. Методология проектной деятельности в образовании. – Издательские решения, 2019. – 128 с.
5. Фаулер Мартин. UML. Основы. Краткое руководство по стандартному языку объектного моделирования. – Символ-Плюс, 2018. – 192 с.

### ***Дополнительные источники***

1. М. Нейгард. Java. Release it! Проектирование и дизайн ПО для тех, кому не всё равно. — Питер, 2016 — 320 с.
2. Вольфсон Борис Ильич. Гибкое управление проектами и продуктами. – Питер, 2017. – 144 с.
3. Луис Розенфельд и Питер Морвиль. Информационная архитектура в Интернете, 2 е издание. – Символ Плюс, 2019. – 544 с.

*Перечень ресурсов информационно-телекоммуникационной сети  
«Интернет»*

1. Портал обучения: <https://myitschool.ru>
2. Документация для разработчиков приложений: <https://developer.android.com>
3. Портал Хабрахабр: <https://habrahabr.ru>
4. Науменко О.М. Творчествоведение на современном этапе. Академия творческоведческих наук и учений: <http://atnu.narod.ru/tvorit.html>
5. Портал обучения Университета ИТМО: <https://de.ifmo.ru>

**Примерные задания предварительного тестирования дополнительной  
общеобразовательной общеразвивающей программы «Мобильная  
разработка. ИТ школа Samsung»**

1. Какое из приведенных чисел удовлетворяет логическому условию: (нечетное И делится без остатка на 3) ИЛИ простое? Если таких чисел несколько, укажите меньшее

Выберите один ответ:

- a. 45
- b. 20
- c. 29
- d. 33

2. Вычислите в той же системе счисления (указывать основание счисления в ответе не нужно)

$$121_3 + 12_3 =$$

3. Максимальное пятиразрядное число в двоичной системе счисления (указывать основание счисления в ответе не нужно)

4. В ящике 40 фруктов желтого и зеленого цвета, среди которых 22 яблока, остальные - груши. Известно, что 16 фруктов желтого цвета, а 14 груш зеленого. Сколько желтых яблок?

5. Определите значение переменной с после выполнения следующего фрагмента программы (все переменные целого типа):

Паскаль	Си	Python
a := 18;	a = 18;	a = 18
b := 2;	b = 2;	b = 2
a := a * b;	a = a * b;	a = a * b
b := a mod 6;	b = a % 6;	b = a % 6
c := a - b;	c = a - b;	c = a - b

6. Переменные  $a, b, c$  - длины сторон треугольника. Фрагмент программы, которая вычисляет периметр треугольника при условии возможности его существования (длина каждой стороны меньше суммы остальных) - это...

Выберите один ответ:

а) Ни один из вариантов

б)

Паскаль	Си
<pre>if (a &gt; b + c) and (b &gt; a + c) and (c &gt; a + b) then   writeln(a + b + c);</pre>	<pre>if ((a &gt; b + c) &amp;&amp; (b &gt; a + c) &amp;&amp; (c &gt; a + b)) {   printf("%d", a + b + c); }</pre>

в)

Паскаль	Си
<pre>if (a &lt; b + c) and (b &lt; a + c) and (c &lt; a + b) then   writeln(a + b + c);</pre>	<pre>if ((a &lt; b + c) &amp;&amp; (b &lt; a + c) &amp;&amp; (c &lt; a + b)) {   printf("%d", a + b + c); }</pre>

г)

Паскаль	Си
<pre>if (a &lt; b + c) or (b &lt; a + c) or (c &lt; a + b) then   writeln(a + b + c);</pre>	<pre>if ((a &lt; b + c)    (b &lt; a + c)    (c &lt; a + b)) {   printf("%d", a + b + c); }</pre>

д)

Паскаль	Си
<pre>if (a &gt; b + c) or (b &gt; a + c) or (c &gt; a + b) then   writeln (a + b + c);</pre>	<pre>if ((a &gt; b + c)    (b &gt; a + c)    (c &gt; a + b)) {   printf("%d", a + b + c); }</pre>

7. Приведенный фрагмент программы вычисляет минимум в заданном массиве

Паскаль	Си	Python
//A - целочисленный массив с индексами от 0 до 5 rez := A[0]; for i := 1 to 5 do if ____ then rez := A[i];	//A - целочисленный массив с индексами от 0 до 5 rez = A[0]; for (i = 1; i <= 5;i++) if ( ____ ) rez = A[i];	#A - целочисленный массив с индексами от 0 до 5 rez = A[0] for i in range(1, len(A)): if ____ : rez = A[i]

Ведите пропущенное условие в наиболее короткой форме без пробелов.

