



Правительство Санкт-Петербурга
Комитет по образованию

Государственное бюджетное общеобразовательное учреждение
Лицей №126
Калининского района Санкт-Петербурга

Принята
на педагогическом совете

Протокол № 13 от 07.06.2021

Утверждаю

директор _____ П.С. Розов
Приказ № 96 от 07.06.2021

РАБОЧАЯ ПРОГРАММА
к дополнительной общеобразовательной общеразвивающей программе
«Мобильная разработка. Начальный уровень»

Группа 1

1-ый год обучения
срок реализации: 72 часов
2021-2022 учебный год

Разработчик:
Курапов Михаил Михайлович
педагог дополнительного образования

Санкт-Петербург,
2021

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Трудно представить современный мир без мобильных устройств и разного рода гаджетов. То, что казалось ещё 20 лет назад фантастикой, сейчас распахнуло двери и стремительно врывается в наш мир, который даже по человеческим меркам ещё совсем недавно пользовался дисковыми телефонными аппаратами. А сейчас всё вокруг неумолимо и стремительно переходит к новому технологическому укладу. Согласно Элвину Тоффлеру, следующий мировой технологический и социальный уклад установит ценность человеческого ума и талантов как высший приоритет. При этом новом укладе мобильные устройства являются не только предтечей и воплощением будущего, и должны быть не только инструментом постижения мира, но и проводником, способствующим нашей трансформации. Посредством этих инструментов человечество должно преодолеть непростые ступени нового мира и застолбить своё место в грядущем новом мире. Первые мобильные приложения появились еще в далёком 1993 году. А первый мобильный телефон появился за 20 лет до этого, в ещё более далёком 1973 году, когда 3 апреля два инженера-разработчика компаний Bell Labs и Motorola осуществили первый разговор.

На сегодняшний день мир мобильной разработки представлен двумя основными операционными системами и технологиями на их базе: Android и iOS. С большим отрывом превалирует Android. Средства разработки под ОС Андроид можно поделить на две группы. Первая группа использует непосредственно Android SDK (пакет разработчика Андроид) и языки, соответственно, Java или Kotlin. В этом случае разработка ведётся в среде Android Studio (реже используется Eclipse, или, как вариант, можно использовать обычную версию IntelliJ и настроить специальный плагин для платформы Андроид). Удобнее использовать среду Android Studio, которая является специальной сборкой IntelliJ для создания мобильных приложений Андроид. Вторая группа средств активно развивается и представляет мобильную разработку на базе фреймворков. В данном курсе рассматривается разработка Андроид-приложений на языке программирования Java и Kotlin.

Обучающиеся **12-14** лет занимаются за индивидуальными компьютерами, что создает условия комфортности при выполнении заданий, предусмотренных программой: каждый ученик работает с оптимальной для него нагрузкой.

Продолжительность занятия 40 минут.

Цель первого года обучения

Целью данной программы является обучение учащихся разработке мобильных приложений на базе интегрированных сред разработки, получение практического опыта в данной сфере; развитие алгоритмического и объектного стиля мышления; формирование мотивации к получению образования в ИТ-сфере посредством организации практической деятельности.

Для достижения целей программы формируются следующие **задачи для первого года обучения**:

Обучающие (предметные):

- Сформировать и развить навыки алгоритмического и логического мышления, грамотной разработки программ.
- Ознакомить с принципами и методами функционального программирования.
- Ознакомить с принципами и методами объектно-ориентированного программирования.
- Сформировать навыки работы в интегрированной среде разработки на языке Kotlin.
- Изучить основы программирования Android приложений.
- Ознакомить с основами разработки серверной части мобильных приложений.

Развивающие (метапредметные):

- Развить умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности.

- Развить умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата; понимание, что в программировании длинная программа не всегда лучшая.
- Развить умение критически оценивать правильность решения учебно-исследовательской задачи.
- Развить умение корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями.
- Сформировать владение основами самоконтроля, способность к принятию решений.
- Развить умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ.
- Сформировать компетентность в области использования информационно-коммуникационных технологий (ИКТ-компетенция).
- Развить умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно-исследовательской деятельности.

