

Министерство образования и занятости населения Приморского края
Краевое государственное автономное профессиональное образовательное
учреждение
«Владивостокский судостроительный колледж»

«УТВЕРЖДАЮ»
Директор КГА ПОУ «ВСК»
_____ Глушкова И.В.

«3» сентября 2024 г.

Основы программирования на Kotlin

Дополнительная общеобразовательная общеразвивающая программа
технической направленности

Возраст учащихся: 14-17 лет
Срок реализации программы: 1 год

Преподаватель
Виноградова А.В.

г. Владивосток
2024 г.

Раздел № 1. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ПРОГРАММЫ

1.1 Пояснительная записка

Актуальность программы

Программа разработана в соответствии с методическими рекомендациями по созданию и функционированию центров цифрового образования «ИТ-Куб» от 18 апреля 2023 года, а также федеральными нормативными правовыми актами в области дополнительного образования, государственными требованиями к образовательным программам системы дополнительного образования детей, а также локальными нормативными правовыми актами организации.

Обучение программированию на языке Kotlin является актуальным и востребованным в современном мире информационных технологий. Kotlin – это современный язык программирования, который становится все более популярным в различных областях, таких как:

Разработка мобильных приложений для Android: Kotlin является официальным языком для Android-разработки и активно используется для создания высококачественных мобильных приложений.

Разработка веб-приложений: Kotlin может использоваться для бэкенд-разработки, благодаря своей эффективности и удобству работы с современными веб-фреймворками.

Разработка серверных приложений: Kotlin подходит для создания надежных и масштабируемых серверных приложений, благодаря своей статической типизации и поддержке функционального программирования.

Разработка игр: Kotlin может применяться для разработки игр, используя популярные игровые движки, благодаря своей производительности и поддержке современных технологий.

Программа «Основы программирования на Kotlin» предназначена как для тех, кто только начинает изучать программирование, так и для тех, кто уже имеет некоторый опыт работы с другими языками программирования. Она позволит освоить основы программирования, заложить фундамент для дальнейшего профессионального развития в области разработки программного обеспечения.

Направленность программы: техническая.

Язык реализации программы: русский.

Уровень освоения: стартовый (ознакомительный).

Отличительные особенности: программа не дублирует школьный курс информатики, а является его дополнением с профориентационными целями. Преимущество программы выражено в подборе интерактивных и практико-ориентированных форм занятий, способствующих формированию основных компетенций у обучающихся.

В программе предложен авторский подход в части структурирования учебного материала, определения последовательности его изучения, путей формирования системы знаний, умений и способов деятельности, развития, воспитания и социализации обучающихся.

Адресат программы: школьники города Владивостока 14-17 лет.

Особенности организации образовательного процесса:

Для обучения на дополнительной общеобразовательной программе «Основы программирования на Kotlin» не требуется начальных знаний, так как уровень освоения программы – “Стартовый (ознакомительный)”. Дети могут обучаться по данной программе без каких-либо знаний в этой области.

- Режим занятий: занятия проводятся в группах от 6 до 12 человек, продолжительность одного занятия — 2 академических часа (1 час – 25 минут);

- Продолжительность образовательного процесса: общая продолжительность программы – 1 год, 72 часа, 2 часа в неделю.

1.2 Цель и задачи программы

Цель программы: формирование базовых навыков программирования у школьников города Владивостока 14-17 лет с помощью языка “Kotlin”.

Задачи программы:

Воспитательные:

1. Сформировать способности к саморазвитию и самообразованию средствами информационных технологий на основе иллюстрированной среды программирования, мотивации к обучению и познанию;
2. Сформировать умения работать индивидуально и в группе для решения поставленной задачи;
3. Сформировать целостное мировоззрение, соответствующее современному уровню развития информационных технологий;
4. Сформировать осознанное позитивное отношение к другому человеку, его мнению, результату его деятельности;
5. Обеспечить усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой;
6. Сформировать культуру начального программирования.

Развивающие:

1. Развить умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
2. Развить умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата;
3. Развить умение критически оценивать правильность решения учебно-исследовательской задачи;
4. Сформировать владение основами самоконтроля, способность к принятию решений;
5. Формировать мотивацию к профессиональному самоопределению учащихся.