**Примерные задания контрольного тестирования по модулю 1  
дополнительной общеобразовательной общеразвивающей программы  
«Мобильная разработка. ИТ школа Samsung»»**

**Модуль 1. Основы программирования**

1. Типы данных Java. Фрагменты, которые не вызовут сообщение компилятора об ошибке преобразования типа:

- int x = 21; double y = x;
- short z = 13; int x = z;
- long x = 15; int y = x;
- int x = 14; short z = x.

2. Операции и выражения в Java. Выражения, результат которых имеет тип float или double:

- 5.0 \* 3
- 5 / 2
- 5 \* 0.5
- 2.5 \* 2

3. Условные операторы в Java. Аналогом фрагмента кода if (a == 2 || b == 3) {f ();} является:

- if (a == 2) {f ();} else if (b == 3) {f ();}
- if (a == 2 && b == 3) {f ();}
- if (a! = 2 && b! = 3) {f ();}
- if (a == 2) if (b == 3) {f ();}
- if (a! = 2) {} else if (b! = 3) {f ();}

4. Циклы в Java. Значение переменной a после окончания выполнения цикла равно:

```
int a = 10;  
for (int i = 1; i < 3; ++i) a *= 4;
```

5. Массивы в Java. Правильное обращение к среднему элементу массива a нечетной длины приведено в:

- a[a.length / 2]
- a[a.length / 2 + 1]
- a[a.length % 2]
- a.middle
- a[a.length/2-0.5]

6. Видимость переменных. Фрагмент кода, который вызовет ошибку компиляции:

- static int a = 7;  
static int f() { int a = 3, b = 4; return a + b; }
- static int a = 7;  
static int f() { int b = 4; return a + b; }
- static int a = 7;  
static int f(int a) { int b = 4; return a + b; }
- static int a = 7;  
static int f(int a, int b) { int b = 4; return a + b; }
- static int f(int a) { int b = 4; return a + b; }

7. Передача параметров в методы Java. В результате работы приведенного ниже фрагмента кода будет выведено:

```
static void swap(int a, int b) {  
    int tmp = a; b = a; a = tmp; }  
  
public static void main(String[] args) {  
    int a = 5; int b = 10; swap(a, b); System.out.print(a + ", " + b); }
```

8. Многомерные массивы. Пропущенный фрагмент функции, подсчитывающей сумму элементов, находящихся под побочной диагональю квадратной матрицы  $n*n$  (побочной диагональю матрицы называют линию, соединяющую левый нижний и правый верхний угол матрицы) это:

```
int SumUnderSecondaryDiagonal(int n, int[][] matrix) {  
    int sum = 0;  
    if (n<2) return 0;  
  
    ....  
  
    return sum; }
```

- for(int i=0;i<n;i++)  
for(int j=n-i-1;j<n;j++) sum=sum+matrix[i][j];
- for(int i=n-1;i>0;i--)  
for(int j=i;j>0;j--) sum+=matrix[i][j];
- for(int i=0;i<n;i++)  
for(int j=0;j<i;j++) sum=sum+matrix[i][j];
- for(int i=n-1;i>0;i--)  
for(int j=n-i;j<n;j++) sum+=matrix[i][j];

**Примерные задания контрольного тестирования по модулю 2  
дополнительной общеобразовательной общеразвивающей программы  
«Мобильная разработка. ИТ школа Samsung»»**

**Модуль 2. Введение в объектно-ориентированное программирование**

1. Передача параметров строк. В результате работы приведенного ниже фрагмента кода будет выведено:

```
public class Example{  
    static String str = "ABC";  
    public static void changeStr(String s) {  
        s = "abc"; }  
    public static void main(String[] args) {  
        System.out.print(str);  
        changeStr(str);  
        System.out.print(str); }  
}
```

2. Про приемы тестирования и отладки. Виды тестирования, которые могут заменить отладчик с применением точек останова с условием:

- Модульное тестирование;
- Стресстестирование;
- Логирование;
- Проверка утверждения (assert);
- Интеграционное тестирование.

