**Паспорт Образовательной программы**

**«Основы программирования на языке Kotlin (базовый курс)»**

|  |  |
| --- | --- |
| **Версия программы** | **1** |
| **Дата Версии** | **14.10.2020** |

1. **Сведения о Провайдере**

|  |  |  |
| --- | --- | --- |
| 1.1 | Провайдер | **АНО ДПО "СофтЛайн Эдюкейшн"** |
| 1.2 | Логотип образовательной организации |  |
| 1.3 | Провайдер ИНН | **7736228783** |
| 1.4 | Ответственный за программу ФИО | **Максименко Денис Владиславович** |
| 1.5 | Ответственный должность | Руководитель проектного офиса |
| 1.6 | Ответственный Телефон | +7-495-232-0023 доб. 0889 |
| 1.7 | Ответственный Е-mail | Edu2035@softline.com |

1. **Основные Данные**

|  |  |  |
| --- | --- | --- |
| **№** | **Название** | **Описание** |
| 2.1 | Название программы | **Основы программирования на языке Kotlin (базовый курс)** |
| 2.2 | Ссылка на страницу программы | <https://go.teachbase.ru/course_sessions/vhodnoe-testirovanie-kurs-osnovy-programmirovaniya-na-yazyke-kotlin-bazovyy-kurs-11-01/apply> - входное тестирование  <https://go.teachbase.ru/course_sessions/osnovy-programmirovaniya-na-yazyke-kotlin-bazovyy-kurs-11-01/apply> - страница программы |
| 2.3 | Формат обучения | Заочное с использованием дистанционных образовательных технологий |
|  | Подтверждение от ОО наличия возможности реализации образовательной программы с применением электронного обучения и (или) дистанционных образовательных технологий с возможностью передачи данных в форме элементов цифрового следа | Подтверждаем возможность реализации образовательной программы с применением дистанционных образовательных технологий с возможностью передачи данных в форме элементов цифрового следа |
| 2.4 | Уровень сложности | Базовый |
| 2.5 | Количество академических часов | 72 академических часа |
|  | Практикоориентированный характер образовательной программы: не менее 50 % трудоёмкости учебной деятельности отведено практическим занятиям и (или) выполнению практических заданий в режиме самостоятельной работы | 57 ак. часов отведено практическим занятиям и выполнению практических заданий в режиме самостоятельной работы, что составляет около 79,2 % трудоёмкости учебной деятельности |
| 2.6 | Стоимость обучения одного обучающегося по образовательной программе, а также предоставление ссылок на 3 (три) аналогичные образовательные программы иных организаций, осуществляющих обучение, для оценки объективности стоимости или обоснование уникальности представленной образовательной программы в случае отсутствия аналогичных образовательных программ на рынке образовательных услуг | 29900 руб.   1. <https://www.specialist.ru/course/kotlin>   Стоимость: 22 450 (32 ак. часа)   1. <https://beonmax.com/courses/kotlin/?pref=18584>   Стоимость: 2769 ( 5 ак. часов)   1. <http://levelp.ru/courses/mobilnye-tekhnologii/android/>   Стоимость: 36 990 (72 ак часа)   1. <https://y-skills.com/backend_razrabotka_na_kotlin_otus-1381>   Стоимость: 35 000 |
| 2.7 | Минимальное количество человек на курсе | 1 |
| 2.8 | Максимальное количество человек на курсе | 1000 |
| 2.9 | Данные о количестве слушателей, ранее успешно прошедших обучение по образовательной программе | отсутствует |
| 2.10 | Формы аттестации | тестирование |
|  | Указание на область реализации компетенций цифровой экономики, к которой в большей степени относится образовательная программа, в соответствии с Перечнем областей | Программирование и создание ИТ-продуктов |

1. **Аннотация программы**

Kotlin — молодой и относительно несложный для изучения язык программирования, позволяющий создавать программы под платформы JVM и Android. Разработан петербургской компанией JetBrains. Так как Kotlin и Java — полностью интероперабельные языки, то это позволяет их использовать в смешанных проектах, создаваемых на нескольких языках программирования. Программы на Kotlin могут использовать все имеющиеся Java-библиотеки, и наоборот. В настоящее время программы на Kotlin пишут сотни тысяч программистов, основная ниша его промышленного применения — мобильные приложения под платформу Android и, в несколько меньшей степени, web-разработка. С 2017 года Google выбрал именно Kotlin в качестве рекомендуемого и основного языка для платформы Android. Все это подтверждает актуальность программы повышения квалификации. Программа повышения квалификации направлена на совершенствование профессиональной компетенции «Способен ставить и решать прикладные задачи по созданию и совершенствованию программного кода с использованием языка программирования Kotlin». Для успешного освоения программы обучающимся необходимо владеть базовыми компетенциями цифровой грамотности, в том числе навыками владения ПК на уровне среднего или продвинутого пользователя; компетенции в области алгоритмизации (в том числе знаниями и умениями в области основ алгоритмизации (знание основных алгоритмических структур, умения выражать их на псевдокоде и в виде блок-схем) и основ программирования, базовыми знаниями в области языка Java. В процессе обучения по программе обучающимся рекомендуется прослушать все видеолекции и видеоролики по практическим занятиям, выполнить по каждой теме обязательные практические задания с последующей проверкой преподавателем, проработать самостоятельно рекомендуемые дополнительные материалы (преимущественно документацию по Kotlin) и выполнить задания по самостоятельной работе, после этого необходимо выполнить тесты по каждой теме и для успешного завершения обучения выполнить итоговый тест (70% верных ответов). Обучающийся в результате освоения программы: *будут знать* синтаксис языка программирования Kotlin и особенности программирования на Kotlin, особенности выбранной среды программирования, место языка программирования Kotlin среди развивающихся языков программирования, технологии программирования на языке Kotlin, особенности Android-разработки; *будут уметь* применять Kotlin для написания программного кода, использовать инструментальные средства разработки, доступные у платформы IntelliJ IDEA, осуществлять подготовку данных для реализации работы над приложением; *будут владеть* приемами создания программного кода на Kotlin в соответствии с спецификациями, приемами и методами реализации базовых алгоритмических конструкции на языке программирования Kotlin, способами определения и манипулирования данными на языках Java и Kotlin, приемами работы в интегрированной среде разработки, позволяющей реализовывать разработку проекта, приемами запуска и отладки программного кода в интегрированной среде разработки. Результаты обучения по программе будут востребованы при решении следующих профессиональных задач: создание программного кода на Kotlin в соответствии с спецификациями и его совершенствование с использованием специализированных программных средств; размещение программного кода в репозиториях; разработка, отладка проектов для мобильных приложений под Android.

**Автономная некоммерческая организация**

**дополнительного профессионального образования**

**«СофтЛайн Эдюкейшн»**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ:  Генеральный директор  АНО ДПО «СофтЛайн Эдюкейшн»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.Э. Разуваев  «\_14\_» \_\_Октября\_\_\_ 2020\_ г. |

**Дополнительная профессиональная программа повышения квалификации**

**«Основы программирования на языке Kotlin (базовый курс)»**

72 часа

Москва, 2020

**ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ**

**1.Цель программы**

Формирование профессиональных компетенций, необходимых для решения прикладных задач по созданию и совершенствованию программного кода с использованием специализированных программных средств (IntelliJ IDEA) на языке Kotlin.

**2.Планируемые результаты обучения:**

2.1.Знание (осведомленность в областях)

2.1.1 синтаксиса языка программирования Kotlin и особенности программирования на Kotlin;

2.1.2 особенности выбранной среды программирования;

2.1.3 место языка программирования Kotlin среди развивающихся языков программирования;

2.1.4 технологии программирования на языке Kotlin;

2.1.5 особенности Android-разработки;

2.2. Умение (способность к деятельности)

2.2.1 применять Kotlin для написания программного кода;

2.2.2 использовать инструментальные средства разработки, доступные у платформы IntelliJ IDEA;

2.2.3 осуществлять подготовку данных для реализации работы над приложением;

2.3.Навыки (использование конкретных инструментов)

2.3.1 создания программного кода на Kotlin в соответствии с спецификациями;

2.3.2 реализации базовых алгоритмических конструкции на языке программирования Kotlin;

2.3.3 определения и манипулирования данными на языках Java и Kotlin;

2.3.4 работы в интегрированной среде разработки, позволяющей реализовывать разработку проекта;

2.3.5 запуска и отладки программного кода в интегрированной среде разработки.

