

Департамент образования администрации Города Томска  
Муниципальное автономное образовательное учреждение дополнительного образования  
Центр творческого развития и гуманитарного образования  
«Томский Хобби – центр»  
Структурное подразделение ЦЦОД «IT-куб.Томск»

Принята на заседании  
Педагогического совета  
От «28» 08. 2023 г.  
Протокол № 1



Утверждаю:

Директор Томского Хобби-центра

Л.В.Дубровина

20 23 г.

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ  
ПРОГРАММА ТЕХНИЧЕСКОЙ НАПРАВЛЕННОСТИ**

**«Мобильная разработка»**

Возраст учащихся: 14-17 лет

Срок реализации 1 год

Автор составитель:

Филатова Анастасия Владимировна,  
методист

г. Томск, 2023.

## ХАРАКТЕРИСТИКА ПРОГРАММЫ

**Название программы:** Дополнительная общеобразовательная общеразвивающая программа на внебюджетной основе «Мобильная разработка»

**Направленность:** техническая

**Возраст учащихся:** 14-17 лет

**Срок обучения:** 1 год

**Особенности состава учащихся:** постоянный

**Форма обучения:** очная

**По уровню содержания:** базовая

**По срокам реализации:** долгосрочная

### Пояснительная записка

Программа технической направленности «Мобильная разработка» направлена на привлечение учащихся к современным технологиям программирования и практического применения языка при разработке мобильных приложений. Долгосрочная со сроком обучения один год. По уровню содержания программа является базовой.

В ее основе лежат нормативные документы:

- Конституция РФ;
- Конвенция ООН о правах ребенка;
- Федеральный закон от 29 декабря 2012 года №273-ФЗ (ред. от 07.10.2022) «Об образовании в Российской Федерации»;
- Федеральный проект «Успех каждого ребенка» в рамках Национального проекта «Образование». Министерство Просвещения Российской Федерации 2019 г.;
- Федеральный проект «Цифровая образовательная среда» в рамках национального проекта «Образование» государственной программы «Развитие образования». Министерство Просвещения Российской Федерации 2018 г.;
- Письмо Министерства просвещения Российской Федерации «О корректировке методических рекомендаций по созданию и функционированию центров цифрового образования «IT-куб» от 07.09.2022г. № АЗ 1346/04.
- Письмо Минобрнауки России от 18.11.15 года № 09-3242 «О направлении информации» (вместе с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые)»;

- Концепция развития дополнительного образования детей до 2030 года (Распоряжение Правительства РФ от 31.03. 2022 г. № 678-р)
- Письмо Минобрнауки России от 18.11.15 года № 09-3242 «Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые)»;
- Приказ Министерства Просвещения России от 27.07.2022 № 629 "Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам"
- Постановление Главного государственного санитарного врача Российской Федерации 28.09.2020 № 28 г. Москва «Об утверждении СанПиН 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;
- Устав МАОУ «Томский Хобби-центр».

### **Пояснительная записка**

Программирование входит в число ключевых навыков XXI века, и, в скором времени, базовое понимание кода станет обязательным для успешного карьерного роста. Программирование как одно из самых популярных направлений в дополнительном образовании для учащихся школьного возраста появилось в России несколько лет назад и продолжает успешно развиваться.

Использование современных информационных технологий является необходимым условием успешного развития общества и соответственно государства в целом. В связи с этим актуальной задачей является подготовка специалистов сферы информационных технологий в соответствии с профессиональными требованиями динамично развивающихся отраслей. Программа служит средством внутрипрофильной специализации в области новых информационных технологий, что способствует созданию дополнительных условий для проявления индивидуальных образовательных интересов учащихся, их дальнейшей профессиональной ориентации.

#### **Направленность программы**

Программа имеет техническую направленность, т.к. содержание программы рассматривается как средство формирования образовательного потенциала, позволяющего развивать наиболее передовые на сегодняшний день технологии — информационные, интегрирующие в себе науку, технологию, инженерное дело.

**Педагогическая целесообразность** данной программы заключается в развитии

основных познавательных процессов, умения анализировать, выявлять сущности и отношения, описывать планы действий и делать логические выводы, опираясь на такие дисциплины, как механика, теория управления, программирование, теория информации. Кроме того создает благоприятные условия для социально-культурного и профессионального самоопределения, развития познавательной активности и творческой самореализации учащихся. Обеспечивает пропедевтику профессионального изучения программирования в высшей школе.

### **Актуальность программы**

Актуальность данной программы состоит в том, что она составлена с учётом современных потребностей рынка в специалистах в области мобильной разработки и программирования на Java.

Чтобы двигаться в ногу со временем и иметь фору перед другими специалистами, требуется постоянно развиваться как специалисту, приобретать новые компетенции, и не только специальные (Hard Skills), но и общие (Soft Skills).

### **Практическая значимость**

В рамках предлагаемой программы изучение основ программирования на языке Java – это не столько средство подготовки к будущей профессиональной деятельности, сколько подготовка учащихся к вычислительному мышлению, которое облегчит для них решение комплексных задач XXI века. Способность поэтапно решать задачи, вовремя выявлять и вносить коррективы, выстраивать фрагменты в единый алгоритм для получения результата, смогут решить задачи в любой области.

**Новизна** данной программы заключается в формировании профессиональных знаний и умений в области программирования на языке Java с детьми среднего и старшего школьного возраста для последующего применения в учебной, познавательной деятельности и повседневной жизни.

### **Цели программы:**

создание условий для изучения методов программирования на языке программирования Java и использование его при разработке мобильных приложений, также подготовка к изобретательской деятельности в инженерно-техническом направлении.