3. Язык разметки XML. Верные утверждения:

- XML-документ всегда должен содержать ровно один корневой элемент;
- В XML-документе не могут присутствовать цифры;
- В XML-документе могут присутствовать только латинские символы;
- XML-документ может содержать произвольное количество корневых элементов;
- Ни один из вариантов.

4. Layout. Свойство, позволяющее назначить элементу важность его размера относительно других элементов в контейнере:

- layout\_weight
- layout\_width
- layout\_height
- layout\_gravity
- layout\_wrap

5. Основные понятия объектно-ориентированного подхода в программировании (теория). Создание нового класса на основе уже существующего называют:

- наследование;
- полиморфизм;
- декомпозиция программы;
- инкапсуляция.

6. Классы. Правильное имя публичного класса в файле “Point.java”

- Point
- Point.java
- point.java
- PointClass
- должно начинаться со слова Point

7. Перегрузка методов. Инициализация данных класса. Ошибка, к которой приводит отсутствие инициализации локальной переменной:

- происходит ошибка компиляции;
- переменная инициализируется случайными данными;
- переменная инициализируется null;
- происходит инициализация исключения.

8. Программы на конструкторы и перегрузку.

```
class Point {  
    float x, y;  
    // Конструктор  
}  
public class Example {  
    public static void main(String[] args) {  
        Point x = new Point(4.0f, 5.0f);  
        System.out.print(x.x + x.y);  }  
}
```

9. Метод класса MyDate whatIs() возвращает ложь, если:

- дата в параметре метода позже либо равна дате объекта, вызвавшего метод;

- дата в параметре метода строго позже, чем дата объекта, вызвавшего метод;
  - дата в параметре метода строго раньше, чем дата объекта, вызвавшего метод;
  - дата в параметре метода раньше либо равна дате объекта, вызвавшего метод;
  - ни один из вариантов.
10. Наследование. Количество различных методов (из определенных в приведенном коде), которое может быть вызвано в методе main у переменной x без преобразования ее типа равно:

```
class Base {  
    public void f1() {}  
    private void f2() {}  
    private void f3() {}  
}  
  
class Subclass extends Base {  
    public void f1() {}  
    public void f4() {}  
}  
  
class EntryPoint {  
    public static void main(String[] argv) {  
        Subclass x = new Subclass();  
    }  
}
```

11. Наследование. Правильный вызов конструктора базового класса Base из конструктора его наследника приведен в:

- super(args);
- new Base(args);
- Base(args);
- new super(args);
- ни один из вариантов.

12. Полиморфизм. Методы переменной x, которые могут быть вызваны без преобразования типа переменной x:

```
class Base {  
    public int f() { ... }  
    public int g() { ... }  
}  
  
class Derived extends Base {  
    public int g() { ... }  
}
```

- ```
public int h() { ... }  
}  
Base x = new Derived();  
• f;  
• g;  
• h;  
• ни один из вариантов.
```

**Примерные задания контрольного тестирования по модулю 3  
дополнительной общеобразовательной общеразвивающей программы  
«Мобильная разработка. ИТ школа Samsung»»**

**Модуль 3. Основы программирования Android приложений**

1. ОО проектирование. Какой из вариантов лучше описывает отношение "Человек (man) проживает (live) по адресу (address)"?
  - class Man extends Address{ };
  - class Man { private Live address; };
  - class Man implements Address{ };
  - class Man { private Address live; };
  - class Address{ private Man live; }.
2. Activity. Последовательность запуска методов жизненного цикла Activity, начиная с onCreate():
3. Создание Андроид проекта. Основным классом Android приложения, имеющего интерфейс пользователя, является:
  - Activity;
  - KeyEvent;
  - View;
  - Fragment;
  - Canvas.
4. Ввод, вывод и исключения. Ключевое слово, показывающее, что в процессе работы метода могут произойти не обработанные исключения:
  - try;
  - catch;
  - finally;
  - throws.
5. Класс(ы), используемый(е) для чтения файла.
  - FileInputStream;
  - FileOutputStream;
  - FileInputStream, FileOutputStream;
  - ни один из перечисленных.
6. Внутренние и анонимные классы. Пропущенный фрагмент, после вставки которого программа выведет на экран значение 50 – это:

```
public class OuterClass {
```

```

private int x = 0;
private class InnerClass {
    InnerClass() {
        OuterClass.this.x = 50;
    }
}
private InnerClass xy;
public OuterClass() {____}
public static void main(String[] args) {
    OuterClass obj = new OuterClass();
    System.out.println(obj.x);
}
}

```

- xy = new InnerClass();
- x = 50;
- this.xy = new InnerClass();
- xy = InnerClass();
- xy = 50.