**3.Категория слушателей** (возможно заполнение не всех полей)

* 1. Образование: среднее профессиональное, высшее
  2. Квалификация: уровень квалификации не ниже 4
  3. Наличие опыта профессиональной деятельности: опыт разработки программного обеспечения, электронных образовательных ресурсов и информационных систем
  4. Предварительное освоение иных дисциплин/курсов /модулей

**4.Учебный план программы «Основы программирования на языке Kotlin (базовый курс)»**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Модуль** | **Всего, час** | **Виды учебных занятий** | | | |
| **лекции** | **практические занятия** | **самостоятельная работа** | |
|  | Модуль 1. Вводное занятие. | **3** | 1 | 1 | 1 | |
|  | Модули 2. Основы синтаксиса языка Kotlin | **10** | 2 | 6 | 2 | |
|  | Модуль 3. Тип nullable и non-nullable | **5** | 1 | 3 | 1 | |
|  | Модуль 4. Циклы | **5** | 1 | 3 | 1 | |
|  | Модуль 5. Массивы и коллекции | **5** | 1 | 3 | 1 | |
|  | Модуль 6. Функциональное программирование | **16** | 4 | 9 | 3 | |
|  | Модули 7. Объектно-ориентированное программирование | **20** | 3 | 9 | 8 | |
|  | Модули 8. Введение в Android-разработку | **7** | 2 | 3 | 2 | |
| **Итоговая аттестация** | | **1** | **Тест, зачет** | | | **1** |
| **ИТОГО:** | | **72** | **15** | **37** | | **20** |

**5.Календарный план-график реализации образовательной программы** **«Основы программирования на языке Kotlin (базовый курс)»**

(дата начала обучения – дата завершения обучения) в текущем календарном году, указания на периодичность набора групп (не менее 1 группы в месяц)

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Наименование учебных модулей** | **Трудоёмкость (час)** | **Сроки обучения** |
| **1** | Модуль 1. Вводное занятие. | **3** | **01.11** |
| **2** | Модули 2. Основы синтаксиса языка Kotlin | **10** | **01.11-02.11** |
| **3** | Модуль 3. Тип nullable и non-nullable | **5** | **03.11** |
| **4** | Модуль 4. Циклы | **5** | **04.11** |
| **5** | Модуль 5. Массивы и коллекции | **5** | **05.11** |
| **6** | Модуль 6. Функциональное программирование | **16** | **06.11-09.11** |
| **7** | Модули 7. Объектно-ориентированное программирование | **20** | **10.11-13.11** |
| **8** | Модули 8. Введение в Android-разработку | **7** | **14.11** |
| **9** | Итоговая аттестация | **1** | **15.11** |
| **Всего:** | | **72** | **15 календарных дней** |

**6.Учебно-тематический план программы «Основы программирования на языке Kotlin (базовый курс)»**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Модуль / Тема** | **Всего, час** | **Виды учебных занятий** | | | **Формы контроля** |
| **лекции** | **практические занятия** | **самостоятельная работа** |
| 1 | **Модуль 1. Вводное занятие** | **3** | 1 | 1 | 1 |  |
| 1.1 | Тема 1.1. Введение в язык Kotlin | 3 | 1 | 1 | 1 | тест |
| 2 | **Модуль 2. Основы синтаксиса языка Kotlin** | **10** | 2 | 6 | 2 |  |
| 2.1 | Тема 2.1. Функции и переменные | 4 | 1 | 2 | 1 | тест |
| 2.2 | Тема 2.1. Типы данных, выражения и конструкции | 6 | 1 | 4 | 1 | тест |
| 3 | **Модуль 3. Тип nullable и non-nullable** | **5** | **1** | **3** | **1** |  |
| 3.1 | Тема 3.1. Тип nullable и non-nullable | 5 | 1 | 3 | 1 | тест |
| 4 | **Модуль 4. Циклы** | **5** | **1** | **3** | **1** |  |
| 4.1 | Тема 4.1. Циклы | 5 | 1 | 3 | 1 | тест |
| 5 | **Модуль 5. Массивы и коллекции** | **5** | 1 | 3 | 1 |  |
| 5.1 | Тема 5.1 Массивы и коллекции | 5 | 1 | 3 | 1 | тест |
| 6 | **Модуль 6. Функциональное программирование** | **16** | 4 | 9 | 3 |  |
| 6.1 | Тема 6.1 Введение в функциональное программирование | 5 | 1 | 3 | 1 | тест |
| 6.2 | Тема 6.2 Функции | 5 | 1 | 3 | 1 | тест |
| 6.3 | Тема 6.3 Функциональные выражения | 6 | 2 | 3 | 1 | тест |
| 7 | **Модуль 7. Объектно-ориентированное программирование** | **20** | 3 | 9 | 8 |  |
| 7.1 | Тема 7.1. Введение в объектно-ориентированное программирование | 4 | 1 | 3 | - | тест |
| 7.2 | Тема 7.2. Конструкторы объектов | 7 | 1 | 3 | 3 | тест |
| 7.3 | Тема 3.3. Классы и их типы | 9 | 1 | 3 | 5 | тест |
| 8 | **Модуль 8. Введение в Android-разработку** | **7** | 2 | 3 | 2 |  |
| 8.1 | Тема 8.1 Введение в Android-разработку | 7 | 2 | 3 | 2 | тест |
|  | **Итоговая аттестация** | 1 |  |  | 1 | **Итоговый тест** |

**7. Учебная (рабочая) программа повышения квалификации «Основы программирования на языке Kotlin (базовый курс)»**

**Модуль 1. Вводное занятие (3 час.)**

Тема 1.1. Введение в язык Kotlin (3 час.)

Описание языка Kotlin. Установка IntelliJ IDEA. Первая программа на Kotlin.

**Модуль 2. Основы синтаксиса языка Kotlin (10 ч.)**

Тема 2.1. Функции и переменные (4 час.)

Функции. Переменные. Именование переменных

Тема 2.2. Типы данных, выражения и конструкции (6 час.)

Типы данных. Выражение if. Конструкция when.

**Модуль 3. Тип nullable и non-nullable (5 час.)**

Тема 3.1. Тип nullable и non-nullable (5 час.)

Тип Nullable. Оператор Элвис

**Модуль 4. Циклы (5 час.)**

Тема 4.1. Циклы (5 час.)

Циклы while, do-while. Цикл for.

**Модуль 5. Массивы и коллекции (5 час.)**

Тема 5.1 Массивы. Работа с массивом. Введение в коллекции. Работа с коллекцией. (5 час.)

**Модуль 6. Функциональное программирование (16час.)**

Тема 6.1 Введение в функциональное программирование (5 час.)

Параметры по умолчанию. Именованные параметры. Аргументы переменной длины. Исключения.

Тема 6.2 Функции (5 час.)

Локальные функции. Лямбда-выражения

Тема 6.3 Функциональные выражения (6 час.)

Ключевое слово it. Extension-функции. Ключевое слово infix

**Модуль 7. Объектно-ориентированное программирование (20 час.)**

Тема 7.1. Введение в объектно-ориентированное программирование (4 час.)

Классы. Объявление классов. Свойства класса. Геттеры и сеттеры

Тема 7.2 Конструкторы объектов (7 час.)

Основной конструктор. Вторичный конструктор. Методы в классах. Extension-функции в классах

Тема 7.3 Классы и типы (9 час.)

Модификаторы доступа. Вложенные и внутренние классы. Интерфейсы. Наследование. Data-классы. Enum-классы

**Модуль 8. Введение в Android-разработку (7 час.)**

Тема 8.1 Введение в Android-разработку (7 час.)