Для достижения поставленной цели необходимо решить следующие **задачи**:

### **Образовательные:**

- дать представление о значении информационных технологий в развитии общества и предназначения языков программирования;
- познакомить с основными понятиями информатики непосредственно в процессе

создания информационного продукта;

- обучить методам программирования на языке Java, применяемых в современной вычислительной технике, и работе в интегрированных средах разработки;
- сформировать навыки проектирования мобильных приложений, создания программ и их отладки на мобильных устройствах;
- формировать и развивать навыки публичного выступления.

#### ***Воспитательные:***

- воспитать стремление учащихся к созданию собственных программных продуктов;
- привить стремление к получению качественного законченного результата в проектной деятельности;
- привить информационную культуру: ответственное отношение к информации с учетом правовых и этических норм её распространения;
- формировать правильное восприятие системы ценностей, принципов, правил информационного общества;

#### ***Развивающие:***

- способствовать развитию творческих способностей учащихся, познавательных интересов, развитию индивидуальности и самореализации;
- расширять технологические навыки при подготовке различных информационных материалов;
- развивать познавательные способности ребенка, память, внимание, пространственное мышление, аккуратность и изобретательность;
- развивать навыки инженерного мышления, умения работать по инструкциям и находить свои собственные пути решения поставленных задач;
- формировать навыки рефлексивной деятельности.

#### **Отличительные особенности программы**

Программа предназначена для учащихся, проявляющих повышенный интерес к программированию.

Ключевым элементом обучения является проектная деятельность. Проектная деятельность ориентирована на использование знаний, умений и навыков, полученных в ходе обучения, для постановки и решения практических задач, которые носят прикладной характер. Она позволяет учащимся участвовать в создании конкретного результата и научиться работать в условиях ограниченного времени, под руководством заказчика, презентовать проект, работать в команде, а также обрести навыки профессиональной коммуникации с контрагентами.

**Сроки реализации** данной программы составляют 1 год. Программа рассчитана на 144 учебных часа.

**Возраст учащихся:** средний и старший школьный возраст 14-17 лет. Программа рассчитана на детей 14–17 лет, проявляющих интерес к информационно-коммуникационным технологиям, программированию. Группы комплектуются на основании заявлений законных представителей учащихся (самих учащихся с 14 лет) из школьников разного возраста по результатам предварительной аттестации, получившим по итогам тестирования в рейтинговой таблице наивысший балл.

### **Психолого-педагогические особенности возрастной категории учащихся**

Программа составлена с учётом индивидуальных и возрастных особенностей учащихся. Подростковый возраст обычно характеризуют как переломный, переходный, критический, но чаще как возраст полового созревания.

Л. С. Выготский [29:288] подробно рассматривал проблему интересов в переходном возрасте, называя ее «ключом ко всей проблеме психологического развития подростка». Он писал, что все психологические функции человека на каждой ступени развития, в том числе и в подростковом возрасте, действуют не бессистемно, не автоматически и не случайно, а в определенной системе, направляемые конкретными, отложившимися в личности стремлениями, влечениями и интересами. В подростковом возрасте, подчеркивал Л. С. Выготский, имеет место период разрушения и отмирания старых интересов, и период созревания новой биологической основы, на которой впоследствии развиваются новые интересы.

Старший школьный возраст — ранняя юность. Главное психологическое приобретение ранней юности — это осознание своего внутреннего «Я», подросток подсознательно готовит себя к будущему. Ведущая деятельность в этом возрасте — учебно-профессиональная, в процессе которой формируются такие новообразования, как мировоззрение, профессиональные интересы, самосознание, мечта и идеалы. Старший школьный возраст — начальная стадия физической зрелости и одновременно стадия завершения полового развития.

Психолого-педагогические особенности учащихся определяют методы индивидуальной работы педагога с каждым из них, темпы прохождения образовательного маршрута.

### **Количество учащихся в объединении, их возрастные категории**

Данная программа ориентирована как на мальчиков, так и на девочек. Занятия групповые, в каждой группе от 8 до 12 человек. Набор в группы ведется с июля по сентябрь. В программе задействованы учащиеся среднего и старшего школьного возраста от 14 до 17

лет.

### **Особенности набора**

Набор в группы ведется *на основании результатов предварительной аттестации партнера программы «IT школа SUMSUNG»*. Прием на обучение в данную программу проводится на условиях, определенных локальным актом учреждения «Правила приема, перевода, отчисления обучающихся в МАОУ «Томский Хобби-центр», в соответствии с законодательством РФ (ч. 5 ст. 55 Федерального закона № 273-ФЗ) и партнера программы.

### **Форма занятий, режим и продолжительность занятий**

Форма реализации программы- *очная*. Общий объем часов по данной программе - 144 часов за 1 год. Занятия проходят 2 раза в неделю по 2 академических часа (1 академический час равен 45 минутам, что соответствует СанПиНу 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»).

### **Количество занятий и учебных часов в неделю.**

Количество занятий в неделю - 2, каждое по 2 академических часа. Один академический час составляет 45 мин, где 30 минут уделяется непосредственно работе за компьютером, 15 минут теоретической подготовке, проверке знаний. Занятие имеет следующую структуру:

15 минут – постановка учебной цели и задач, проверка знаний, объяснение нового материала.

30 минут – освоение нового материала на практике непосредственно за компьютером.

10 минут – проверка нового знания и его закрепление.

30 минут - работа над собственным проектом непосредственно за компьютером.

5 минут – рефлексия, выявление проблемных мест при работе по новой теме.

### **Ожидаемые результаты**

Основным результатом обучения является достижение высокой информационно-коммуникационной компетентности учащегося в области программирования мобильных приложений на языке Java. В результате освоения программы обучающийся должен приобрести следующие знания, умения и навыки:

#### ***знать:***

- правила работы с компьютером и технику безопасности;
- назначение и функции используемых информационных технологий;

- особенности работы с интегрированной средой разработки;
- базовые и сложные конструкции, способы организации процедур и функций в языке программирования Java;
- знание техники ведения проектной деятельности.