7. Слушатели. Необходимо, чтобы когда пользователь отпускал кнопку myButton (класс android.widget.Button), запускался метод doIt(). Возможные способы реализации...

- Назначить в XML layout атрибуту android:onClick значение “doIt”;
- Назначить кнопке слушателя onClickListener и вызвать doIt() из метода onClick();
- Назначить кнопке слушателя OnTouchListener и вызвать doIt() из метода onTouch();
- Вызвать startActivity с параметром myButton из myButton.onClick();
- ни один из вариантов.

8. Параллелизм и синхронизация. Потоки. Операции, которые не должны выполняться одновременно в разных потоках:

- Два чтения одной и той же переменной;
- Запись и чтение одной и той же переменной;
- Две записи в одну и ту же переменную;
- Изменение двух разных экземпляров одного и того же класса.

9. Context и Intent. Верные утверждения:

- Класс android.context.Context используется для доступа к глобальной информации об окружении Android приложения;
- Для передачи намерения в Activity можно использовать метод startActivityForResult();
- Метод putExtra() предназначен для сохранения данных в Intent;

- Явным (explicit) называется Intent, который содержит флаг Intent.FLAG\_EXPLICIT;
- Для передачи намерения в Activity можно использовать любой метод.

## 10. Анализ программ с Canvas.

```
paint.setStyle(Paint.Style.FILL);
paint.setColor(Color.GREEN);
canvas.drawRect(50, 80, 400, 280, paint);
paint.setColor(Color.YELLOW);
canvas.drawCircle(150, 150, 50, paint);
```

Приведенный фрагмент рисует:

- желтый круг в верхнем левом углу зеленого прямоугольника;
- желтый круг в верхнем правом углу зеленого прямоугольника;
- желтый круг в нижнем правом углу зеленого прямоугольника;
- зеленый круг в верхнем левом углу желтого прямоугольника;
- ни один из вариантов.

**Примерные задания контрольного тестирования по модулю 4  
дополнительной общеобразовательной общеразвивающей программы  
«Мобильная разработка. ИТ школа Samsung»»**

**Модуль 4. Алгоритмы и структуры данных**

1. Массивы: Arrays, ArrayList. Имеется класс:

```
class A { public int[] arr; }
```

В поле arr переменной a после выполнения операции: a = new A(); будет хранится:

- null;
- массив из 0 элементов;
- не определено;
- массив случайной длины из случайных чисел.

2. Списки (общие вопросы, теория). В двусвязном списке необходимо поменять два соседних элемента местами. Количество ссылок, которое будет изменено (если ни один из элементов не является ни первым ни последним) –  
3. Списки: LinkedList, Queue, Stack. Операция взятия с удалением элемента из Queue:

- push;
- pop;
- poll;
- peek.

4. Деревья и двоичный поиск (общие вопросы, теория). Двоичное сбалансированное дерево обладает следующим свойством:

- Высота левого поддерева отличается от высоты правого поддерева не более, чем на 1;
- Все значения в вершинах левого поддерева не больше значений в вершинах правого поддерева;
- Все значения в вершинах левого поддерева больше значения в корне поддерева;
- Высота левого поддерева не меньше высоты правого поддерева.

5. Деревья: TreeSet. В результате работы ниже приведенного фрагмента на экран будет выведено:

```
TreeSet<Integer> tree = new TreeSet<Integer>();
```

```

tree.add(15);
tree.add(25);
tree.add(15);
tree.add(11);
Iterator<Integer> iterator = tree.iterator();
while (iterator.hasNext()) System.out.print(iterator.next() + " ");

```

6. Рекурсия. Для данной функции количество рекурсивных вызовов (т. е. вызовов функции из самой себя) при  $n=5$  равно...

```

int fib(int n) {
    if (n < 3) { return 1; }
    return fib(n - 1) + fib(n - 2);
}

```

7. Алгоритм двоичного поиска (класс Arrays и Collections).  
`Arrays.binarySearch(new int[]{-1, 3, 5, 5, 19}, -1)` возвратит:

- 0;
- 1;
- -1;
- true;
- false.