Воспитательные (личностные):

- Сформировать ответственное отношение к учению, способности довести до конца начатое дело аналогично завершённым творческим учебным проектам.
- Сформировать способности к саморазвитию и самообразованию средствами информационных технологий на основе приобретённой благодаря иллюстрированной среде программирования мотивации к обучению и познанию.
- Развить опыт участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам.
- Сформировать коммуникативную компетенцию в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня
- Сформировать целостное мировоззрение, соответствующее современному уровню развития информационных технологий.
- Сформировать осознанное позитивное отношение к другому человеку, его мнению, результату его деятельности.
- Сформировать ценность здорового и безопасного образа жизни.
- Обеспечить усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

Занятия состоит из 3 частей:

- Вводной части
- Основной части
- Заключительной части

Во вводной части занятия сообщаются задачи занятия, осуществляется организация обучающихся и их функциональная и психологическая подготовка к основной части занятия, постановка проблемных вопросов и создание проблемных ситуаций.

В основной части занятия решаются задачи практического характера, изучаются пути решения проблем, производится промежуточный анализ достижений, презентация результатов работы.

Заключительная часть занятия направлена на проведение рефлексии. Завершает занятие подведение итогов.

Особенности организации образовательного процесса

Научность. Этот принцип предопределяет сообщение обучаемым только достоверных, проверенных практикой сведений, при отборе которых учитываются новейшие достижения науки и техники.

Доступность. Предусматривает соответствие объема и глубины учебного материала уровню общего развития учащихся в данный период, благодаря чему, знания и навыки могут быть сознательно и прочно усвоены.

Связь теории с практикой. Обязывает вести обучение так, чтобы обучаемые могли сознательно применять приобретенные ими знания на практике.

Воспитательный характер обучения. Процесс обучения является воспитывающим, ученик не только приобретает знания и нарабатывает навыки, но и развивает свои способности, умственные и моральные качества.

Сознательность и активность обучения. В процессе обучения все действия, которые отрабатывает ученик, должны быть обоснованы. Нужно учить, обучаемых, критически осмысливать, и оценивать факты, делая выводы, разрешать все сомнения с тем, чтобы процесс усвоения и наработки необходимых навыков происходили сознательно, с полной убежденностью в правильности обучения. Активность в обучении предполагает самостоятельность, которая достигается хорошей теоретической и практической подготовкой и работой педагога.

Наглядность. Объяснение техники сборки робототехнических средств на конкретных изделиях и программных продукта. Для наглядности применяются существующие видео материалы, а также материалы своего изготовления.

Систематичность и последовательность. Учебный материал дается по определенной системе и в логической последовательности с целью лучшего его освоения. Как правило, этот принцип предусматривает изучение предмета от простого к сложному, от частного к общему.

Прочность закрепления знаний, умений и навыков. Качество обучения зависит от того, насколько прочно закрепляются знания, умения и навыки учащихся. Не прочные знания и навыки обычно являются причинами неуверенности и ошибок. Поэтому закрепление умений и навыков должно достигаться неоднократным целенаправленным повторением и тренировкой.

Индивидуальный подход в обучении. В процессе обучения педагог исходит из индивидуальных особенностей детей (уравновешенный, неуравновешенный, с хорошей памятью или не очень, с устойчивым вниманием или рассеянный, с хорошей или замедленной реакцией, и т.д.) и опираясь на сильные стороны ребенка, доводит его подготовленность до уровня общих требований.

Материально-техническое оснащение:

Для реализации данного курса требуется следующее оборудование:

- Проектор и экран для демонстрации учебного материала
- Доска
- Персональные компьютеры для обучающихся

Требуемое программное обеспечение:

- Среда разработки Android Studio
- Пакет офисных приложений
- Браузер Google Chrome, Mozilla Firefox или «Яндекс Браузер».

Содержание программы 1-го года обучения.

ТЕМА 1. Знакомство со средой программирования Android и ее инструментарием.

Теория: Демонстрация готовых примеров программ и приложений. Проверка компьютерной грамотности. Проведение мотивирующих бесед с обучающимися. Комплектование группы. Инструктаж по технике безопасности.

ТЕМА 2. Здравствуй мир (Hello, World)!

Теория: Порядок создания, компиляции, сборки и запуска в IDE приложения Kotlin. Порядок инсталляции IDE в домашних условиях.

Практика: Знакомство с системой обучения в IT-Cube. Первичное знакомство с интерфейсом Android Studio

ТЕМА 3. Типы данных и операции

Теория: переменные, примитивные типы данных, арифметические операторы, выражения и присваивания. Представление данных в памяти: представление числовых (в прямом коде) и символьных данных. Представление отрицательных и вещественных чисел в памяти. Представление чисел в обратном и дополнительном коде. Поразрядные операции.

Практика: Отработка способов представления данных.

ТЕМА 4. Логические выражения.

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

Практика: Отработка способов представления логических выражений.

ТЕМА 5. Условные конструкции.