***уметь:***

- искать информацию с применением правил поиска в компьютерных сетях, некомпьютерных источниках информации;
- использовать приобретенные знания и умения в практической деятельности и повседневной жизни;
- создания простейших моделей объектов и процессов в виде изображений и чертежей,
- создания информационных объектов, в том числе для оформления результатов учебной работы;
- организации индивидуального информационного пространства, создания личных коллекций информационных объектов;
- эффективно использовать интегрированную среду разработки;
- проектировать мобильные приложения, создавать программы и выполнять их отладку на мобильных устройствах;
- писать код программы на языке Java;
- формировать цели, ставить задачи для её достижения в ходе решения проблемных ситуаций;

- презентовать свой продукт;

***обладать навыками:***

- самопрезентации;
- исследовательской, проектной и социальной деятельности, строить логическое доказательство;
- использования, создания и преобразования различных символьных записей, схем и моделей для решения познавательных и учебных задач в различных предметных областях, исследовательской и проектной деятельности;
- проектирования, разработки, документирования и представления собственных проектов в составе команды;
- самообразования - периодической оценкой своих успехов и собственной работы самими обучающимися.



## способы определения результативности реализации программы

Освоение данной программы сопровождается процедурой промежуточной аттестации учащихся 3 раза в год: в сентябре, декабре и мае, проводимой в формах, определенных программой, учебно-тематическим планом, как составной частью образовательной программы и в порядке, установленном приказами и Уставом МАОУ «Томский Хобби-центр» (ч. 1 ст. 58, ч. 2 ст. 30 Федерального закона № 273-ФЗ) и условиями партнера реализуемой программы - партнера Samsung.

Текущий контроль за реализацией программы проводится в разных формах:

- тестирование;
- устный опрос;
- самостоятельные и контрольные работы;
- участие в проектной деятельности.

### Критерии оценки

Работа в команде	Самостоятельность в выполнении задач	Участие в конкурсах	Специальные знания ( <i>по результатам Samsung партнера программы</i> )	Алгоритмическое, логическое мышление	Поиск и обработка информации
3 балла – умеет вести конструктивный диалог, эффективно распределять роли в команде с целью получения	3 балла- справляется с заданием самостоятельно, возможны подсказки со стороны педагога. 2 балла- справляется	3 балла – участие в разного вида конкурсах и проектах, неоднократно является победителем.	3 балла- 100% выполнении 2 балла- 70% выполнении 1 балл – 50%	3 балла- не возникает проблем с выявлением причины и следствия определенного решения 2 балла- возникают	3 балла- без проблем ориентируется в просторах интернета, находит информацию по запросу, умеет фильтровать и

<p>результата, доброжелателен ко всем участникам команды. 2 балла – не умеет рационально распределять роли в команде, при общении с членами команды возникают недопонимания. 1 балл – нетерпимым к окружающим, не умеет распределять роли в команде.</p>	<p>самостоятельно не всегда, прибегает к помощи педагога. 1 балл – без помощи в большинстве случаев справится с заданием не может.</p>	<p>2 балла – участие в е различного вида конкурсах и проектах, наблюдают ся разовые победы, входит в 10-ку лучших участников по рейтинговой таблице соревнований. 1 балл – участие в различного вида конкурсах и проектах, входит в первые 50% рейтинговой таблицы соревнований.</p>	<p>выполнении</p>	<p>проблемы с выявлением причины, либо следствия. 1 балл – у учащегося возникают проблемы и с тем и с другим.</p>	<p>отсеивать ненужную информацию. 2 балла – ориентируется в просторах интернета, находит информацию по запросу, иногда теряется в информационном потоке. 1 балл – может найти информацию в интернете, но подобрать подходящий информационный контент без помощи со стороны других участников образовательного процесса не в состоянии, сомневается в правильности выбора информации.</p>
--	--	--	-------------------	---	--

Педагогическое наблюдение, педагогический анализ результатов решения задач с использованием автоматизированной системы контроля знаний, результаты участия в интеллектуальных конкурсах всероссийского уровня.

### Учебно-тематический план

№	Наименование раздела, темы	Количество часов		
		Теория	Практика	Всего
	<b>Раздел 1. Основы программирования на языке Java.</b>	<b>6</b>	<b>12</b>	<b>18</b>
1	Тема 1.1. Знакомство со средой программирования на языке Java. Типы данных и операции.	1	1	2
2	Тема 1.2. Логические выражения.	1	1	2
3	Тема 1.3. Условный оператор.	1	1	2
4	Тема 1.4. Циклы while, do while	1	1	2
5	Тема 1.5. Цикл for. Массивы.	1	1	2
6	Тема 1.6. Методы (процедуры и функции).	1	1	2
7	Тема 1.7. Многомерные массивы.		2	2
8	Промежуточный тест по модулю 1.		4	4
	<b>Раздел 2. Введение в объектно-ориентированное программирование.</b>	<b>9</b>	<b>15</b>	<b>24</b>
9	Тема 2.1. Классы и объекты	2	2	4
10	Тема 2.2. Конструкторы и их перегрузка. Статические поля и методы.	2	2	4
11	Тема 2.3. Приемы тестирования и отладки на примерах со строками.	1	1	2
12	Тема 2.4. Знакомство с Android-разработкой.	1	1	2
13	Тема 2.5. Интерфейс Android-приложения.	1	3	4
14	Тема 2.6. Наследование и полиморфизм в Java.	2		2
15	Тема 2.7. Намерения.		2	2
16	Промежуточный тест по модулю 2.		4	4
	<b>Раздел 3. Основы программирования Android-приложений.</b>	<b>8</b>	<b>12</b>	<b>20</b>
17	Тема 3.1. Объектно-ориентированное проектирование	2		2
18	Тема 3.2. Ввод, вывод и исключение.	1	1	2
19	Тема 3.3. Внутренние и анонимные классы.	1	1	2
20	Тема 3.4. Параллелизм и синхронизация, потоки.	2		2
21	Тема 3.5. Фрагменты в Android.	1	1	2
22	Тема 3.6. Двумерная графика в Android-		2	2