8. Алгоритмы сортировки (класс Arrays). При вызове метода `Arrays.sort(a)`; переменная `a` может быть следующего типа:

- `int[]`;
- `String[]`;
- `ArrayList<Integer>`;
- `String`;
- `TreeMap<Integer, Integer>`.

9. Хэш-таблица и функция хэширования (практика). Заданная программа выведет на экран число:

```

class Point {
    int x, y;
    public Point(int x, int y) { this.x = x; this.y = y; }
    @Override
    public int hashCode() {
        final int prime = 31;
        return prime + prime * x + prime * prime * y;
    }
}
public class Test {
    public static void main(String[] args) {

```

```

        HashSet<Point> set = new HashSet<Point>();
        set.add(new Point(12, 13));
        set.add(new Point(9, 0));
        set.add(new Point(12, 13));
        set.add(new Point(9, 30));
        System.out.println(set.size()); }
    }

```

10. Ассоциативные массивы: `HashMap`, `TreeMap`. Порядок, в котором появляются элементы при итерации по классу `HashMap`:

- зависит от реализации библиотеки;
- по неубыванию ключей;
- по невозрастанию ключей;
- по неубыванию значений;
- по невозрастанию значений

11. Анализ программ.

```

class WhoAmI {
    public int whatAmIDoing(int x) {
        set.add(x); return set.headSet(x).size(); }
    public static long howDoIPProcessThisArray(int[] a) {
        WhoAmI obj = new WhoAmI();
        long res = 0;
        for (int i = a.length - 1; i >= 0; --i) {
            res += obj.whatAmIDoing(a[i]); }
        return res; }
    private SortedSet<Integer> set = new TreeSet<Integer>();
}

```

Вызов `WhoAmI.howDoIPProcessThisArray(new int[]{2, 4, 7, 2, 5, 3, 1})` возвратит?

12. Реляционная модель, СУБД – понятия. Для отдельно взятой таблицы в реляционной модели баз данных верно утверждение:

- Каждая строка может иметь собственное количество столбцов;
- В одном столбце в разных строках могут быть записаны элементы разного типа;
- Все строки имеют одинаковое количество столбцов и в каждом столбце хранятся данные одного типа;
- Каждая строка содержит произвольное количество однотипных данных.

13. CREATE 2. Выполнен следующий запрос:

```
CREATE TABLE pupil (
```

name varchar (30), surname varchar (30),  
age integer, birthday date,  
height float, weight float).

Количество колонок, которые будут иметь строковой тип в созданной таблице -?

14. Запросы CREATE. Выберите таблицу, содержание которой соответствует запросу: CREATE TABLE pupil (name varchar(3), age integer)?

1)

| name | age |
|------|-----|
| Ivan | 13  |
| Wu   | 10  |

2)

| name | age |
|------|-----|
| Hu   | 13  |
| Wu   | 10  |

3)

| name | age  |
|------|------|
| Ha   | 13.5 |
| Wu   | 10   |

4)

| name | age    |
|------|--------|
| Ivan | twelwe |
| Wu   | eleven |

**Примерные задания контрольного тестирования по модулю 5  
дополнительной общеобразовательной общеразвивающей программы  
«Мобильная разработка. ИТ школа Samsung»»**

**Модуль 5. Основы разработки серверной части мобильных приложений**

1. Протоколы, IP-сети. У каждого компьютера в сети всегда есть:
  - MAC адрес;
  - IP адрес;
  - URL адрес;
  - Bluetooth;
  - WiFi.
2. Маски. Количество IP адресов, которое можно закодировать в сети с маской подсети 255.255.254.0:
  - 29;
  - 28;
  - 27;
  - 2254;
  - ни один из вариантов.
3. Утилиты. Утилита, которая позволяет проверить информацию о всех установленных на компьютере сетевых адаптерах:
  - ipconfig/ifconfig
  - ping
  - tracert/traceroute
  - nslookup
  - netstat
4. Retrofit. В приведенном фрагменте с использованием Retrofit 2.x

```
public interface APIService {  
    @POST("user/list");  
    Call<Users> loadUsers();}
```

- декларируется POST запрос, синхронный или асинхронный будет определено позже;
- декларируется асинхронный POST запрос;
- декларируется синхронный POST запрос;
- декларируется синхронный GET запрос;