Обучающие:

1. Знание основных алгоритмических структур и умение применять его для построения алгоритмов решения задач по их математическим моделям, умения и навыки составления простых программ, по построенному алгоритму, на Kotlin;
2. Умения и навыки эффективного использования основных типов прикладных программ (приложений) общего назначения и информационных систем для решения с их помощью практических задач;
3. Умение грамотно интерпретировать результаты решения практических задач с помощью информационных технологий, применять полученные результаты в практической деятельности.

1.3 Содержание программы

Учебный план программы «Основы программирования на Kotlin»

№ п/п	Наименование раздела	Количество часов			Форма аттестации
		Теория	Практика	Всего	
1	Введение. Знакомство и ознакомление с ТБ.	2	0	2	Беседа
2	Знакомство с Kotlin. Первая программа.	1	1	2	Устный опрос
3	Типы данных. Переменные.	3	3	6	Устный опрос
4	Ветвление в языке Kotlin.	2	2	4	Тестирование
5	Решение задач по пройденным темам.	0	8	8	Зачёт
6	Циклы в языке Kotlin.	2	4	6	Тестирование
7	Решение задач по пройденным темам.	0	8	8	Зачёт
8	Коллекции в языке Kotlin.	3	5	8	Тестирование
9	Решение задач по пройденным темам.	0	6	6	Зачёт
10	Функции в языке Kotlin.	2	2	4	Устный опрос
11	Объектно-ориентированное программирование в Kotlin.	3	5	8	Тестирование
12	Решение задач по пройденным темам.	0	8	8	Зачёт
13	Основы программирования на Kotlin.	0	2	2	Контрольная работа
Итого		18	54	72	

Содержание учебного плана

1. Тема: Введение. Знакомство и ознакомление с ТБ

Теория. Введение в образовательную программу. Ознакомление обучающихся с программой, приёмами и формами работы. Вводный инструктаж по ТБ.

2. Тема: Знакомство с Kotlin. Первая программа.

Теория. История, преимущества и применение Kotlin.

Практика. Установка и настройка окружения разработки. Разработка первой программы.

3. Тема: Типы данных. Переменные.

Теория. Именованье переменных. Типы данных. Ввод и вывод данных.

Практика. Разработка линейных программ.

4. Тема: Ветвление в языке Kotlin.

Теория. Условный оператор. Логические операторы. Вложенные условные операторы.

Практика. Написание программ с использованием оператора ветвления if..else, логических операторов.

5. Тема: Решение задач по пройденным темам.

Практика. Решение задач по темам «Типы данных. Переменные.», «Ветвление в языке Kotlin.».

6. Тема: Циклы в языке Kotlin.

Теория. Циклические конструкции. Цикл for. Цикл while. Цикл do...while. Оператор continue. Оператор break.

Практика. Написание программ с использованием цикла while. Написание программ с использованием цикла do..while. Написание программ с использованием цикла for.

7. Тема: Решение задач по пройденным темам.

Практика. Решение задач по темам «Ветвление в языке Kotlin.», «Циклы в языке Kotlin.».

8. Тема: Коллекции в языке Kotlin.

Теория. Массивы. Списки. Множества. Словари.

Практика. Написание программ с использованием массивов, списков, множеств, словарей.

9. Тема: Решение задач по пройденным темам.

Практика. Решение задач по теме «Коллекции в языке Kotlin.».

10. Тема: Функции в языке Kotlin.

Теория. Понятие «Функция». Лямбда-выражения и анонимные функции. Работа с функциями.

Практика. Написание программ с использованием функций.

11. Тема занятия: Объектно-ориентированное программирование в Kotlin.

Теория. Основные принципы ООП. Классы и объекты. Наследование. Модификаторы доступа. Полиморфизм. Интерфейсы. Абстрактные классы.

Практика. Написание программ используя основные принципы ООП.

12. Тема: Решение задач по пройденным темам.

Практика. Решение задач по всем пройденным темам за курс.

13. Тема: Основы программирования на Kotlin.

Практика. Задачи по программе всего курса.

1.4 Планируемые результаты

Личностные результаты:

Обучающийся будет готов и способен к саморазвитию и личностному самоопределению;

У обучающегося будет сформирована мотивация к обучению и целенаправленной познавательной деятельности.

Метапредметные результаты:

Обучающийся будет знать, как ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные), делать выводы в процессе работы и по её окончании, корректировать намеченный план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности.