	приложениях.			
23	Тема 3.7. Разработка игровых приложений SurfaceView.	1	3	4
24	Промежуточный тест по модулю 3.		4	4
	<b>Раздел 4. Алгоритмы и структуры данных на языке Java.</b>	<b>13</b>	<b>15</b>	<b>28</b>
25	Тема 4.1. Массивы. Класс ArrayList.	1	1	2
26	Тема 4.2. Связанные списки. Очереди, стеки, деки.	2	2	4
27	Тема 4.3. Списки в Android. Адаптеры.	1	1	2
28	Тема 4.4. СУБД. Реляционная модель.	2		2
29	Тема 4.5. СУБД SQLite. Основы языка SQL.	2	2	4
30	Тема 4.6. Рекурсия.	1	1	2
31	Тема 4.7. Деревья.	1	1	2
32	Тема 4.8. Алгоритмы сортировок.	1	1	2
33	Тема 4.9. Множества. Хеширование.	1	1	2
34	Тема 4.10. Ассоциативные массивы.	1	1	2
35	Промежуточный тест по модулю 4.		4	4
	<b>Раздел 5. Основы разработки серверной части мобильных приложений.</b>	<b>18</b>	<b>36</b>	<b>54</b>
36	Тема 5.1. IP – сети.	2	2	4
37	Тема 5.2. Веб-сервер, HTTP-запросы и ответы.	2	2	4
38	Тема 5.3. Клиент-серверная архитектура мобильных приложений.	2	2	4
39	Тема 5.4. Облачные платформы. REST-взаимодействие.	2	2	4
40	Тема 5.5. Серверные СУБД.	2	2	4
41	Промежуточный тест по модулю 5.		4	4
42	Работа над итоговым проектом.	4	6	10
43	Подготовка к защите проектов.		4	4
44	Защита итоговых проектов.		12	12
45	Рефлексия.	4		4
	<b>Итого</b>	<b>54</b>	<b>90</b>	<b>144</b>

## Содержание

### Раздел 1. Основы программирования на языке Java.

Уровень предъявления материала обеспечивает учащимся знакомство с основами программирования через создание простых мобильных приложений на языке программирования Java.

*Учащиеся должны знать:*

- базовые и сложные конструкции языка программирования Java;
- основные методы сортировки массивов (метод простого выбора, метод простого обмена, метод простой вставки);
- алгоритмы обработки символьных переменных;
- понятие подпрограммы, функции пользователя;
- способы организации процедур (подпрограммы) и функций, принципы модульной технологии программирования;
- алгоритмы обработки массивов (поиск элемента с заданными свойствами, удаление элементов, вставка элементов, замена элементов, обмен местами групп элементов).

*Учащиеся должны уметь:*

- устанавливать необходимый инструментарий для разработки приложений;
- составлять программный код с использованием разных типов данных и операций над ними;
- запускать созданные программы на мобильном устройстве;
- применять условные конструкции if/else, switch/case;
- решать задачи с использованием разных видов циклов;
- составлять программы обработки массивов;
- использовать функции и процедуры;
- искать информацию в свободных источниках и структурировать ее;
- объективно оценивать результаты своей работы.

*Формы занятий, используемые при изучении данного раздела:*

- лекционная,
- групповая работа,
- групповые консультации,
- индивидуальная работа,
- тестирование.

**Тема 1.1.** Знакомство со средой программирования на языке Java. Типы данных и

операции.

*Теория.* Понятия «алгоритм», «программа», «компилятор», «отладчик», «программирование», «язык программирования».

*Практика.* Установка и знакомство с интерфейсом сред разработки Eclipse, Android Studio, IntelliJ IDEA. Составление первой простой программы.

*Форма подведения итогов:* выполнение теста по теме.

### **Тема 1.2.** Логические выражения.

*Теория.* Операторы сравнения. Приоритет операций. Логические выражения. Тип boolean. Тернарная операция.

*Практика.* Принадлежность точки закрашенной области. Составление логических выражений.

*Форма подведения итогов:* выполнение теста по теме.

### **Тема 1.3.** Условный оператор.

*Теория.* Конструкция if/else. Условные выражения. Вложенность условных операторов.

*Практика.* Создание простой игры-погони с использованием переменных.

*Форма подведения итогов:* выполнение теста по теме.

### **Тема 1.4.** Циклы while, do while

*Теория.* Конструкция while, do while. Паттерны использования циклов. Оператор break.

*Практика.* Пример решения задачи с помощью цикла. Android-практикум: рисование узоров на Canvas.

*Форма подведения итогов:* выполнение теста по теме.

### **Тема 1.5.** Цикл for. Массивы.

*Теория.* Конструкция for. Оператор break с меткой. Оператор continue. Одномерные массивы. Цикл for each.

*Практика.* Примеры программ обработки массивов. Создание простейшей анимации. Графики функций. Модель солнечной системы. Маятники.

*Форма подведения итогов:* выполнение теста по теме.

### **Тема 1.6.** Методы (процедуры и функции).

*Теория.* Определение метода. Параметры метода. Возвращаемое значение. Области видимости переменной.

*Практика.* Применение функций.

*Форма подведения итогов:* выполнение теста по теме.

### **Тема 1.7.** Многомерные массивы.

*Теория.* Создание многомерных массивов и обращение к их элементам. Двумерный массив. Трехмерный массив. «Неровные» массивы.

*Практика.* Клетчатое поле в Android.

*Форма подведения итогов:* промежуточное тестирование по модулю.

## **Раздел 2. Введение в объектно-ориентированное программирование.**

Уровень предъявления материала обеспечивает учащимся возможность создавать программы для мобильных устройств на языке программирования Java.