- ни один из вариантов.
5. JSON. Количество JSON объектов и JSON массивов в файле:

```
{  
  "contacts": [  
    {"id": 5, "name": "Ivan", "gender": "male", "phone": {  
      "mobile": "+7(915)365-2500",  
      "home": "+7(495)123-4567",  
      "office": "+7(495)797-2500" } }]  
}
```

- 3 json объекта, 1 json массив;
  - 8 json объектов, 1 json массив;
  - 3 json объекта, 0 json массивов;
  - 8 json объектов, 1 json массив;
  - ни один из вариантов
6. Android. Верные утверждения о HTTP запросах в Android приложениях:
- Для отправки данных формы заказа товара следует использовать HttpPost запрос;
  - Обязательным компонентом любого запроса является строка запроса (Start-Line);
  - Для получения данных с сервера следует использовать только HttpGet запрос;
  - Для отправки данных авторизации можно использовать HttpGet запрос;
  - ни один из вариантов.

## Приложение 7

Проект является одним из видов самостоятельной работы, предусмотренной в ходе обучения по программе. Педагог оказывает консультационную помощь в выполнении проекта.

В комплект обязательных материалов, которые предоставляются обучающимся, входит: исходный код в архиве, презентация проекта.

## Ведомость

## на защиту индивидуальных проектов по программе «IT школы Samsung»

Площадка \_\_\_\_\_

Группа \_\_\_\_\_

### Члены комиссии:

(ФИО, место работы, должность)

(подпись)

(ФИО, место работы, должность)

(подпись)

(ФИО, место работы, должность)

(подпись)

(ФИО, место работы, должность)

## Примечания:

**Примерные темы итогового контроля реализации дополнительной  
общеобразовательной общеразвивающей программы  
«Мобильная разработка. ИТ школа Samsung»»**

**Тема 1. Поле чудес**

Создать аналог игры «Поле чудес». Реализовать различную функциональность для ролей “Администратор игры” и “Игрок”.

Администратор игры может задавать банк слов для отгадывания и их описание. В процессе игры из этого банка случайным образом выбираются слова.

**Тема 2. Scrabble**

Создать аналог игры «Scrabble». Игра должна быть обучаемая (т.е. должен быть реализован интерфейс добавления новых слов) и должна быть возможность игры на одном устройстве как с компьютером, так и с другим игроком.

**Тема 3. Угадай мелодию. New Version**

Создать гибрид игр «Угадай мелодию» и «Кто хочет стать миллионером», для двух игроков, в котором реализовано:

- Торг за прослушивание песни (1-Н секунд).
- Воспроизведение песни.
- Выбор из 4-х вариантов правильного.
- При выборе неправильного варианта вычтываются балы.
- Условие победы: первым набрать определенное количество балов.

**Тема 4. Текстовый квест**

Создать игру жанра «текстовый квест», в которой реализованы следующие компоненты, описанные ниже.

Игровое поле с отображением количества здоровья, силы, основных навыков персонажа (таких как урон, магия, защита, дополнительные

артефакты и их свойства), а также поле с выводом текста квеста и поле выбора действия (radiogroup).

В поле вывода текста описывается ситуация, в которую попал персонаж, в поле выбора действия предлагается выбор путей развития ситуации.

Квест считается выполненным, если достигнута цель, поставленная в начале (например, добыть диск из пещеры тролля, перевезти артефакты через реку) \*можно ориентироваться на текстовые квесты в игре «Космические рейнджеры».

Например, цель и сюжет квеста:

Текст выдачи задания: Наш президент – страстный коллекционер, и он желает заполучить себе в коллекцию знаменитый бриллиант Софиты. Единственный способ — это сделать - ограбить малокский банк на планете <ToPlanet> звездной системы <ToStar>. Конечно, это незаконно, но ведь вас, <Ranger>, судя по вашей репутации, это не должно беспокоить, ведь так? Вся команда уже в сборе, не хватает только взломщика сейфов. На планету надо прибыть не позднее <Date>, иначе грабители найдут другого взломщика. И еще - в случае успеха мы выплатим вам <Money> cr, плюс вам никто не мешает почистить деньгиохранилища банка.