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

Практика: Отработка способов ветвления

ТЕМА 6. Итеративные конструкции

Теория: Итеративные конструкции while, do-while

Практика: Отработка применения данных итеративных конструкций

ТЕМА 7. Итеративные конструкции for. Массивы.

Теория: Вложенные циклы. Одномерные массивы, цикл for each в Kotlin. Разбор примеров нахождения максимума и минимума, поиска на числовых массивах

Практика: Отработка применения данных итеративных конструкций. Задачи с массивами.

ТЕМА 8. Методы. Указатели.

Теория: Изучение понятия функций на примере методов Kotlin. Передача параметров, возвращение результата. Видимость переменных. Указатели на примере языка Си. понятие, использование в качестве аргументов функций. Методы передачи аргументов в функцию. Динамическое выделение памяти

Практика: Отработка механизма методов. Отработка применения указателей

ТЕМА 9. Многомерные массивы.

Теория: Многомерные массивы. Неровные массивы

Практика: Отработка применения многомерных массивов

ТЕМА 10. Практикум

Практика: Закрепление изученных тем.

ТЕМА 11. Контрольное тестирование по модулю.

Практика: Проведение тестирования.

ТЕМА 12. Понятие класса и объекта.

Теория: 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.

Практика: Описание протокола класса. Обзор классов, соответствующих примитивным типам

ТЕМА 13. Работа с экземплярами класса

Теория: Конструкторы и деструкторы. Статические методы. Открытые и закрытые поля. Доступ к полям объекта. Инициализация переменных, массивов, полей классов в конструкторе. Конструктор по умолчанию. Перегрузка методов на примере конструкторов

Практика: Работа с экземплярами класса

ТЕМА 14. Строки. Основы тестирования и отладки

Теория: Начальные приемы тестирования и отладки, сценарии тестирования на примерах со строками

Практика: Отработка указанных приемов

ТЕМА 15. Знакомство с Android разработкой

Теория: ОС Android. Среда разработки. Принципиальная архитектура Android-приложения

Практика: Создание первого Android приложения в IDE. Изучение жизненного цикла Activity

ТЕМА 16. Интерфейс Android приложения

Теория: Построение простейшего интерфейса пользователя. Язык разметки XML. Описание ресурсов Android с помощью XML. Разметки (Layouts) и их применение. Представления (Views)

Практика: Создание приложения с простейшим интерфейсом и обработкой событий

ТЕМА 17. Наследование, инкапсуляция и полиморфизм

Теория: Производные классы и наследование. Защищенные части классов и правила доступа для классов и объектов в Kotlin. Сравнение иерархии классов и контейнеризации классов. Полиморфные методы и позднее связывание в Kotlin. Абстрактные методы и классы, интерфейсы.

Практика: Создание класса, хранящего информацию о человеке (возраст, имя). Реализация иерархии классов «Геометрические фигуры».

ТЕМА 18. Context и Intent. Параметризованные типы

Теория: Понятия контекста (Context) и намерения (Intent) в Android. (Generic) как пример статического полиморфизма в Kotlin

Практика: Намерения (Intents) в Android. Использование Generic

ТЕМА 19. Практикум

Практика: закрепление изученного материала

ТЕМА 20. Постановка задачи на индивидуальный проект

Практика: Формирование постановки задачи на индивидуальный проект (срок публикации в системе обучения темы индивидуального проекта – окончание 2 модуля)

ТЕМА 21. Контрольное тестирование по модулю

Практика: Проведение тестирования.

ТЕМА 22. Практикум ООП проектирования.

Теория: Разбор кейсов проектирования архитектуры классов приложения. Диаграммы UML

Практика: Разбор задания мини-проекта. Получение диаграммы классов мини-проекта

ТЕМА 23. Ввод-вывод в Kotlin. Обработка исключений

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

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

ТЕМА 24. Внутренние и анонимные классы

Теория: Внутренние и анонимные классы на примерах обработчиков событий пользовательского интерфейса

Практика: Разбор примеров использования Listener. Работа над мини-проектом

ТЕМА 25. Решение задач.

Практика: Закрепление изученного материала

ТЕМА 26. Защита прототипа индивидуального проекта

Практика: Защита прототипа индивидуального проекта

Планируемые результаты первого года обучения

Обучающие (предметные):

- Сформированы и развиты навыки алгоритмического и логического мышления, грамотной разработки программ.
- Изучены принципы и методы функционального программирования.

- Изучены принципы и методы объектно-ориентированного программирования.
- Сформированы навыки работы в интегрированной среде разработки на языке Kotlin.
- Изучены основы программирования Android приложений.
- Изучены основы разработки серверной части мобильных приложений.