*Учащиеся должны знать:*

- базовые и сложные конструкции языка программирования Java;
- способы организации процедур (подпрограммы) и функций, принципы модульной технологии программирования;
- определения и описания классов и объектов;
- понятия «инкапсуляция», «наследование», «полиморфизм»;
- конструкторы и способы их перегрузки;
- статические поля и методы.

*Учащиеся должны уметь:*

- запускать созданные программы на мобильном устройстве;
- тестировать и осуществлять отладку программ;
- объективно оценивать результаты своей работы.

*Формы занятий,* используемые при изучении данного раздела:

- лекционная,
- групповая работа,
- групповые консультации,
- индивидуальная работа,
- тестирование.

### **Тема 2.1. Классы и объекты.**

*Теория.* Классы и объекты. Инкапсуляция. Наследование. Полиморфизм. Описание класса.

*Практика.* Обзор классов-оболочек примитивных типов.

*Форма подведения итогов:* выполнение теста по теме.

### **Тема 2.2. Конструкторы и их перегрузка. Статические поля и методы.**

*Теория.* Конструкторы. Перегрузка методов. Ключевое слово this. Спецификаторы доступа. Статические компоненты класса.

*Практика.* Инициализация различных типов данных.

*Форма подведения итогов:* выполнение теста по теме.

### **Тема 2.3. Приемы тестирования и отладки на примерах со строками.**

*Теория.* Строки. Отладочный вывод и логирование. Использование отладчика.



Использование утверждений (assertions). Модульное тестирование.

*Практика.* Применение логирования.

*Форма подведения итогов:* выполнение теста по теме.

**Тема 2.4.** Знакомство с Android-разработкой.

*Теория.* Платформа Android. Структура проекта. Активности (Activity).

*Практика.* Создаем Android-проект. Запуск приложения.

*Форма подведения итогов:* выполнение теста по теме.

**Тема 2.5.** Интерфейс Android-приложения.

*Теория.* Язык разметки XML. XML-документ. Описание ресурсов Android с помощью XML. Строковые ресурсы.

*Практика.* Интерфейс пользователя. Разметка (layout). Компоненты (View).

*Форма подведения итогов:* выполнение теста по теме.

**Тема 2.6.** Наследование и полиморфизм в Java.

*Теория.* Понятие наследования. Защищенные члены класса. Ключевое слово super. Понятие полиморфизма. Абстрактные классы. Ключевое слово final. Понятие интерфейса.

*Практика.* Графическое описание структуры классов в UML. Иерархия наследования и преобразования типов.

*Форма подведения итогов:* выполнение теста по теме.

**Тема 2.7.** Намерения.

*Теория.* Контекст в Android. Намерения (Intent). Неявные намерения (Intent).

*Форма подведения итогов:* промежуточное тестирование по теме модуля.

### **Раздел 3. Основы программирования Android-приложений.**

В процессе работы по данному кейсу учащиеся овладеют навыками проектирования мобильных приложений, создания программ и отладки на мобильных устройствах.

*Учащиеся должны знать:*

- основные понятия объектно-ориентированного проектирования;
- назначение обработки исключений;
- внутренние и анонимные классы;
- типы сенсоров;
- определение параллелизма и синхронизации;
- основы графики в Android;
- общие способы создания потоков в Java.

*Учащиеся должны уметь:*

- генерировать идеи;
- обрабатывать исключения с помощью конструкции try-catch;

- работать с файлами, считывать и записывать в файл;
- управлять потоками;
- разрабатывать игровые приложения;
- разрабатывать 3D-игры с использованием фреймворка libGDX;
- объективно оценивать результаты своей работы.

*Формы занятий, используемые при изучении данного раздела:*

- лекционная,
- групповая (командная) работа,
- групповые консультации;
- защита проектов.

### **Тема 3.1. Объектно-ориентированное проектирование.**

*Теория.* Операции с дробями. Шаблоны и принципы проектирования.

*Практика.* Текстовый квест. Электронный журнал.

### **Тема 3.2. Ввод, вывод и исключение.**

*Теория.* Обработка исключений. Ключевые слова `throw` \ `throws`. Обработка исключения с помощью конструкции `try-catch`. Ключевое слово `finally`. Основные методы класса `Exception`.

*Практика.* Работа с файлами. Чтение \ запись в файл.

### **Тема 3.3. Внутренние и анонимные классы.**

*Теория.* Понятие внутреннего класса. Внутренние классы-члены. Локальные внутренние классы. Анонимные классы.

*Практика.* Текстовый квест.

### **Тема 3.4. Параллелизм и синхронизация, потоки.**

*Теория.* Общий способ создания потоков в Java. Реализация логики потока. Синхронизация потоков. Управление потоками. Блокировки.

### **Тема 3.5. Фрагменты в Android.**

*Теория.* Фрагменты. Класс `Fragment` и его методы. Взаимодействия между фрагментами и активностями.

*Практика.* Создание фрагментов. Управление фрагментами.

*Форма подведения итогов:* выполнение теста по теме.

### **Тема 3.6. Двумерная графика в Android-приложениях.**

*Практика.* Игра «Забавные птички»: игровое поле, создание класса `Sprite` для управления анимацией, добавление противника и контроль столкновений.

*Форма подведения итогов:* выполнение теста по теме.

### **Тема 3.7. Разработка игровых приложений SurfaceView.**

*Теория.* Общие подходы для реализации игровых приложений. Понятие игрового движка и его использование при разработке игры. Класс SurfaceView.

*Практика.* Создание приложений с помощью SurfaceView.

*Форма подведения итогов:* промежуточное тестирование по теме модуля.

#### **Раздел 4. Алгоритмы и структуры данных на языке Java.**

В процессе работы по данному кейсу учащиеся овладеют навыками работы с массивами и списками, научатся работать с системой управления базами данных, познакомятся с основами криптографии и криптоанализа.