Установка Android Studio. Создание первого проекта. Файл Build Gradle. Создание второй Activity. Интерфейс Android Studio

**Итоговая аттестация (1 час.)**

**Описание практико-ориентированных заданий и кейсов**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Номер темы/модуля** | **Наименование практического занятия** | **Описание** |
| **1** | **Модуль 1. Тема 1.1** | Вывести результат на консоль | *Видеоролик с демонстрацией примеров практических заданий и рекомендаций по выполнению обязательных заданий. Самостоятельное выполнение обязательных практических заданий с последующей проверкой преподавателем*  Практические задания на отработку умения выводить результат на консоль.  *Пример задания.* Создайте программу, которая выведет на экран следующие значения:  “Меня зовут $ВашеИмя”  1598520  true  false  45+18  Каждое значение выводить с новой строки.  Для вывода не используйте переменные. |
| **2** | **Модуль 2. Тема 2.1** | 1. Передача параметров | *Видеоролик с демонстрацией примеров практических заданий и рекомендаций по выполнению обязательных заданий. Самостоятельное выполнение обязательных практических заданий с последующей проверкой преподавателем*  Практические задания на отработку умения создавать функцию.  *Пример задания.* Создайте функцию c тремя параметрами. Функция должна возвращать результат умножения чисел (числа произвольные). |
| **3** | **Модуль 2. Тема 2.2** | 1. Переменные  2. Сравнение чисел  3. Вывести четное число  4. Конструкция when | *Видеоролик с демонстрацией примеров практических заданий и рекомендаций по выполнению обязательных заданий. Самостоятельное выполнение обязательных практических заданий с последующей проверкой преподавателем* Практические задания на отработку знаний базового синтаксиса языка.  *Пример задания.* Попросите пользователя ввести имя, а также возраст.  Выведите полученные данные в формате: «Привет, $ИМЯ! Вам $ВОЗРАСТ».  Используйте класс Scanner для получения данных, введенных с клавиатуры.  *Пример задания.* Создайте две переменных со значением 10 и 20. Создайте функцию, которая сравнивает числа пользователя и возвращает результат в виде знаков >, < или =.  *Пример задания.* Из двух чисел с разной четностью вывести на экран четное число.  a, b - данные числа  Используйте выражение if.  *Пример задания.* Введите число от 1 до 7, которое будет соответствовать дню недели.  Например: при вводе числа 1 на экране будет выводиться «Понедельник» и т. д.  Сделайте подобную проверку при помощи оператора when. |
| **4** | **Модуль 3. Тема 3.1** | 1.Проверка на null | *Видеоролик с демонстрацией примеров практических заданий и рекомендаций по выполнению обязательных заданий. Самостоятельное выполнение обязательных практических заданий с последующей проверкой преподавателем* Практические задания на отработку умения проверять значение на null.  *Пример задания*  Вам необходимо будет создать функцию, которая будет принимать nullable строку и возвращать её значение. Если данная строка не null и не пустая, то необходимо будет вернуть информацию о её длине. В противном случае, если она null, то вывод должен содержать в себе информацию о том, что строка пустая или null. |
| **5** | **Модуль 4. Тема 4.1** | 1. Цикл for  2. Найти символ в диапазоне | *Видеоролик с демонстрацией примеров практических заданий и рекомендаций по выполнению обязательных заданий. Самостоятельное выполнение обязательных практических заданий с последующей проверкой преподавателем* Практические задания на отработку умения работать с циклами.  *Пример задания*. С помощью цикла for и оператора in получить следующий результат.  a  b  c  d  e  f  *Пример задания.* Создать функцию, объявить в качестве параметра - переменную и передать ей тип **Char**, проверить является ли символ буквой. Использовать операторы **in** и **!in**. Если символ будет являться буквой, то возвращается **true**, если будет цифра или специальные символы, то будет возвращаться **false**. |
| **6** | **Модуль 5. Тема 5.1** | 1. Определение массива  2. Сортировка массива | *Видеоролик с демонстрацией примеров практических заданий и рекомендаций по выполнению обязательных заданий. Самостоятельное выполнение обязательных практических заданий с последующей проверкой преподавателем*  Практические задания на отработку умения работать с массивами и коллекциями.  *Пример задания.* Напишите программу для определения массива createArray из количества элементов, которые выберет сам пользователь. Попросите пользователя заполнить массив произвольными числами. После заполнения, выведите массив на экран.  *Пример задания.* Создайте программу, которая будет сортировать массив в порядке возрастания элементов. В конце программы выведите отсортированный массив на экран. Числа: (5, -15, 100, -55, 20, -5, -80, 45, 25, -10).  Подсказка (для вывода отсортированного массива используйте цикл foreach):  arr.*forEach* **{** element **->**  *println*(element)  **}** |
| **7** | **Модуль 6. Темы 6.1** | 1. Параметры по умолчанию  2. Аргументы переменной длины vararg | *Видеоролик с демонстрацией примеров практических заданий и рекомендаций по выполнению обязательных заданий. Самостоятельное выполнение обязательных практических заданий с последующей проверкой преподавателем* Практические задания на отработку умения работать с функциями.  *Пример задания*  Создать функцию, которая будет находить наименьшее число из двух целочисленных значений. В качестве параметров объявить 2 целочисленных параметра типа **Int**, и специальный параметр, который будет определять нужно ли ему сравнивать числа, значение по умолчанию у данного параметра будет **false**.  *Пример задания*  Установить значение префикса к числам, у вас должен получится следующий вывод, где префиксом будет number:  number: one  number: two  number: three  number: four  number: five |
| **8** | **Модуль 6. Тема 6.2** | 1. Найти произведение чисел  2. Найти сумму чисел  3. Найти площадь периметра | *Видеоролик с демонстрацией примеров практических заданий и рекомендаций по выполнению обязательных заданий. Самостоятельное выполнение обязательных практических заданий с последующей проверкой преподавателем* Практические задания на отработку умения работать с функциями  *Пример задания*  Напишите функцию с именем mult(), которая будет принимать три числа и возвращать произведение этих чисел.  *Пример задания*  Напишите функцию с именем sum(), которая будет принимать три числа и возвращать true, если сумма первых двух чисел равно третьему числу. В противном случае возвращается false.  *Пример задания*  Создать лямбда-выражения и сохранить их в переменные, а затем вызвать их:  Принимает две стороны прямоугольника в качестве параметра и возвращает площадь периметра. |
| **9** | **Модуль 6. Тема 6.3** | 1. Ключевое слово it  2. Extension-функции | *Видеоролик с демонстрацией примеров практических заданий и рекомендаций по выполнению обязательных заданий. Самостоятельное выполнение обязательных практических заданий с последующей проверкой преподавателем*  Практические задания на отработку умений работать с лямбда-функциями и функциями-расширения.  *Пример задания*  Функция в качестве параметра принимает имя, выводит на экран строку: “Привет, [имя]!” и ничего не возвращает.  *Пример задания*  Создайте функцию-расширение типа Int, которое возвращает значение - является ли число простым (простое число делится только на себя и на единицу) |
| **10** | **Модуль 7. Тема 7.1** | 1. Класс Попугай | *Видеоролик с демонстрацией примеров практических заданий и рекомендаций по выполнению обязательных заданий. Самостоятельное выполнение обязательных практических заданий с последующей проверкой преподавателем*  Практические задания на отработку умения работать с классами и объектами  *Пример задания*  Создайте класс - Попугай, который содержит свойства - кличка и возраст. Переопределите геттеры и сеттеры таким образом, чтобы возраст нельзя было установить меньше нуля, при получении клички попугая нам возвращалась строка, начинающаяся с заглавной буквы. |
| **11** | **Модуль 7. Тема 7.2** | 1. Класс самолет  2. Класс машина | *Видеоролик с демонстрацией примеров практических заданий и рекомендаций по выполнению обязательных заданий. Самостоятельное выполнение обязательных практических заданий с последующей проверкой преподавателем*  Практические задания на отработку умения осуществлять работу с классами и объектами  *Пример задания*  Создайте класс - Самолет. Добавьте конструктор, который содержит следующие параметры - количество мест в салоне, год выпуска и модель самолета. При создании объекта есть два обязательных параметра - это количество мест в салоне и год выпуска самолета, параметр модели самолета по умолчанию null. После создания класса Самолета, невозможно изменить изменить год выпуска, остальные поля изменяемые.  *Пример задания*  Создайте класс - Машина. Конструктор которого будет содержать следующие обязательные поля: тип транспорта (легковые, грузовые, спецтехника, автобусы, мотоциклы и т.д.), название марки, и год выпуска. И в функции main создайте extension функцию, которая будет выводить всю информацию о машине. |
| **12** | **Модуль 7. Тема 7.3** | 1. Классы | *Видеоролик с демонстрацией примеров практических заданий и рекомендаций по выполнению обязательных заданий. Самостоятельное выполнение обязательных практических заданий с последующей проверкой преподавателем*  Практические задания на отработку умения создавать сложные классы  *Пример задания*   1. Создайте класс-наследник «Airbus» <https://ru.wikipedia.org/wiki/Airbus> со значениями:  * поля: количество моделей самолетов; * конструктор для установки всех переменных в том числе переменных что находятся в родительском классе; * метод для вывода всех переменных из данного класса и класса родителя.   2. Создайте еще один класс-наследник «Boeing» <https://ru.wikipedia.org/wiki/Boeing> со значениями:   * поля: является ли конкурентом; * конструктор для установки всех переменных в том числе переменных что находятся в родительском классе; * метод для вывода всех переменных из данного класса и класса родителя. В методе создайте проверку на поле «является ли конкурентом». Выводить слово «Да» в случае если является конкурентом и слово «Нет», если значение этой переменной false. [https://ru.wikipedia.org/wiki/Конкуренция\_между\_Airbus\_и\_Boeing](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D0%BA%D1%83%D1%80%D0%B5%D0%BD%D1%86%D0%B8%D1%8F_%D0%BC%D0%B5%D0%B6%D0%B4%D1%83_Airbus_%D0%B8_Boeing) |
| **13** | **Модуль 8. Тема 8.1** | Создать проект Send Button | *Видеоролик с демонстрацией примеров практических заданий и рекомендаций по выполнению обязательных заданий. Самостоятельное выполнение обязательных практических заданий с последующей проверкой преподавателем*  Практические задания на отработку умения создавать проекты в Android Studio.  *Пример задания*  Создать проект Send Button, в котором нужно добавить кнопку button с именем “Click Me” и с помощью нее перейти на новый экран. |