Развивающие (метапредметные):

- Развито умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности.
- Развито умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата; понимание, что в программировании длинная программа не всегда лучшая.
- Развито умение критически оценивать правильность решения учебно-исследовательской задачи.
- Развито умение корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями.
- Сформировано владение основами самоконтроля, способность к принятию решений.
- Развито умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ.
- Сформирована компетентность в области использования информационно-коммуникационных технологий (ИКТ-компетенция).
- Развито умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно-исследовательской деятельности.

Воспитательные (личностные):

- Сформированы ответственное отношение к учению, способности довести до конца начатое дело аналогично завершённым творческим учебным проектам.
- Сформированы способности к саморазвитию и самообразованию средствами информационных технологий на основе приобретённой благодаря иллюстрированной среде программирования мотивации к обучению и познанию.
- Развиты опыт участия в социально значимых проектах, повышен уровень самооценки благодаря реализованным проектам.
- Сформирована коммуникативная компетенция в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня.
- Сформировано целостное мировоззрение, соответствующее современному уровню развития информационных технологий.
- Сформировано осознанное позитивное отношение к другому человеку, его мнению, результату его деятельности.
- Сформирована ценность здорового и безопасного образа жизни.
- Усвоены правила индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

Календарно-тематическое планирование 1-го года обучения

№ п/п	Тема занятия	Кол-во часов	Дата занятия	
			план	факт
Сентябрь				
1.	Вводное занятие. Инструктаж по технике безопасности.	1	1 неделя	

	Знакомство			
2.	Знакомство с Kotlin. Комплектование групп	1	2 неделя	
3.	Первая программа. Функции	1	3 неделя	
4.	Переменные, константы и типы	1	4 неделя	
Октябрь				
5.	Встроенные типы	1	5 неделя	
6.	Переменные доступные только для чтения	1	6 неделя	
7.	Условные конструкции	1	7 неделя	
8.	Условное выражение when	1	8 неделя	
Ноябрь				
9.	Циклы	1	9 неделя	
10.	Функции	1	10 неделя	
11.	Анатомия функций	1	11 неделя	
12.	Решение задач	1	12 неделя	
Декабрь				
13.	Анонимные функции	1	13 неделя	
14.	Решение задач	1	14 неделя	
15.	Анонимные функции	1	15 неделя	
16.	Ключевое слово it	1	16 неделя	
Январь				
17.	Null — безопасность и исключения	1	17 неделя	
18.	Строки	1	18 неделя	
19.	Работа со строками	1	19 неделя	
20.	Числа	1	20 неделя	
Февраль				
21.	Числовые типы. Преобразование типов. Манипуляции с битами	1	21 неделя	
22.	Стандартные функции	1	22 неделя	
23.	Работа со стандартными функциями	1	23 неделя	
24.	Списки и множества	1	24 неделя	
Март				
25.	Списки и множества. Решение задач	1	25 неделя	
26.	Ассоциативные массивы	1	26 неделя	
27.	Классы	1	27 неделя	
28.	Наследование классов	1	28 неделя	
Апрель				
29.	Объекты.	1	29 неделя	
30.	Интерфейсы и абстрактные классы	1	30 неделя	
31.	Архитектура Android	1	31 неделя	
32.	Инструменты разработчика	1	32 неделя	
Май				
33.	Архитектура UI в Android	1	33 неделя	
34.	Activity	1	34 неделя	
35.	Решение задач	1	35 неделя	
36.	Итоговое занятие.	1	36 неделя	