Обучающийся приобретёт умение оценивать правильность выполнения учебной задачи, корректировать свои действия в соответствии с изменяющейся ситуацией, определять способы действий в рамках

предложенных условий.

Предметные результаты:

1. Обучающийся будет владеть теоретической базой и практическими навыками программирования на языке Kotlin;
2. Обучающийся будет владеть навыками разработки, тестирования и отладки несложных программ;
3. Обучающийся будет уметь развивать логические, аналитические и творческие способности.

РАЗДЕЛ № 2. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ

1.1 Условия реализации программы

1. Материально-техническое обеспечение:

Характеристика помещения для занятий по программе – учебное помещение образовательного учреждения, оснащенное наглядными пособиями, учебным оборудованием, мебелью и техническими средствами обучения, в котором проводится учебная, индивидуальная и внеклассная работа с обучающимися в соответствии с действующими государственными образовательными стандартами, учебным планом и образовательными программами.

Для реализации данного курса требуется следующее оборудование:

- Проектор и экран для демонстрации учебного материала
- Доска
- Персональные компьютеры для обучающихся
- Раздаточные материалы
- Наушники с микрофоном

Требуемое программное обеспечение:

- Пакет офисных приложений;
- Программное обеспечение Visual Studio Code, IntelliJ IDEA;
- Браузер Google Chrome, Mozilla Firefox.

2. Учебно-методическое и информационное обеспечение:

1. <https://kotlinlang.org/docs/reference/> – официальный сайт со справочником по языку Kotlin (на английском языке);
2. Жемеров Д. Kotlin в действии [Электронный ресурс], 2020;
3. Дон Гриффитс Head First. Kotlin [Электронный ресурс] / Гриффитс Дон, Гриффитс Дэвид, 2021;
4. Сомон. П. И. Волшебство Kotlin [Электронный ресурс]: руководство / П.И. Сомон., 2020.

2.2 Оценочные материалы и формы аттестации

В процессе изучения программы для оценки текущей работы используются методы устного, письменного, практического, машинного контроля и самоконтроля. Педагогический контроль осуществляется в несколько этапов и включает в себя несколько уровней.

Виды и формы контроля знаний и навыков учащихся:

- Текущий контроль осуществляется регулярно, посредством проведения лабораторных занятий, заключается в ответе учащихся на контрольные вопросы, фронтальных опросах учителем, а также в демонстрации полученных скриптов в среде разработки (опрос, лабораторные работы).
- Тематический контроль — определение результатов обучения за определенный раздел программы, проводится посредством выполнения учащимися контрольных работ (контрольные работы, тестовые занятия).
- Итоговый контроль — проводится по окончании обучения по программе, он предполагает комплексную проверку образовательных результатов по всем ключевым целям и направлениям (разработка учащимися индивидуальных/групповых проектов и их защита).

Формы контроля результатов (промежуточная аттестация):

- Беседа – это диалогический метод обучения, при котором учитель путем постановки тщательно продуманной системы вопросов подводит учеников к пониманию нового материала или проверяет усвоение ими уже изученного. (применяется во время занятия «Тема 1: Введение. Знакомство и ознакомление с ТБ»);

- Устный опрос – позволяет выявить правильность ответа по содержанию, его последовательность, самостоятельность суждений и выводов, степень развития логического мышления, культуру речи учащихся. Эта форма применяется для текущего и тематического учета, а также для отработки и развития экспериментальных умений учащихся. Причем устную проверку считают эффективной, если она направлена на выявление осмысленности восприятия знаний и осознанности их использования, если она стимулирует самостоятельность и творческую активность учащихся. (применяется во время занятий: “Тема: 2-3”, “Тема: 10”);

- Тестирование - представляет собой кратковременное технически сравнительно просто составленное испытание, проводимое в равных для всех испытуемых условиях и имеющее вид такого задания, решение которого поддается качественному учету и служит показателем степени развития к данному моменту известной функции у данного испытуемого. (применяется во время занятий по темам №4, 6, 8, 11);