**8.Оценочные материалы по образовательной программе** **«Основы программирования на языке Kotlin (базовый курс)»**

**8.1. Вопросы тестирования по модулям**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ модуля** | **Вопросы входного тестирования** | **Вопросы промежуточного тестирования** | **Вопросы итогового тестирования** |
| **0** | 1. Что такое Котлин?  * Это прагматичный язык, предназначенный для решения реальных проблем * Это обучающий язык программирования, предназначенный для обучения студентов * Это активно развивающийся язык, но еще не готовый к промышленному использованию * Это академический язык, предназначенный для изучения инновационных идей в области компьютерных наук  1. Минусы языка Kotlin. Выберите несколько вариантов из списка:  * Статическая типизация * Неочевидность ряда конструкций * Отсутствие устоявшегося code style * Си-образный синтаксис * Наличие лямбда выражений * Увеличение количества методов в приложении  1. Выберите правильное утверждение о совместимости с Java.  * Котлин не совместим с Java * Kotlin и Java имеют ограниченную совместимость * Kotlin прекрасно работает со всем существующим Java-кодом и библиотеками * Нет правильного ответа  1. Язык Kotlin начал разрабатываться в:  * 2008 * 2009 * 2010 * 2011 * 2012  1. Плюсы языка Kotlin. Выберите несколько вариантов из списка:  * Мало инструментов для статического анализа кода * Статическая типизация * Nullsafety * Неочевидность ряда конструкций * Возможность компиляции в JavaScript и нативный код * Увеличение количества методов в приложении  1. В каком году, язык Kotlin был включен в Android Studio 3.0?  * 2016 * 2017 * 2018 * 2019 * 2020  1. Как называется функция, представляющая точку входа в программу?  * App * app * Run * run * Main * main * Program * program  1. Дан следующий код:   fun main() {  println("Hello, World!")  }  Ключевым словом здесь является:   * fun * "Hello, World!" * () * main * println  1. В каком году на конференции Google I/O язык Kotlin был объявлен приоритетным языком программирования в разработке под Android устройства?  * 2015 * 2016 * 2017 * 2018 * 2019  1. Версия Kotlin 1.2 с поддержкой JavaScript появилась в:  * 2016 * 2017 * 2018 * 2019 * 2020  1. Метод print() - это:  * Команда для печати строк * Команда для выведения сообщения на консоль * Команда для перевода каретки на следующую строку * Команда для преобразования значений  1. Метод println() - это:  * Команда для печати строк * Команда для выведения сообщения на консоль * Команда для перевода каретки на следующую строку * Команда для преобразования значений  1. Версия Kotlin 1.3 с поддержкой корутин появилась в:  * 2016 * 2017 * 2018 * 2019 * 2020  1. Язык Kotlin получил название в честь острова:  * В Аденском заливе * В Финском заливе * В Кадисском заливе * В Сиамском заливе * В Персидском заливе  1. Где может работать код Kotlin?  * Браузер * Рабочий стол * Виртуальная машина Java * Android-устройство * Все ответы верны  1. Нужно ли в Kotlin явно указывать тип void?  * Да * Нет * Нет правильного ответа  1. Какой из пунктов ниже является допустимой строкой?  * \\ какой-то текст \\ * "какой-то текст" * какой-то текст * ‘какой-то текст’  1. Выберите первую неправильную строку в коде ниже:   1: val number = 5  2: number= 10  3: number= "simple number"  4: number= "difficult number"   * 1 * 2 (правильный ответ) * 3 * 4  1. Оператор присваивания в Kotlin выглядит так.  * = (правильный ответ) * -> * <- * =:  1. Дан следующий код:   val a = 100  val b = 200  val c = 300  val d = c  val e = a  val f = d  Определите значение f:   * 100 * 200 * 300 * d * f | - | **-** |
| **1.1** | **-** | 1) Дан следующий код:  fun main() {      println("Hello, World!")  }  Ключевым словом здесь является:   * fun * "Hello, World!" * () * main * println   2) Официальный релиз Kotlin был выпущен в:   * 2014 * 2015 * 2016 * 2017 * 2018   3) Плюсы языка Kotlin. Выберите несколько вариантов из списка:   * Мало инструментов для статического анализа кода * Статическая типизация * Си-образный синтаксис * Неочевидность ряда конструкций * Отсутствие устоявшегося code style * Наличие лямбда выражений * Компиляция в Java Runtime   4) В каком году на конференции Google I/O язык Kotlin был объявлен приоритетным языком программирования в разработке под Android устройства?   * 2015 * 2016 * 2017 * 2018 * 2019   5) Обязательно ли в Kotlin писать весь код внутри классов?   * Да * Да, но есть исключения * Нет   7) Плюсы языка Kotlin. Выберите несколько вариантов из списка:   * Наличие лямбда выражений * Увеличение количества методов в приложении * Проблемы с производительностью * Поддержка со стороны IDE * Возможность компиляции в JavaScript и нативный код * Nullsafety * Отсутствие устоявшегося code style   8) Релиз Kotlin 1.0 был выпущен в:   * 2016 * 2017 * 2018 * 2019 * 2020   9) Версия Kotlin 1.2 с поддержкой JavaScript появилась в:   * 2016 * 2017 * 2018 * 2019 * 2020   10) Минусы языка Kotlin. Выберите несколько вариантов из списка:   * Мало инструментов для статического анализа кода * Увеличение количества методов в приложении * Компиляция в Java Runtime * Проблемы с производительностью * Возможность компиляции в JavaScript и нативный код * Nullsafety   10) Метод print() - это:   * Команда для печати строк * Команда для выведения сообщения на консоль * Команда для перевода каретки на следующую строку * Команда для преобразования значений | **-** |
| **2.1** |  | 1. Выберите все правильные способы объявления переменной типа Int.  * val Int number = 10 * val number: Int = 20 * val number = 30 * Int number = 40 * val number : Int = 50.05      1. Выберите все правильные способы объявления переменной типа String.  * val name = "Mike" * val String name = "Kate" * val name = 1528 * val name = “1683” * val name: String = "Daniel" * String name = "Anna" * val name: String = 1568  1. Выберите выражение, где возникнет ошибка несоответствия типов.  * val number: Int = 27 * val number: Int = "27" * val number= "27" * val number: String = "27"      1. В чем отличие между var и val?  * val - позволяет установить и изменять значение, var - позволяет установить значение без изменения в дальнейшем * var - позволяет установить и изменять значение, val - позволяет установить значение без изменения в дальнейшем * val служит для создания массивов, var для создания переменных * Отличий между ними нет  1. Выберите первую неправильную строку в коде ниже:   1: val pet = "Cat"  2: val breed = "Siamese cat"  3: age = 3  4: owner = "Ivan Ivanovich"  5: age = 24  6) Какое ключевое слово отвечает за объявление изменяемой переменной?   * any * val * final * var |  |
| **2.2** | **-** | 1) Дан следующий код:  val n = ?  val result = (n < 15) || (n <= 25)  Определите все возможные значения переменной a, когда результат равен true.   * 10 * 25 * 30 * 0 * 35   2) Дан следующий код:  val result = a >= b && b != c  Определите подходящее значение, где результат будет true:   * 1 2 1 * 2 1 2 * 1 2 2 * 1 1 1 * 2 2 2     3) Выберите наиболее подходящее утверждение об аргументах функций.   * Функция может принимать ноль, один или несколько аргументов. * Функция принимает ровно один аргумент. * Функция может принимать только ноль или один аргумент. * Функция всегда принимает хотя бы один аргумент.     4) Можно ли в файле Kotlin прописать код на Java?   * Можно, но лишь некоторые классы и функции * Да, можно прописывать любой Java код * Нельзя. Kotlin вмещает множество схожих функций и классов что и в Java, но писать код на Java в Kotlin файле нельзя       5) Выберите правильное утверждение о результате функции.   * Функция всегда возвращает результат или специальное значение. * Функция не возвращает ничего. * Функция не возвращает результат, только специальное значение. * Функция возвращает несколько результатов одновременно, один из этих результатов является специальным значением.     6) Какая функция предназначена для чтения данных?   * readString() * readLine() * nextString() * nextLine()     7) Дан следующий код:  val scanner = Scanner(System.`in`)  val string1 = scanner.nextLine()  val number = scanner.nextInt()  val string2 = scanner.next()    Какие входные данные может успешно прочитать этот код? После прочтения ввода каждая переменная должна иметь значение. Выберите несколько вариантов правильного ответа.     * Kotlin   25 Java   * Kotlin 25 Java * Kotlin 25   Java    8) Что будет выведено в результате выполнения кода?  var num = 9  num++  num += 6  --num  println(num)   * Будет выведено число 14 * Будет выдана ошибка * Будет выведено число -16 * Будет выведено число 16 * Будет выведено число 15 | **-** |
