

Государственное бюджетное общеобразовательное учреждение Лицей №126

Калининского района Санкт-Петербурга

Принята	Утверждаю			
на педагогическом совете				
	директор	П.С. Розов		
Протокол № 13 от 07.06.2021	Приказ № 96 от (6 от 07.06.2021		

Дополнительная общеобразовательная общеразвивающая программа «Мобильная разработка. Начальный уровень»

возраст учащихся 12-14 лет срок реализации:1 год

Разработчик: Курапов Михаил Михайлович Педагог дополнительного образования

Санкт-Петербург,

І. 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.

1.1. Направленность дополнительной общеобразовательной общеразвивающей программы

Дополнительная общеобразовательная общеразвивающая программа «Мобильная разработка. Начальный уровень» обладает технической направленностью и предназначена для использования в системе дополнительного образования детей.

1.2. Актуальность и отличительная особенность ДООП

Актуальность данной программы определена тем, что одним из основных аспектов влияющих на внедрение информационных технологий в жизнедеятельность общества, является умение оперативно и качественно работать с мобильным устройством, используя для этого все современные технические и программные средства и методы.

Быстрое развитие информационных технологий оказывает существенное влияние на наше общество проникая в каждую из сфер жизнедеятельности.

С развитием сферы мобильных технологий появляются все более емкие и удобные среды разработки программного обеспечения, которые направлены на формирование и поддержку интереса молодежи к изучению информатики и программирования, а также повышение престижности ИТ-специальностей в глазах обучающихся.

Отличительная особенность программы в том, что одной из популярных сред по созданию мобильных приложений является «Android Studio» от компании Google. Для создания приложений для своего мобильного устройства на платформе Android, понадобится Google-аккаунт и среда для программирования «Android Studio».

С помощью данной среды программирования, обучающиеся поймут, что программирование доступно, интересно и актуально.

1.3 Адресат ДООП, объем и срок реализации программы

Дополнительная общеобразовательная общеразвивающая программа «Мобильная разработка. Начальный уровень» предполагает начальный уровень компьютерной грамотности. Программа рассчитана на учащихся 12-14 лет, при этом возраст учащихся внутри одной группы может не совпадать.

Общее количество учебных часов запланированных на весь период обучения: 72 академических часа. Срок реализации программы 1 год.

1.4. Цель и задачи ДООП

Целью дополнительной общеобразовательной общеразвивающей программы является:

- обучение обучающихся разрабатывать мобильные приложения на базе интегрированных сред разработки, получить практический опыт в данной сфере;
- развитие алгоритмический и объектный стиль мышления; формировать мотивацию к получению образования в ИТ-сфере посредством организации практической деятельности.

ДООП «Мобильная разработка. Начальный уровень» ставит перед собой следующие задачи:

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 уровняю
- 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.

1.5. Условия реализации ДООП

Программа ориентирована на обучающихся 12-14 лет.

Набор в группу осуществляется по желанию, предполагает начальный уровень компьютерной грамотности. Возраст учащихся внутри одной группы может не совпадать.

Количество детей в группе до 15 человек.

Общее количество учебных часов запланированных на весь период обучения: 72 академических часа. Срок реализации программы 1 год.

Режим занятий:

1 год обучения - 1 раз в неделю по 2 академических часа — 72 часа в год. Продолжительность занятия 40 минут. Занятия проводятся с перерывом 5-10 минут.

Обучение по программе предполагает виды занятий:

- учебное занятие;
- занятие игра;
- открытое занятие;

Формы организации деятельности учащихся на занятии:

- фронтальная со всей группой;
- индивидуальная самостоятельная работа учащегося над проектом под руководством и с консультацией педагога;
- групповая если над одним проектом работают несколько человек.

Формы проведения занятий:

- практическое занятие;
- теоретическое занятие;
- комбинированное занятие;
- презентация проектов;
- конкурс;
- открытое занятие.

Занятие состоит из 3 частей:

- Вводной части
- Основной части
- Заключительной части

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

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

<u>Заключительная часть</u> 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:

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

1.6. Планируемые результаты

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 уровняю
- 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.

Фомой подведения итогов реализации ДООП «Мобильная разработка. Начальный уровень» является:

- участие в школьных праздниках и мероприятиях;
- презентация и защита собственного проекта;
- регулярные текущие проверки знаний и практических навыков.

Проверка знаний и умений обучающихся является неотъемлемой и важной составляющей частью процесса обучения, они строятся на принципах систематичности и проводятся в течение всего процесса обучения.

ІІ.УЧЕБНЫЙ ПЛАН

2.1 Учебный план 1-го года обучения.

№	Наименование темы	Количество часов		Формы	
п/п		Всего	Теория	Практика	контроля
1.	Знакомство со средой программирования Android и ее инструментарием. Демонстрация готовых примеров программ и приложений. Проверка компьютерной грамотности. Проведение мотивирующих бесед с обучающимися. Комплектование группы. Инструктаж по технике безопасности.	8	8	0	Беседа
2.	Здравствуй мир! (Kotlin)	2	1	1	Устный опрос
3.	Типы данных и операции	2	1	1	Устный опрос
4.	Логические выражения.	2	1	1	Устный опрос
5.	Условные конструкции.	2	1	1	Устный опрос
6.	Итеративные конструкции	2	1	1	Устный опрос
7.	Итеративные конструкции (for, when). Массивы	2	1	1	Устный опрос
8.	Методы. Указатели.	2	1	1	Устный опрос
9.	Многомерные массивы.	2	1	1	Устный опрос
10.	Практикум	6	0	6	Устный опрос
11.	Контрольное тестирование по модулю	2	0	2	Устный опрос
12.	Понятие класса и объекта	2	1	1	Устный опрос
13.	Работа с экземплярами класса	4	2	2	Разработанн ые программы
14.	Строки. Основы тестирования и отладки	2	1	1	Устный опрос

15.	Знакомство с Android разработкой	2	1	1	Устный
					опрос
16.	Интерфейс Android приложения	4	2	2	Разработанн
					ые
					программы
17.	Наследование, инкапсуляция и	4	2	2	Разработанн
	полиморфизм в Kotlin				ые
					программы
18.	Context и Intent. Параметризованные типы	2	1	1	Устный
					опрос
19.	Практикум	2	0	2	
20.	Постановка задачи на индивидуальный	2	1	1	
	проект	2			
21.	Контрольное тестирование по модулю		0	2	
22.	Практикум ООП проектирования.	4	0	4	Разработанн
					ые
					программы
23.	Ввод-вывод в Kotlin. Обработка	2	1	1	Устный
	исключений				опрос
24.	Внутренние и анонимные классы	4	2	2	Разработанн
					ые
					программы
25.	Решение задач	2	0	2	Разработанн
					ые
					программы
26.	Итоговое занятие.	2	0	2	Презентация
					проектов
	ОТОТИ	72	30	42	

ІІ. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Год	Дата	Дата	Всего	Количест	Режим занятий
обучения	начала	окончания	учебных	ВО	
	обучени	обучения	недель	учебных	
	Я	ПО		часов	
	по	программе			
	програм				
	ме				
1 год обучения					
1 год			36	72	1 раз в неделю по 2
		50	12	часа	

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