*Учащиеся должны знать:*

- структуры данных (массивы, списки, множества);
- алгоритмы поиска элементов по значению;
- понятия стеков, очереди, дека;
- понятие реляционной модели, классификацию СУБД;
- определение рекурсии (линейной, ветвящейся);
- определение и разновидности деревьев;
- различные виды шифров (подстановки, перестановки);
- методы шифрования (симметричный, на основе алгоритма DES, ассиметричный, алгоритм RSA)
- понятие криптоанализа.

*Учащиеся должны уметь:*

- генерировать идеи;
- работать с СУБД (SQLite на Android-устройстве);
- производить сортировку (пузырьком, вставками, быстрая);
- осуществлять синхронизацию ассоциативных массивов;
- использовать контент-провайдеры;
- объективно оценивать результаты своей работы.

*Формы занятий,* используемые при изучении данного раздела:

- лекционная,
- групповая (командная) работа,
- групповые консультации;
- защита проектов.

#### **Тема 4.1. Массивы. Класс ArrayList.**

*Теория.* Структуры данных. Сложность алгоритмов. Массив — базовая структура данных. Операции с массивами. Класс Arrays. Сравнение объектов. Компараторы. Расширяемый массив. Класс ArrayList.

*Практика.* Алгоритмы поиска элементов по значению.

*Форма подведения итогов:* выполнение теста по теме.

**Тема 4.2.** Связанные списки. Очереди, стеки, деки.

*Теория.* Связные списки. Класс LinkedList. Сравнение ArrayList и LinkedList.

Коллекции. Интерфейс List. Стеки, очереди, деки.

*Практика.* Демонстрация работы очередей и стеков.

*Форма подведения итогов:* выполнение теста по теме.

**Тема 4.3.** Списки в Android. Адаптеры.

*Теория.* Список из ресурсов. ArrayAdapter. Собственная разметка.

*Практика.* Задание по Android-практикуму.

**Тема 4.4.** СУБД. Реляционная модель.

*Теория.* Реляционная модель. Реляционная БД из нескольких таблиц. Классификация СУБД.

**Тема 4.5.** СУБД SQLite. Основы языка SQL.

*Теория.* СУБД SQLite. Создание таблиц. Добавление записей в таблицу. Выборка данных. Изменение таблицы. Удаление записей. Агрегированные запросы.

*Практика.* Работа с базой данных SQLite на Android-устройстве.

**Тема 4.6.** Рекурсия.

*Теория.* Рекурсия в программировании и не только. Стек вызовов. Линейная рекурсия. Ветвящаяся рекурсия.

*Практика.* Разработка приложения, которое рисует рекурсивные фигуры на плоскости.

*Форма подведения итогов:* выполнение теста по теме.

**Тема 4.7.** Деревья.

*Теория.* Дерево. Разновидности деревьев. Понятие бинарного дерева. Понятие сбалансированного дерева. Библиотечный класс TreeSet.

*Практика.* Создание приложения с использованием деревьев.

*Форма подведения итогов:* выполнение теста по теме.

**Тема 4.8.** Алгоритмы сортировок.

*Теория.* Введение в сортировку данных. Сортировка пузырьком. Сортировка вставками. Быстрая сортировка.

*Практика.* Реализация сортировок в библиотечных классах Java.

*Форма подведения итогов:* выполнение теста по теме.

**Тема 4.9.** Множества. Хеширование.

*Теория.* Множества. Множество целых чисел от 0 до 100. Хеширование. Хеш-

таблица.

*Практика.* Интерфейс Set. Классы HashSet и TreeSet.

*Форма подведения итогов:* выполнение теста по теме.

**Тема 4.10.** Ассоциативные массивы.

*Теория.* Ассоциативный массив как набор пар «ключ — значение». Интерфейс Map. Классы для Map. Контейнер HashMap. Контейнер TreeMap. Синхронизация ассоциативных массивов. Хранение данных в Android Preferences.

*Практика.* Разработка приложения с сохранением строк и целого числа.

*Форма подведения итогов:* выполнение теста по теме и промежуточного тестирования по теме модуля.

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

В процессе работы по данному кейсу учащиеся овладеют основными навыками клиент-серверной разработки приложений.

*Учащиеся должны знать:*

- основные понятия компьютерных сетей;
- принцип работы глобальной сети Интернет и локальных сетей;
- понятие эталонной сетевой модели OSI;
- принципы передачи данных – методы управления;
- структуру пакетов;
- принципы работы протокола TCP/IP и примеры прикладных сетевых протоколов;
- основы маршрутизации в сетях TCP/IP;
- определение IP адреса, его версии и формат;
- понятия доменных имен, URL-ссылок;
- структуру HTTP-запроса;
- понятие веб-сервера
- облачные технологии;
- клиент-серверные архитектуры.

*Учащиеся должны уметь:*

- применять популярные сетевые команды;
- устанавливать необходимое программное обеспечение для работы с сервером;
- писать HTTP-приложения под Android;
- осуществлять сериализацию с помощью JSON;
- использовать библиотеку Retrofit;
- реализовать серверную часть приложения, выполненную ранее на языке Java, при помощи скриптового языка программирования PHP;

- объективно оценивать результаты своей работы.

*Формы занятий*, используемые при изучении данного раздела:

- лекционная,
- групповая (командная) работа,
- групповые консультации;
- защита проектов.

### **Тема 5.1. IP – сети.**

*Теория.* Интернет и протоколах TCP/IP. Адресация в IP-сетях. Версия интернет-протокола IPv4. Автоматизация процесса назначения IP-адресов. Доменные имена (DNS), URL-ссылки.

*Практика.* Сервисы работы с IP-адресами. Популярные сетевые команды.

### **Тема 5.2. Веб-сервер, HTTP-запросы и ответы.**

*Теория.* HTTP-протокол. Структура HTTP-запроса. Ответы сервера. Веб-сервер.