| **3.1** | **-** | 1) Как называется оператор, который заменяет null значения на значения по умолчанию?   * Принц * **Элвис** * Король * Пресли   2) Какие вызовы NPE являются безопасными?   * Оператор !! * **Оператор ?:** * **Оператор ?.** * Все ответы верны   3) Дан следующий код:  **private var** *name*: String? = **"Kotlin User Group"**  **fun** main() {  **val** lang = *name*?.**length**?:0  *println*(lang)  }  Каков будет результат?   * 0 * null * **17** * NPE * 15   4) Дан следующий код:  **val** language: String? = **null**  **val** version1\_0: Int? = language?.**length**  **val** result: Int? = version1\_0?.minus(1)  Каков будет результат?   * NullPointerException * 1 * Ошибка компиляции * null   5) Дан следующий код:  var stringKT: String? = "String Kotlin"  stringKT = null  *print*(stringKT)  Что выведется на консоль?   * String Kotlin * Исключение NPE * Ошибка компиляции * null * 0   6) Что вызывает NullPointerException в Kotlin?   * Оператор !! * Явный вызов исключения NPE * Доступ к свойству объекта null * Все ответы верны   7) Какой вызов NPE является небезопасным?   * Оператор !! * Оператор ?: * Оператор ?. * Все ответы верны   8) Как можно присвоить значение null переменной?   * Объявить переменную как null, присвоить не null значение * Объявить переменную как не null значение, присвоить не null значение * Объявить переменную как null, присвоить значение null * Объявить переменную как не null, присвоить значение null   9) Выберите вариант, где необходимо использовать переменную null:   * Программа не может определить точное значение переменной * Программа может получить значение в будущем, но пока она ей не нужна * Программа не может работать без данного значения переменной и необходимо её обязательно объявить * Все ответы верны   10) Оператор безопасного вызова обозначается следующим образом:   * ?^ * ?\* * ?. * !! * :? | **-** |
| **4.1** | **-** | 1) Что выполнит данный код?  **val** lines: Array<String> = *arrayOf*(**"a"**, **"b"**, **"c"**, **"d"**, **"5"**, **"e"**)  **for**(item **in** lines)  *print*(item)   * Будут выведены все элементы массива lines * Будут выведены все элементы массива lines, кроме значения “5” * Будет выдана ошибка * Нет правильного ответа   2) Что напечатает данный код?  **var i = 10**  **do {**  **i++**  ***print*("$i ")**  **i -= 2**  **} while (i > 1)**   * **10 9 8 7 6 5 4 3 2 1** * **10 9 8 7 6 5 4 3 2** * **10 9 8 7 6 5 4 3** * **10 9 8 7 6 5 4** * **11 10 9 8 7 6 5 4 3 2 1** * **11 10 9 8 7 6 5 4 3 2** * **11 10 9 8 7 6 5 4 3** * **11 10 9 8 7 6 5 4**   3) Дан следующий код:  **var** numbersOfArray: Array<Int> = *arrayOf*<Int>(1, 3, **"5"**, 7, 9)  **for**(item **in** numbersOfArray)  *print*(item)  Каков будет результат вывода?   * Будут выведены все элементы кроме элемента "5" * Будет выдана ошибка * Будут выведены все элементы из массива numbersOfArray   4) Что выполнит данный код?  **val** lines: Array<Char> = *arrayOf*(**'a'**, **'b'**, **'c'**, **'d'**, **'%'**)  **for**(item **in** lines)  *print*(item)   * Будут выведены все элементы массива lines * Будут выведены все элементы массива lines, кроме значения ‘%’ * Будет выдана ошибка * Нет правильного ответа   5) Какого цикла нет в языке Kotlin?   * for * do while * foreachIndexed * while * Все представленные циклы есть в Kotlin   6) Дан следующий код:  **for** (i **in** 1..2) {  **for** (j **in** 1..3) {  *print*(j)     }  }  Что выведет на экран print(j)?   * 121123 * 112121 * 123121 * 123123   7) Что выполнит данный код?  val numbers: Array<Int> = *arrayOf*<Int>(5, 7, 12, 4, "13", 2)  for(item in numbers)  *print*(item)   * Будут выведены все элементы массива numbers * Будут выведены все элементы массива numbers, кроме значения “13” * Будет выдана ошибка * Нет правильного ответа   8) Что напечатает данный код?  **var i = 10**  **do {**  **i++**  ***print*("$i ")**  **i -= 4**  **} while (i > 1)**   * **10 7 4 1** * **10 7 4** * **11 8 5 1** * **11 8 5**   9) Что выполнит данный код?  **val** lines: Array<String> = *arrayOf*(**"a"**, **"b"**, **"c"**, **"d"**, 5, **"e"**)  **for**(item **in** lines)  *print*(item)   * Будут выведены все элементы массива lines * Будут выведены все элементы массива lines, кроме значения “5” * Будет выдана ошибка * Нет правильного ответа   10) Что выполнит данный код?  **val** lines: Array<String> = *arrayOf*(**"a"**, **"b"**, **"c"**, **"d"**, **"5"**, **"e"**)  **for**(item **in** lines)  *print*(item)   * Будут выведены все элементы массива lines * Будут выведены все элементы массива lines, кроме значения “5” * Будет выдана ошибка * Нет правильного ответа | **-** |
| **5.1** | **-** | 1) Дан массив из целых чисел. Как получить последний элемент данного массива?   * numbersOfArray[100\_000\_000] * numbersOfArray[numbersOfArray.size] * numbersOfArray[numbersOfArray.size + 1] * numbersOfArray[numbersOfArray.size - 1]   2) Какие функции в MutableMap отвечают за удаление записей?   * remove() * clear() * add() * delete() * clean() * size   3) Дан следующий код:  val setOfNum = *mutableMapOf*<Int, String>()  setOfNum[3] = "2"  setOfNum[0] = "4"  setOfNum[1] = "3"  setOfNum[2] = "1"  for (entry in setOfNum) {  *print*(entry.value)  }  Каков будет результат программы?   * 4312 * 3124 * 2431 * 1234 * 4311   4) Дан следующий код:  val numOfArr: MutableList<Int> = *mutableListOf*(5, 10, 15)  numOfArr.addAll(2, *listOf*(-5, -10, -15))  numOfArr.remove(5)  for (n in numOfArr) {  *print*(n)  }  Какой будет вывод?   * 10-5-10-15 * 51015-5-10 * 10-5-10-1515 * 51015-5-10-15 * 51510-5-15   5) Дан следующий код:  **val** doubles = DoubleArray(6)  *println*(doubles.*joinToString*())  Каков будет результат?   * 0, 0, 0, 0, 0 * 0, 0, 0, 0, 0, 0 * 0.0, 0.0, 0.0, 0.0, 0.0, * 0.0, 0.0, 0.0, 0.0, 0.0, 0.0   6) Дан следующий код:  **val** lang = *arrayOf*(**"kotlin"**, **"java"**, **"c#"**, **"javascript"**, **"scala"**, **"haskell"**)  Как получить значение “haskell”?   * println(lang.last()) * println(lang.lastIndex) * println(lang.takeLast(6)) * println(lang.takeLast(5))   7) Дан следующий код:  **val** lang = *arrayOf*(**"kotlin"**, **"java"**, **"c#"**, **"javascript"**, **"scala"**, **"haskell"**)  Как получить значение “kotlin”?   * println(lang.last()) * println(lang.first()) * println(lang.indexOfFirst { true }) * println(lang.first { true }) * println(lang.takeLast(5))   8) Дан следующий код:  **val** arrayNum\_1 = *intArrayOf*(5, 7, 9, 11, 13, 15)  **val** arrayNum\_2 = *intArrayOf*(5, 7, 9, 11, 13, 15, 17)  **val** arrayNum\_3 = *intArrayOf*(5, 7, 9, 11, 13, 15)  **val** arrayNum\_4 = *intArrayOf*(3, 5, 7, 9, 11, 13, 15)  *println*(arrayNum\_1.*contentEquals*(arrayNum\_2))  *println*(arrayNum\_2.