- Зачёт – проводится для определения достижения конечных результатов обучения по определенной теме каждым учащимся. Перед началом изучения материала, учащиеся знакомятся с перечнем вопросов и обязательных задач по теме, а также дополнительными вопросами и задачами. Иногда целесообразны закрытые зачеты, когда учащиеся получают вопросы и задания непосредственно во время проведения зачета. Его достоинство заключается в том, что он предполагает комплексную проверку всех знаний и умений учащихся. (применяется во время занятий по темам №5, 7, 9, 12);

- Контрольная работа – проводится с целью определения конечного результата в обучении по данной теме или разделу, контролировать знания

одного и того же материала неоднократно. Целесообразно проводить контрольные работы различного вида. С помощью промежуточной контрольной работы Преподаватель проверяет усвоение учащимися материала в период изучения темы. Итоговая контрольная работа проводится с целью проверки знаний и умений учащихся по отдельной теме, курсу. (применяется во время занятия “Тема 13: Основы программирования на Kotlin.”).

Формы фиксации результатов:

- Ведомости результатов аттестации учащихся. (применяются после завершения курса для фиксации итоговых результатов);

- Бланки тестовых заданий и контрольных работ по темам программы. (применяются на протяжении всего периода обучения. Тема 1 – Тема 13);

- Разработанные, в ходе проведения занятий, программы в виде файлов. (применяются на протяжении всего периода обучения. Тема 1 – Тема 13);

- Журнал посещаемости. (применяется на протяжении всего периода обучения. Тема 1 – Тема 13);

- Отзывы детей и родителей. (собираются после завершения курса для оперативной обратной связи по поводу предложений и замечаний);

- Свидетельство (сертификат). Выдается участнику после успешного завершения учебной программы.

2.3 Методические материалы

Дидактические и методические материалы: раздел «Объектно-ориентированное программирование в Kotlin» дидактических материалов пособия.

На рисунке 1 изображен фрагмент дидактического материала по теме «Объектно-ориентированное программирование в Kotlin».

В Kotlin классы используются для определения объектов, которые могут содержать свойства и функции. Объекты классов могут использоваться для создания экземпляров, которые имеют своё собственное состояние и поведение.

Объявление класса:

```
class MyClass {  
    // свойства (поля) класса  
    var property1: Int = 0  
    var property2: String = ""  
  
    // функции (методы) класса  
    fun myFunction() {  
        println("Вызвана функция myFunction()")  
    }  
}
```

Создание объекта класса:

```
val myObject = MyClass()
```

В Kotlin также существует возможность создания объектов без явного объявления класса с помощью конструкции `object`. Это так называемые объекты-одиночки (Singleton), которые могут использоваться для хранения глобальных данных или реализации шаблона проектирования Singleton.

Объявление объекта-одиночки:

```
object MySingleton {  
    // свойства и функции объекта-одиночки  
}
```

Пример использования:

```
MySingleton.myFunction()
```

Рисунок 1 – Фрагмент дидактических материалов по Теме №11

Дидактические и методические материалы: раздел «Ветвление в языке Kotlin.» дидактических материалов пособия.

На рисунке 2 изображен фрагмент дидактического материала по теме «Ветвление в языке Kotlin».

Условное выражение `if...else` позволяет выполнять определенные блоки кода в зависимости от условия.

```
val x = 10
if (x > 5) {
    println("x больше 5")
} else {
    println("x меньше или равен 5")
}
```

В этом примере, если значение переменной `x` больше 5, то будет выведено сообщение `"x больше 5"`, иначе будет выведено `"x меньше или равен 5"`.

Вы также можете использовать несколько условий с помощью конструкции `if...else if...else`:

```
val num = 0
if (num > 0) {
    println("Число положительное")
} else if (num < 0) {
    println("Число отрицательное")
} else {
    println("Число равно нулю")
}
```

Этот код проверяет, является ли число положительным, отрицательным или равным нулю, и выводит соответствующее сообщение.

Рисунок 2 – Фрагмент дидактических материалов по Теме №4

Дидактические и методические материалы: раздел «Циклы в языке Kotlin» дидактических материалов пособия.

На рисунке 3 изображен фрагмент дидактического материала по теме «Циклы в языке Kotlin»

Цикл `do-while` в Kotlin очень похож на цикл `while`, но с одним отличием: в цикле `do-while` блок кода выполняется хотя бы один раз, даже если условие не выполняется. То есть проверка на условие происходит после блока кода.