СОДЕРЖАНИЕ ЗАНЯТИЙ

1. Знакомство со средой программирования Android и ее инструментарием.
Теория: Демонстрация готовых примеров программ и приложений. Проверка компьютерной грамотности. Проведение мотивирующих бесед с обучающимися. Комплектование группы. Инструктаж по технике безопасности.
2. Здравствуй мир (Hello, World)!
Теория: Порядок создания, компиляции, сборки и запуска в IDE приложения Kotlin. Порядок инсталляции IDE в домашних условиях.
Практика: Знакомство с системой обучения в IT-Cube. Первичное знакомство с интерфейсом Android Studio
3. Типы данных и операции
Теория: переменные, примитивные типы данных, арифметические операторы, выражения и присваивания. Представление данных в памяти: представление числовых (в прямом коде) и символьных данных. Представление отрицательных и вещественных чисел в памяти. Представление чисел в обратном и дополнительном коде. Поразрядные операции.
Практика: Отработка способов представления данных.
4. Логические выражения.
Теория: Тип Boolean. Операции отношения, логические и тернарные операции
Практика: Отработка способов представления логических выражений.
5. Условные конструкции.
Теория: Понятие блока, область действия блоков. Условные конструкции: if-else, when
Практика: Отработка способов ветвления
6. Итеративные конструкции
Теория: Итеративные конструкции while, do-while
Практика: Отработка применения данных итеративных конструкций
7. Итеративные конструкции for. Массивы.
Теория: Вложенные циклы. Одномерные массивы, цикл for each в Kotlin. Разбор примеров нахождения максимума и минимума, поиска на числовых массивах
Практика: Отработка применения данных итеративных конструкций. Задачи с массивами.
8. Методы. Указатели.
Теория: Изучение понятия функций на примере методов Kotlin. Передача параметров, возвращение результата. Видимость переменных. Указатели на примере языка Си. понятие, использование в качестве аргументов функций. Методы передачи аргументов в функцию. Динамическое выделение памяти
Практика: Отработка механизма методов. Отработка применения указателей
9. Многомерные массивы.
Теория: Многомерные массивы. Неровные массивы
Практика: Отработка применения многомерных массивов
10. Практикум
Практика: Закрепление изученных тем.
11. Контрольное тестирование по модулю.
Практика: Проведение тестирования.
12. Понятие класса и объекта.
Теория: 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.
Практика: Описание протокола класса. Обзор классов, соответствующих примитивным типам
13. Работа с экземплярами класса

Теория: Конструкторы и деструкторы. Статические методы. Открытые и закрытые поля. Доступ к полям объекта. Инициализация переменных, массивов, полей классов в конструкторе. Конструктор по умолчанию. Перегрузка методов на примере конструкторов
Практика: Работа с экземплярами класса

14. Строки. Основы тестирования и отладки

Теория: Начальные приемы тестирования и отладки, сценарии тестирования на примерах со строками

Практика: Отработка указанных приемов

15. Знакомство с Android разработкой

Теория: ОС Android. Среда разработки. Принципиальная архитектура Android-приложения

Практика: Создание первого Android приложения в IDE. Изучение жизненного цикла Activity

16. Интерфейс Android приложения

Теория: Построение простейшего интерфейса пользователя. Язык разметки XML. Описание ресурсов Android с помощью XML. Разметки (Layouts) и их применение. Представления (Views)

Практика: Создание приложения с простейшим интерфейсом и обработкой событий

17. Наследование, инкапсуляция и полиморфизм

Теория: Производные классы и наследование. Защищенные части классов и правила доступа для классов и объектов в Kotlin. Сравнение иерархии классов и контейнеризации классов. Полиморфные методы и позднее связывание в Kotlin. Абстрактные методы и классы, интерфейсы.

Практика: Создание класса, хранящего информацию о человеке (возраст, имя). Реализация иерархии классов «Геометрические фигуры».

18. Context и Intent. Параметризованные типы

Теория: Понятия контекста (Context) и намерения (Intent) в Android. (Generic) как пример статического полиморфизма в Kotlin

Практика: Намерения (Intents) в Android. Использование Generic

19. Практикум

Практика: закрепление изученного материала

20. Постановка задачи на индивидуальный проект

Практика: Формирование постановки задачи на индивидуальный проект (срок публикации в системе обучения темы индивидуального проекта – окончание 2 модуля)

21. Контрольное тестирование по модулю

Практика: Проведение тестирования.

22. Практикум ООП проектирования.

Теория: Разбор кейсов проектирования архитектуры классов приложения. Диаграммы UML

Практика: Разбор задания минипроекта. Получение диаграммы классов мини-проекта

23. Ввод-вывод в Kotlin. Обработка исключений

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

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

24. Внутренние и анонимные классы

Теория: Внутренние и анонимные классы на примерах обработчиков событий пользовательского интерфейса

Практика: Разбор примеров использования Listener. Работа над мини-проектом

25. Решение задач.

Практика: Закрепление изученного материала

26. Защита прототипа индивидуального проекта

Практика: Защита прототипа индивидуального проекта

Диагностическая карта определения результатов по программе
«Мобильная разработка. Начальный уровень»
1-го года обучения

Группа _____ учебный год _____ педагог _____

даты проведения:

вход: _____

1 полугодие: _____

2 полугодие _____

№	Фамилия, имя	1	2	3	4	5	6	7	Итог

Оценка производится по 5-балльной шкале:

"5" – отлично, "4" – хорошо, "3" – посредственно, "2" – плохо.