*contentEquals*(arrayNum\_3))  *println*(arrayNum\_1.*contentEquals*(arrayNum\_3))  *println*(arrayNum\_3.*contentEquals*(arrayNum\_4))  *println*(arrayNum\_1.*contentEquals*(arrayNum\_4))  Каков будет вывод в консоли?   * true, false, false, true, false * false, false, true, false, true * false, false, true, false, false * false, false, true, true, true * false, false, true, true, false   9) Дан следующий код:  **val** numOfArr: MutableList<Int> = *mutableListOf*(5, 10, 15)  numOfArr.add(20)  numOfArr.remove(5)  **for** (n **in** numOfArr) {  *print*(n)  }  Какой будет вывод?   * 51015205 * 51020 * 101520 * 5101520   10) Дан следующий код:  **var** nameOfStars = *mutableMapOf*(**"artist"** *to* **"Billie Eilish"**, **"actor"** *to* **"Matthew McConaughey"**)  nameOfStars[**"lives"**] = **"Los Angeles"**  nameOfStars[**"age"**] = **"Eleven"**  nameOfStars[**"children"**] = 1  *print*(nameOfStars)  Что будет выведено в консоль?   * Ошибка при выводе массива * Ошибка в первой строке * Ошибка в строке nameOfStars[**"lives"**] = **"Los Angeles"** * Ошибка в строке nameOfStars**["children"] = 1**   11) Какого массива нет в Kotlin?   * array * mutableListOf * mutableArrayOf * map * list   12) Как можно инициализировать MutableMap?   * Через конструктор с парами ключ и значение * Как массив с определенным размером * С помощью аннотации @MutableMap * Все ответы верны   13) Дан следующий код:  **val** numOfArr: MutableList<Int> = *mutableListOf*(5, 10, 15)  numOfArr.add(0, 20)  numOfArr.removeAt(2)  **for** (n **in** numOfArr) {  *print*(n)  }  Какой будет вывод?   * 51015020 * 51020 * 101520 * 5101520 * 20515   14. Дан следующий код:  **val** numbers = *intArrayOf*(7, 10, 15, 19, 21, 25, 36, 45)  *println*(numbers.**size**)  Каков будет результат?   * 0 * 9 * 8 * 7 | **-** |
| **6.1** | **-** | 1) Дан следующий код:  **fun** random(funk: String, just: String, fine: Int = 15) : String { ... }  Как её можно вызвать?   * random(“Zero”, “One”, “13”) * random(“Zero”, “One”) * random(“Zero”, “One”, 13) * random(13, “One”) * random(“Zero”, 13) * random(13, 17, “One”)   2) Дан следующий код:  **fun** sum(a: Int, b: Int = 7, c: Int = 8, d: Int) = a + b + c + d  Какой результат напечатает этот вызов?  *println*(*sum*(3, 5, 7, 9))   * 15 * 27 * 24 * 29   3) Где верно создана функция?   * **fun random(): String = "random string"** * **fun** random() { **return "random string"** } * **fun random() { *println*("random string") }** * **fun** random(**val** num1: Int, **val** num2: Int) { print($ **{** num1 \* num2 **}**) } * **fun** random(num1: Int, num2: Int): Int { **return** } ${ num1 \* num2 }   4) Найдите ошибку в вызове функции studentsList:  **fun** studentsList(bachelor: String = **"Yes"**, surname: String, name: String, mark: Int = 0)   * *studentsList*(**"No"**, **"Petrov"**, **"Petr"**) * *studentsList*(**""**, **"Petrov"**, **"Petr"**, 0) * *studentsList*(**""**, surname = **"Petrov"**, **"Petr"**, 0) * *studentsList*(**""**, **"Petrov"**, **"Petr"**, mark = 75) * *studentsList*(**""**, **"Petrov"**, name = **"Petr"**, mark = 90) * *studentsList*(**""**, **""**, **""**)   5) Найдите правильные вызовы функции mult:  **fun** mult(a: Int, b: Int, c: Int)   * ***mult*(5, 6, 7)** * ***mult*(5, 6)** * **mult(a = 5, 6, 7)** * ***mult*(5, 6, 7, 9)** * ***mult*(0, 6, 7)**   6) Какое ключевое слово позволяет передать заранее неизвестное количество параметров?   * vararg * arg * many * var   7) Что происходит с программой, если есть исключение?   * Программа не компилируется * Программа компилируется, но падает сразу после запуска * Программа выполняется до строки с исключением, затем падает * Программа выполняется полностью, игнорируя строку с исключением | **-** |
| **6.2** | **-** | 1) Дан следующий код:  **fun** outter(outterF: String) {  **val** outter\_1 = **"label\_3"**  **val** outter\_2 = **"label\_4"**  **fun** inner(innerF: String) {  *println*(outterF)  *println*(outter\_2)  *println*(innerF)  *println*(outter\_1)     }     inner(**"label\_2"**)  }  **fun** main() {  *outter*(**"label\_1"**)  }  Что будет выведено на консоль?   * label\_2   label\_3  label\_1  label\_4   * label\_1   label\_4  label\_2  label\_3   * label\_2   label\_3  label\_4  label\_1   * label\_1   label\_4  label\_3  label\_2   * Будет выдана ошибка   2) Основу лямбда-выражения составляет лямбда-оператор, который представляет из себя:   * == * -> * => * <- * <=   3) Если в функции всего один параметр, то в качестве переменной можно использовать ключевое слово:   * this * infix * it * лямбдa   4) В каком случае используется ключевое слово it?   * Если в функции используется только один аргумент * Если в функции используется только один аргумент из двух * Если в функции используются переменное количество аргументов * Если в функции используются несколько аргументов   5) Выберите правильное утверждение о результате функции.   * Функция всегда возвращает результат или значение. * Функция не возвращает ничего. * Функция не возвращает результат, только значение. * Функция возвращает несколько результатов одновременно.   6) Дан следующий код:  **val** text = **"womw! kotlimn is commol!"**  **val** deleteSymbol = text.*filter*(**?**)  Какое выражение верно для вывода текста “wow! kotlin is cool!”?   * ({c: Char: -> c != “m”}) * ({c: Char: -> c = ‘m’}) * ({c: Char: -> c != ‘m’}) * ({c: Char: -> c = “m”})   7) Выберите правильный способ объявления функции, которая вычисляет сумму двух чисел.   * sum(a: Int, b: Int): Int = a + b * fun sum(a: Int, b: Int): Int = a + b * def sum(a: Int, b: Int): Int = a + b * function sum(a: Int, b: Int): Int = a + b   8) Вам нужно объявить функцию. Что необходимо указать?   * название функции * фигурные скобки {} для тела функции * ключевое слово fun * круглые скобки () * тип return * все ответы верны   9) Дан следующий код:  **val** printAll = **{**text: String **->** *println*(text)**}**  printAll(**"Kotlin"**)  printAll(**"Java"**)  printAll(**"Kotlin"** + **"Java"**)  printAll(15)  printAll(15 + 18)  Что напечатает данный код?   * Kotlin   Java  KotlinJava  15  33   * Kotlin   Java  Kotlin Java  15  15 + 18   * Kotlin   Java  Kotlin + Java  15  33   * Будет выдана ошибка   10) \_\_\_\_\_\_\_\_\_\_\_\_    позволяют добавить функционал к уже определенным типам:   * Лямбда-выражение * Лямбда-оператор * Функции расширения * Локальные функции * Однострочные функции   11) Дан следующий код:  **fun** convertNumber(findResult: (Int) -> Int) : Int {  **val** result = findResult(10)  **return** result  }  Какое выражение верно для вывода числа 200?   * convertNumber {it \* it \* 1} * convertNumber {it \* it \* 2} * convertNumber {10 \* it \* 1} * convertNumber {10 \* 10 \* it} | **-** |