Вот пример использования цикла `do-while`:

```
var i = 0
do {
    println(i)
    i++
} while (i < 5)
```

Этот пример кода будет выводить числа от 0 до 4 на экран, даже если условие `i < 5` не выполняется с самого начала.

Основные компоненты цикла `do-while`:

1. **Блок кода:** Блок кода внутри цикла `do-while` определяет действия, которые должны выполняться на каждой итерации.
2. **Условие:** Условие в `while` определяет, когда цикл должен продолжаться. После выполнения блока кода внутри `do`, программа проверяет условие, и если оно истинно, цикл повторяется.

Цикл `do-while` полезен, когда вы хотите, чтобы определенный блок кода выполнялся хотя бы один раз, независимо от того, выполнено ли условие или нет. Это может быть полезно, например, при вводе данных с клавиатуры, когда вы хотите, чтобы пользователю предоставилась возможность ввести данные, прежде чем программа продолжит выполнение.

Рисунок 3 – Фрагмент дидактических материалов по Теме №6

Пример шаблона формы для фиксации результатов индивидуальных проектов:

Лист оценивания проекта

Критерии оценивания	1-я группа	2-я группа	...
Актуальность темы			
Соответствие содержания проекта заявленной теме			
Техническая сложность			
Оригинальность			
Дизайн			
Наличие соответствующего <u>музыкального</u> сопровождения с указанием в титрах <u>авторов музыки</u>			
Уровень проработанности проекта			
Возможность применения проекта в школе			
Итоговое количество баллов			

2.4 Календарный учебный график

Вариант календарного учебного графика, когда программа размещается на сайте:

Этапы образовательного процесса		1 год
Продолжительность учебного года, неделя		36
Количество учебных дней		36
Продолжительность учебных периодов	1 полугодие	01.10.2024- 31.12.2024
	2 полугодие	10.01.2025- 31.06.2025
Возраст детей, лет		14-17
Продолжительность занятия, час		2
Режим занятия		1 раз / нед.
Годовая учебная нагрузка, час		72

2.5 Календарный план воспитательной работы

Для выстраивания адресной образовательной деятельности необходимо хорошо понимать логистическую и содержательную структуру работы, включающей обучение и воспитание учащихся.

Рабочая программа воспитательной работы Центра едина для всех творческих объединений и создает единую воспитательную среду учреждения, которая объединяет всех участников образовательного процесса. Практическая реализация цели и задач воспитания учащихся осуществляется в рамках направлений воспитательной работы образовательной организации, представленных в соответствующем модуле.

При составлении календарного плана воспитательной работы творческого объединения было адаптировано содержание модулей календарного плана программы для работы с учащимися творческого объединения по приоритетным направлениям различного уровня с учетом

конкретных условий и особенностей деятельности.

Вызовы времени заставили снова активно заговорить о воспитании личности. С 1 сентября 2020 года вступил в силу Федеральный закон от 31 июля 2020 г. № 304-ФЗ «О внесении изменений в Федеральный закон «Об образовании в Российской Федерации» по вопросам воспитания учащихся». Политика государства в сфере образования определяет воспитание как первостепенный приоритет в образовании, а в качестве важнейших задач выдвигает формирование гражданской ответственности, правового самосознания, духовности и культуры, инициативности, самостоятельности, толерантности, способности к успешной социализации в обществе.

В Федеральном законе акцентируется внимание на том, что система образования не только учит, но и воспитывает, формирует личность, передает ценности и традиции, на которых основано общество, что смысл предлагаемых поправок — «укрепить, акцентировать воспитательную составляющую отечественной образовательной системы».

В соответствии с Федеральным законом № 304-ФЗ вводится механизм организации воспитательной работы (программа воспитания), который является частью общеобразовательной (общеразвивающей) программы педагога.

СПИСОК ЛИТЕРАТУРЫ

Ссылки на печатные источники:

1. Скин Д., Гринхол Д. Kotlin. Программирование для профессионалов / пер. с англ. Киселев А.Н. СПб: Издательский дом «Питер», 2020;
2. Филлипс Б., Стюарт К., Марсикано К. Android. Программирование для профессионалов. 3-е издание / пер. с англ. СПб: Издательский дом «Питер», 2020;
3. Сомон. П. И. Волшебство Kotlin [Электронный ресурс]: руководство / П.И. Сомон., 2020.