*Практика.* Реализация сервера на PHP.

*Форма подведения итогов:* выполнение теста по теме.

### **Тема 5.3. Клиент-серверная архитектура мобильных приложений.**

*Теория.* Архитектура клиент-сервер. Форматы JSON и XML. Сериализация. Библиотека Retrofit.

*Практика.* Отправка запросов из Android-приложений. Использование JSON и библиотеки Retrofit.

### **Тема 5.4. Облачные платформы. REST-взаимодействие.**

*Теория.* Облачные технологии. Модели развертывания. Модели обслуживания. Платформа как услуга. REST-взаимодействие. REST-аутентификация и OAuth-авторизация.

*Практика.* Синхронные и асинхронные запросы. Возможности REST-взаимодействия на примере одного из API Яндекс.Предиктора.

### **Тема 5.5. Серверные СУБД.**

*Теория.* Клиент-серверные архитектуры. Серверные СУБД. Настройка PostgreSQL и подключение к БД. Реализация back end части приложения на языке Java.

*Практика.* Реализация back end части приложения на языке PHP.

*Форма подведения итогов:* промежуточное тестирование по теме модуля.

## Комплекс организационно-педагогических условий реализации программы

### Календарный учебный график на 2023-2024 учебный год

Учебный период	Количество учебных недель	Дата начала учебного периода	Каникулы	
			Продолжительность	Организация деятельности по отдельному расписанию и плану
1 полугодие	17 недель	01 сентября	С 30.12 по 09.01	С 30.12.2023г. по 09.01.2024 г. участие в организации новогодних мероприятий
2 полугодие	19 недель	9 января	С 24 мая по 31 августа	27.05.2023 – 14.06.2024 – работа лагеря с дневным пребыванием детей с Работа загородных детских оздоровительно-образовательных лагерей «Лукоморье» и «Солнечная республика». Подготовка и участие в турнирах, соревнованиях.

Продолжительность учебного года – с 01.09.2023 по 23.05.2024 – 36 учебных недель

**Материально-техническое обеспечение:** для эффективности реализации программы необходимо оборудование и материалы:

- хорошо проветриваемое светлое помещение с хорошим естественным и искусственным освещением;
- столы для компьютера;
- компьютерные стулья;
- шкафы для дидактических материалов, пособий;
- канцтовары.
- персональный компьютер (на каждого участника);
- мультимедийный проектор;
- видеоматериалы разной тематики по программе;
- оргтехника;
- выход в сеть Internet;
- специальная и научно-популярная литература для педагога и учащихся.
- необходимое программное обеспечение операционная система Windows (версия не ниже 7); Android Studio; Eclipse;

- мобильные устройства на базе ОС Android.

**Кадровое обеспечение:** один педагог со специальным и педагогическим образованием, студенты технических СУЗов и ВУЗов.

### Методы организации учебного процесса

При организации занятий по программе для достижения поставленных целей и решения поставленных задач используются следующие **методы работы:**

1. Объяснительно-иллюстративный – предъявление информации различными способами (объяснение, рассказ, беседа, демонстрация и др).
2. Проблемный – постановка задачи и самостоятельный поиск ее решения учащимися.
3. Программированный – набор операций, которые необходимо выполнить в ходе выполнения практических работ.
4. Эвристический – метод творческой деятельности.
5. Многократный повтор способов работы, подходя к изучению последовательно, от простого к сложному, чередуя медленные темпы с быстрыми.
- 6.

### Направления воспитательной работы, реализуемые на Программе :

1. *Гражданско-патриотическое воспитание* осуществляется во время работы с учащимися над проектами. Программой предусмотрены занятия, на которых учащиеся рассуждают на тему «Патриот своей страны».
2. *Нравственное и духовное воспитание* так же реализуется в процессе обучения. На занятиях педагог прививает чувство уважения и любви к Родине, природе, культуре, этике.
3. *Воспитание положительного отношения к труду и творчеству* – это очень важное направление воспитательной работы. В начале занятия учащиеся готовят сами свое рабочее место. После занятий учащиеся сами убирают свое рабочее место: сортируют конструктор по форме и виду деталей, при загрязнении рабочего места, протирают свой стол. Таким образом, учащиеся приучаются к труду;
4. *Интеллектуальное воспитание* – происходит благодаря расширению кругозора учащихся;
5. *Здоровьесберегающее воспитание* – дети на переменах выходят из кабинета, и аудитория проветривается, на перемене педагог предлагает учащимся подвижные и настольные игры. Кроме того педагог контролирует санитарно-гигиенический



режим в кабинете, проводит профилактические беседы о том, что за столом необходимо сидеть ровно и нельзя качаться на стуле, пропагандирует здоровый образ жизни. Во время урока проводятся физкультминутки;

6. **Социокультурное и медиакультурное воспитание** – необходимо для формирования ключевых компетенций личности. С этой целью проводятся мероприятия вне занятий: участие в мастер-классах, конкурсах, фестивалях.
7. **Правовое воспитание и культура безопасности** - формирование у учащихся правовой культуры, представлений об основных правах и обязанностях, об уважении к взрослым, о правилах безопасного поведения на улице и в Хобби-центре;
8. **Воспитание семейных ценностей** – это, прежде всего, формирование у учащихся знаний в сфере этики и психологии семейных отношений. Педагог, проводит родительские собрания, анкетировает и консультирует родителей по различным вопросам, касающимся воспитания и т.д.;
9. **Экологическое воспитание** - воспитание у учащихся бережного отношения к природе. Создание проектов, которые не могут навредить урон природе и экологии.

### **Формы организации учебных занятий**

Основной тип занятий — комбинированный, сочетающий в себе элементы теории и практики. Большинство заданий программы выполняется самостоятельно с помощью персонального компьютера и необходимых программных средств.