Сюжет:Квест представляет собой историю об ограблении банка, с целью получения знаменитого бриллианта Софиты. Помимо самого рейнджера, которому отведена роль "медвежатника" на дело также идут: Рикшар - малок, пират старый знакомый главного героя; Крамер - фэянин, инженер; Зелиф - пеленг, подрывник и Йомир - малок, специалист по оружию. Квест можно разделить на две части: в первой происходит подготовка к ограблению, рейнджер в это время может тренироваться во взломе РГ-сейфа, изучать банк под видом вкладчика и улучшать свои отношения с напарниками. Во второй части квеста происходит собственно проникновение в банк с масштабным штурмом и большим количеством взрывов, в лучших традициях боевиков про ограбления.

### Тема 5. Морской бой. Remake

Создать аналог игры «Морской бой», где вторым игроком будет выступать компьютер. Особенности: вместо четырех однопалубных кораблей, должно выставляться два однопалубных корабля и две мины. Попадая на мины, противник пропускает дополнительно ход.

### Тема 6. Эмулятор космических объектов

Создать эмулятор космических объектов 2D, который должен содержать следующие компоненты, описанные ниже.

Поля ввода, где пользователь задает начальное положение планет в космосе, их массу и скорость.

Кнопку «симуляция», при нажатии на которую система показывает, как планеты летают вокруг солнца. - Для расчета траектории взять формулы из физики.

### Тема 7. Робот

Создать программу программирования робота, содержащую компоненты, описанные ниже.

Игровое поле NxM клеток, на котором в исходной клетке находится робот, стены или предметы, через которые робот пройти не может, и квадрат куда роботу необходимо дойти.

Задание программы действий работу в виде последовательности команд кнопками:  $\leftarrow$   $\uparrow$   $\rightarrow$   $\downarrow$ .

Поле вывода введенных команд.

Кнопка “Идти”, по которой робот выполняет введенную последовательность команд.

Условие победы: робот должен дойти до заданного квадрата.

Должно быть реализовано поэтапное увеличение сложности игрового поля и уменьшение отведённого на ввод команд времени.

### Тема 8. Приложение заметки

Реализовать простой менеджер заметок на основе базы данных, который должен включать в себя следующие функции:

- создание заметки,
- редактирование заметки,
- удаление заметки
- возможность отметить заметку как выполненную,
- сортировка заметок на основе даты ее создания.

## Тема 9. Тамагочи

Создать эмулятор существа, содержащий:

- Изображение существа.
- Поля состояния существа (голод, усталость, скука, счастье и т.д.).
- Кнопки действий: накормить, спать, играть и. т.д.
- Состояние постепенно должно меняться (голод нарастать, счастье уменьшаться и т.д.).
- При достижении пограничных состояний изображение существа меняется (с счастливого на грустное и т.д. и наоборот).
- При достижении граничных состояний существо умирает.
- Со временем скорость игры увеличивается.
- Задача игрока - чтобы существо прожило как можно дольше.

Сохранение лучшего времени.

## Тема 10. Изучаем английский

Создать программу обучения английскому, в которой реализованы следующие компоненты, описанные ниже.

Различные интерфейсы для заданий.

Задания, на которые надо ответить «да» или «нет» (например, пары слов: long-длинный; power-простой).

Задания на правописание (например, вставить недостающие буквы в слове).

Задания на сопоставление пар слов или картинок (например, rabbit – хорек, кролик, удав).

Выводить правильные варианты ответов после каждого курса заданий, с указанием, где ошибся пользователь.

Если правильных ответов меньше N% - повторить курс

### Тема 11. База данных студентов

Реализовать систему ввода и отображения информации о студентах института, включающую следующие сущности и их атрибуты:

- Студент (Имя, Фамилия, Отчество, Дата рождения).
- Группа (Номер, Название факультета).
- Система должна иметь следующие функции:
  - Отображение списка групп.
  - Добавление новой группы, редактирование и удаление существующей.
  - Отображение списка студентов.
  - Фильтрация списка студентов по фамилии и по номеру группы.
  - Добавление нового студента, редактирование и удаление существующего.
  - Система должна иметь защиту от удаления группы, содержащей студентов.

### Тема 12. Бронирование билетов

Создать систему для бронирования билетов в кино\театр\футбол\хоккей, в которой реализованы роль пользователя и администратора. Пользователю доступно:

- Создание учетной записи пользователя.
- Выбор сферы, для бронирования билетов.
- Фильтр по жанрам для кино и театров.
- Выбор даты.
- Отображение названия мероприятия + картинка + возможное время посещения мероприятия (предварительно занесенных в SQLite).
- Подробное описание мероприятия.