Каждая тема программы начинается с постановки задачи — характеристики предметной области или конкретной программы на языке Python, которую предстоит изучить. С этой целью педагог проводит демонстрацию презентации или показывает саму программу, а также готовые работы, выполненные в ней. Закрепление знаний проводится с помощью практики – отработки умений самостоятельно решать поставленные задачи, соответствующих минимальному уровню планируемых результатов обучения.

Методика обучения ориентирована на индивидуальный подход. Для того чтобы каждый учащийся получил наилучший результат обучения, программой предусмотрены дополнительные задания для самостоятельной работы на домашнем компьютере. Такая форма организации обучения стимулирует интерес учащегося к предмету, активность и самостоятельность участников программы способствует объективному контролю глубины и широты знаний, повышению качества усвоения материала обучающимися,

позволяет педагогу получить объективную оценку выбранной им тактики и стратегии работы, методики индивидуального обучения и обучения в группе, выбора предметного содержания.

Для самостоятельной работы используются разные по уровню сложности задания, которые носят репродуктивный и творческий характер. Количество таких заданий в работе может варьироваться.

В ходе обучения проводится промежуточное тестирование по темам для определения уровня знаний учеников. Выполнение контрольных заданий способствует активизации учебно-познавательной деятельности и ведёт к закреплению знаний, а также служит индикатором успешности образовательного процесса.

### **Формы и методы контроля**

Для отслеживания результативности и динамики освоения ребенком образовательной программы педагогу необходимо понимать, какой уровень развития тех или иных навыков имеет учащийся при зачислении на обучение. Поэтому проводится **входящая диагностика** по диагностической карте промежуточной аттестации, чтобы определить уровень развития ребенка по определенным параметрам, заложенным картой.

- наблюдение;
- устный опрос;
- устный анализ самостоятельных работ;
- тестирование, самостоятельные и контрольные работы;
- участие в проектной деятельности;
- промежуточная аттестация - диагностика по параметрам – в сентябре, декабре и мае;
- обучение заканчивается написанием программы для решения одной из задач;

*Наблюдение* – необходимый педагогу метод для осуществления промежуточной и текущей аттестации, применяется педагогом постоянно.

*Устный опрос* – метод при котором педагог может оценить теоретически знания учащихся. Проводится в конце пройденной темы.

*Устный анализ самостоятельных работ* – дает возможность учащимся научиться логически мыслить и уметь высказать собственное суждение, поможет педагогу оценить логическое мышление учащихся. Проводится в конце пройденной темы.

*Тестирование, самостоятельные и контрольные работы* – помогут учащимся проверить свои знания по пройденным темам и оценить приобретенные умения, проводятся в конце

каждого раздела.

*Участие в проектной деятельности* – это условие, необходимое для становления самостоятельности учащихся, которые сформируют установку на самостоятельность, индивидуальные цели и возможности их реализации;

*Промежуточная аттестация* проводится три раза за учебный год: в начале учебного года – в сентябре, в середине учебного года – в декабре и в конце учебного года – в мае. Задачей промежуточной аттестации является определение эффективности педагогического воздействия и полноты освоения образовательной программы.

Параметры и критерии освоения программы в области специальных знаний языка программирования Java при разработке мобильных приложений определены партнером «Samsung». Педагог проводит промежуточную аттестацию по остальным параметрам, не относящимся к специальным знаниям в области языка программирования.

### Диагностическая карта

№	Ф.И.ребенка	Год рождения	Параметры						Итого
			Работа в команде	Самостоятельность в выполнении задач	Участие в конкурсах	Специальные знания (по результатам Яндекс-лицея - партнера программы)	Алгоритмическое, логическое мышление	Поиск и обработка информации в сети Интернет	

1 балл – низкий уровень

2 балла – средний уровень

3 балла – высокий уровень

При подсчете баллов по каждому учащемуся можно определить уровень освоения им программы, также по каждой группе и по объединению в целом.

Определение уровня каждого учащегося выводится после подсчета баллов по всем параметрам:

от 1 до 9 баллов – низкий уровень освоения программы учащимся;

от 10 до 16 баллов – средний уровень освоения программы учащимся;

от 16 до 18 баллов – высокий уровень освоения программы учащимся.

После этого подсчитывается процентное соотношение уровней освоения по группам и по объединению и результат заносится в сводную таблицу.

Сводная таблица

	Высокий уровень %	Средний уровень %	Низкий уровень %
На начало года реализации образовательной программы			
На середину года реализации образовательной программы			
На конец года реализации образовательной программы			

По этой таблице можно провести анализ результативности данной программы.

### **Список литературы**

#### **Литература для учащихся**

1. Белоусова А.С. ИТ-квантум тулкит. / Белоусова А. С., Юбзаев Т.И., – Москва: Фонд новых форм развития образования, 2019 –76 с.
2. Р. Никсон. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. / Р. Никсон, - Санкт-Петербург. Изд-во Питер, 2019, 816с.

#### **Веб-ресурсы**

1. Институт Национальный открытый университет. Основы функционирования веб-приложений. URL: <http://www.intuit.ru/studies/courses/4455/712/lecture/21291?page=2> (Дата посещения: 07.07.2020)
2. "IT школа SAMSUNG" онлайн ресурс по мобильной разработке компании Samsung Electronics. URL: <https://myitschool.ru/opencourse/course/view.php?id=3&section=1> (дата посещения: 06.07.2020)
3. CoderNet. PHP. Быстрый старт. М.А. Райтман. URL: [https://codernet.ru/books/php/php\\_bystryj\\_start\\_m\\_a\\_rajtman/](https://codernet.ru/books/php/php_bystryj_start_m_a_rajtman/) (Дата посещения: 06.07.2020)
4. Stepik. Ресурс для самообразования, образовательная платформа и конструктор онлайн-курсов. URL: <https://stepik.org/catalog>