| **6.3** | **-** | 1) Функция расширения - это функция, которая ...   * Написана внутри класса * Не связана ни с одним классом * Добавляет поведение к уже определенному типу * Принимает объект-получатель в качестве первого параметра   2) Если в функции всего один параметр, то функция может быть помечена с помощью ключевого слова:   * this * infix * it * лямбдa   3) Дан следующий код:  **fun Int.result(): Int = this / this + this**  Что нужно написать, чтобы вывести результат 9?   * *println*(5 + 1.*result*()) * *println*(5 + 2.*result*()) * *println*(5 + 3.*result*()) * *println*(5 + 4.*result*()) * *println*(5 + 5.*result*())   4) Дан следующий код:  **fun** Int.square(): Int = **this** \* **this**  Что нужно написать, чтобы вывести квадрат числа?   * (square(5)) * (square(), 5) * (5, square()) * (5.square())   5) Представьте, что есть функция с именем selectNumber(), которая принимает один аргумент. Как вызвать эту функцию?   * fun arg.selectNumber * fun selectNumber * fun selectNumber [arg] * fun selectNumber (arg)   6) Выберите правильный способ объявления функции, которая вычисляет произведение двух чисел.   * mult(a: Int, b: Int): Int = a \* b * fun mult(a: Int, b: Int): Int = a \* b * def mult(a: Int, b: Int): Int = a \* b * function mult(a: Int, b: Int): Int = a \* b   7) Дан следующий код:  **fun** printSymblols(sym: Char, count: Int = 2): Char  Как его вызвать?   * printSymbols() * printSymbols(“b”) * printSymbols(‘b’) * printSymbols(‘b’, 7) * printSymbols(“b”, 7.5) * printSymbols(‘mb’, 7.5)   8) Дан следующий код:  **fun** Int.sum(): Int = **this** + **this**  Что нужно написать, чтобы вывести сумму чисел?   * (sum(5)) * (sum(), 5) * (5, sum()) * (5.sum())   9) Выберите наиболее подходящее утверждение об аргументах функций.   * Функция принимает ровно один аргумент. * Функция может принимать ни одного, один или несколько аргументов. * Функция может принимать только ни одного или один аргумент. * Функция всегда принимает хотя бы один аргумент | **-** |
| **7.1** | **-** | 1) Дан следующий код:  **class** Car {  **val run**: Boolean = **true**  **var stop**: Boolean = **false**  }  **fun** main() {  **var** car = Car()  *// ?*  }  Определите операции, которые могут быть выполнены в функции main:   * car = 5 * car.run = true * car = Car() * Все ответы верны   2) Дан следующий код:  **fun** main() {  **val** book1 = Book(**"Jane Air"**)  book1.**length** = 869  **val** book2 = Book(**"Pride and Prejudice"**)  book2.**length** = 599  *print*(book1.**length** - book2.**length**)  }  **class** Book(**val name**: String) {  **var length**: Int = 0  **set**(value) {  **field** = **if** (value < 0) value **else** +value  }  }  Что будет выведено на консоль?   * - 269 * 269 * - 270 * 270 * 1468 * -168   3) Дан следующий код:  **fun** main() {  **val** cat = Animal(**"Musya"**)  cat.**age** = 3  **val** dog = Animal(**"Lusya"**)  *print*(dog.**age** - cat.**age**)  }  **class** Animal(**val name**: String) {  **var age**: Int = 7  }  Что будет выведено на консоль?   * 3 * 4 * 5 * 7 * 10   4) Метод set -   * Управляет получением значения * Определяет логику установки значения * Управляет получением и установкой значения * Все ответы верны   5) Ключевое слово для объявления классов в Kotlin является:   * interface * new * class * var * object * type * val   6) Дан следующий код:  **class** Car {  **val run**: Boolean = **true**  **var stop**: Boolean = **false**  }  **fun** main() {  **var** car = Car()  *// ?*  }  Определите операции, которые могут быть выполнены в функции main:   * car = 5 * car.run = true * car.stop = true * Все ответы верны | **-** |
| **7.2** | **-** | 1) Как называется специальный член класса, который позволяет объекту инициализировать себя при создании?   * Setter * Class * Parametr * Constructor * Getter   2) Функция расширения - это функция, которая ...   * Написана внутри класса. * Не связана ни с одним классом. * Добавляет поведение к уже определенному типу. * Принимает объект в качестве первого параметра.   3) Как создать класс без конструктора?   * Добавить блок инициализации и фигурные скобки * Убрать после имени класса круглые скобки и добавить фигурные скобки * Добавить круглые скобки и убрать фигурные скобки * Все ответы верны   4) Какая функция в классе вызывается при создании объекта?   * Деструктор * Все перечисленные * Конструктор * Функция init   5) Где неправильно передаются параметры в функцию:  **fun** sum(first: Int = 0, second: Int) {  *print*(**"Result: ${**first + second**}"**)  }   * sum(5) * sum(second = 5, first = 2) * sum(second = 5) * sum(2, 5)   6) Может ли быть несколько блоков инициализации в одном классе?   * Нет, должен быть только один блок * Нет, не должно быть ни одного блока * Да, может быть несколько блоков * Нет правильного ответа   7) Вторичный конструктор -   * Объявляется после вызова функции * Объявляется вне тела класса * Объявляется внутри тела класса * Нет правильного ответа   8) Дан следующий код:  **class** Square(**var value**: Int) {  **fun** double() {  **value** \* **value**  }  }  Какие операции можно выполнить?   * val operation = Square() * val operation = Square(5) * Square().double() * Square().double(1) * Square(1).double() * Square(1).double(1)   9) Дан следующий код:  **class** Machine(**var drive**: Boolean)  **fun** Machine.engineOff() {  **this**.**drive** = **false**  }  **fun** Machine.engineOn() {  **this**.**drive** = **true**  }  **fun** main() {  **val** car = Machine(**false**)  car.*engineOff*()  *println*(car.**drive**)  }  Данный код:   * Выведет true * Выведет false * Выдаст ошибку * Нет правильного ответа   10) Основной конструктор -   * Объявляется внутри тела класса * Объявляется вне тела класса * Объявляется после вызова функции * Нет правильного ответа   11) Где неправильно передаются параметры в функцию:  **fun** mult(first: Int = 0, second: Int) {  *print*(**"Result: ${**first \* second**}"**)  }   * mult(5) * mult(second = 5, first = 2) * mult(second = 5) * mult(2, 5) | **-** |
| **7.3** | **-** | 1) Каждый класс может наследоваться от:   * Нескольких классов. * Только от одного класса. * Только от абстрактных классов. * Нет правильного ответа.   2) Какое ключевое слово используется для наследования в Kotlin?   * inherit * extend * public * open   3) Чтобы сохранить ссылку на внешний класс, нужно использовать модификатор:   * open * public * internal * override * inner * protected   4) Что разрешено делать с автоматически сгенерированными функциями?   * Функции будут автоматически созданы для всех свойств, объявленных внутри тела класса. * Вы можете переопределить автоматически сгенерированные функции более одного раза в одном классе. * Вы можете переопределить функции toString (), hashCode () и equals (). * Все ответы верны.   5) Верно ли прописан интерфейс?  **interface** Program {  **var number**: Int  **fun** change(value: String): Boolean  **fun** print(text: Any): Boolean {  *println*(**"Result: $**text**"**)  **return if**(text == **null**) **false else true**  }  }   * В интерфейсе есть ошибки * В интерфейсе не хватает модификатора доступа * В интерфейсе все прописано правильно * Нет правильного ответа   6) Метод, проверяющий равенство значений всех свойств:   * hashCode() * toString() * copy() * equals()   7) Какой метод или параметр не существует в Enum?   * value() * values() * valueOf() * ordinal * name   8) Где прописан класс, который можно будет унаследовать?   * class Student {} * abstract class Student {} * open class Student {} * abstract class Student {} и open class Student {} * Все перечисленные   9) Верно ли прописан код?  **abstract class** Client (**var login**: String, **var password**: String, **var email**: String) {  **abstract fun** getInfo()  }  **class** Data(login: String, password: String, email: String)  : Client(login, password, email) {  **fun** getInfo() {  *println*(**"Print"**)  }  }   * К классу Client нужно добавить модификатор open * В классе Data в конструкторе не прописаны слова var или val для параметров * В функции getInfo() нужно добавить модификатор override * Все ответы верны   10) За что отвечает модификатор доступа internal?   * Он делает данные доступные для просмотра из любого места программы * Он делает данные закрытыми для просмотра * Он делает данные открытые только для классов наследников * Он делает данные открытые внутри одного модуля   11) Где правильно создан дочерний класс?   * class Animals () : Cat(name: String, weight: Int) * class Animals (name: String, weight: Int, type: String) : Cat(name: String, weight: Int) * class Animals (name: String, weight: Int, val type: String) : Cat(name, weight) * class Animals (name, weight, val type: String) : Cat(name, weight)   12) Что будет выводом следующего фрагмента кода:  **fun** main() {  **val** ivan = Student(**"Ivan"**, 85.5f, 8)  **val** petr = ivan.copy(name = **"Petr"**, exams = 9)  **val** anna = petr.copy(name = **"Anna"**)  *println*(ivan.**exams** - petr.**exams** + anna.**exams** + 7)  }   * 12 * 15 * 17 * 24 * 33   13) Дан следующий код:  **open class** Samsung(**val model**: String) {  **open fun** getInfo(): String = **"$model model"**  }  **open class** Galaxy(model: String, **val id**: Int) : Samsung(model) {  **open fun** getInfo(): String = **"$id id"**  }   * Данный код будет работать нормально. * В IDE будет предупреждение, но код скомпилируется. * Не хватает модификатора override. * Нет правильного ответа.   14) Какие модификаторы связаны с переопределением функций?   * ordinal * enum * open * data * inherit * public * override   15) Интерфейсы в Kotlin похожи на интерфейсы в Java, а также:   * Могут включать свойства и реализации методов по умолчанию * Всем свойствам по умолчанию присваивается модификатор private * Всем методам по умолчанию присваивается модификатор private * Все ответы верны   16) Где правильно создано перечисление?   * enum Car {} * class Car {} * sealed class Car {} * class enum Car {} * enum class Car {}   17) Где прописан класс, который можно будет унаследовать?   * class Client {} * abstract class Client {} * open class Client {} * abstract class Client {} и open class Client {} * Все перечисленные   18) Где правильно создано перечисление?   * enum Machine {} * class Machine {} * sealed class Machine {} * class enum Machine {} * enum class Machine {} | **-** |
| **8.1** | **-** | 1) С помощью чего можно запустить приложение Android на телефоне?   * Через USB кабель * Через наушники * Через Wi-Fi * Через передачу данных   2) Для чего существуют идентификаторы id?   * Для обращения к элементам * Для редактирования элементов * Для связки элементов * Для добавления ресурсов к элементам   3) Неявные объекты Intent -   * Указывают на компонент, который требуется запустить * Содержат имя конкретного компонента * Объявляют действие, которое требуется выполнить * Обрабатывают запросы приложения   4) Какой метод запускает новый экземпляр компонента Activity?   * startMainActivity(); * startService(); * startIntent(); * startActivity();   5) В какой папке находится исходный код и ресурсы приложения?   * **src** * build * libs * mipmap   6) Что делает AVD (Android Virtual Device)?   * Компилирует и запускает проект * Файл пакета приложения Android * Моделирует физическое устройство * Собирает проект и запускает на устройстве   7) Что выведет на экран следующее выражение: android: text=”@string/hello”?   * Значение ресурсной строки с именем hello * Значение ”hello” * Значение @string/hello * Ничего не выведет   8) Для получения элементов по id в классе Activity используется метод…   * src * findId() * findViewById() * findViewById(id)   9) Назовите два вида Intent.   * Простые и сложные. * Простые и составные. * Явные и неявные.   10) Единица измерения для TextView:   * dp * px * pt * sp   11) В каком файле находятся все идентификаторы ресурсов?   * MainActivity.java * A.java * C.java * R.java   12) Единица измерения для изменения размеров элементов:   * px * sp * dp * pt   13) В каком файле хранятся строковые ресурсы приложения?   * style.xml * strings.xml * activity\_main.xml * colors.xml   14) Что содержит в себе папка values?   * Цвета, строки и стили * Значки и иконки * Файлы макета активити * Изображения и картинки   15) Что НЕ характерно для явных объектов Intent?   * Указывают на компонент, который требуется запустить. * Содержат имя конкретного компонента. * Объявляют действие, которое требуется выполнить. * Обрабатывают запросы приложения.   16) Объект Intent используется:   * Для запуска разметки. * Для запуска Android Studio. * Для запуска сообщений. * Для запуска Activity.   17) Что делает строка setContentView() в методе onCreate()?   * Определяет разметку и запускает проект * Определяет код * Определяет ошибки в проекте * Определяет разметку   18) В чем отличие активити и макета?   * В активити мы настраиваем интерфейс приложения, а в макете пишем логику приложения * В активити мы пишем логику приложения, а в макете настраиваем интерфейс приложения * Отличий нет   19) Что представляет собой компонент View?   * Содержит картинку, текст, кнопку или элемент * Содержит родительский макет LinearLayout * Содержит виджет для прокручивания картинок * Содержит логику приложения   20) Для чего используется система Gradle?   * Для компиляции, построения и упаковки приложений и библиотек. * Для настройки интерфейса программы Android Studio * Для редактирования настроек проекта в режиме реального времени. * Нет правильного ответа.   21) Атрибут для смены размера шрифта TextView:   * textAppear * textSize * text * textStyle   22) Выберите обязательные значения атрибутов для всех элементов:   * match\_parent, wrap\_content * wrap\_content, fill\_parent * text, orientation * text, gravity   23) В чем отличие папки drawable от mipmap?   * В папке drawable хранятся иконки приложения, в папке mipmap стили приложения * В папке mipmap хранятся стили приложения, в папке drawable цвета приложения * В папке drawable хранятся векторные и растровые изображения приложения, в папке mipmap иконки приложения * В папке mipmap хранятся векторные и растровые изображения приложения, в папке drawable иконки приложения   24) Что НЕ характерно для неявных объектов Intent?   * Указывают на компонент, который требуется запустить. * Не содержат имени конкретного компонента. * Объявляют действие, которое требуется выполнить. * Обрабатывают запросы приложения.   25) Что делают интенты?   * Предоставляют общий доступ к компонентам приложения при наличии определенных данных. * Запрашивают выполнение действия у компонента другого приложения и запускают компоненты приложения. * Содержат сведения «ключ-значение», которые необходимы для выполнения запрошенного действия. * Все перечисленное. | **-** |
| **Итоговое тестирование** | **-** |  | 1. Дан следующий код:   val text = "womw! kotlimn is commol!"  val deleteSymbol = text.*filter*(?)  Какое выражение верно для вывода текста “wow! kotlin is cool!”?   * ({c: Char: -> c != “m”}) * ({c: Char: -> c = ‘m’}) * ({c: Char: -> c != ‘m’}) * ({c: Char: -> c = “m”})  1. Основной целью использования локальных функций является:  * Для определения действий, которые используются в других функциях и для вызова их с этих функций. * Для определения действий, которые используются только в рамках какой-то конкретной функции и нигде больше не применяются. * Для определения действий, которые вызываются с нескольких функций. * Все ответы верны.  1. Вторичный конструктор -  * Объявляется после вызова функции * Объявляется вне тела класса * Объявляется внутри тела класса * Нет правильного ответа  1. Где неправильно передаются параметры в функцию:   fun mult(first: Int = 0, second: Int) {  *print*("Result: ${first \* second}")  }   * mult(5) * mult(second = 5, first = 2) * mult(second = 5) * mult(2, 5)  1. Дан следующий код:   class Square(var value: Int) {  fun double() {  value \* value  }  }  Какие операции можно выполнить?   * val operation = Square() * val operation = Square(5) * Square().double() * Square().double(1) * Square(1).double() * Square(1).double(1)  1. Дан следующий код:   fun convertNumber(findResult: (Int) -> Int) : Int {  val result = findResult(3)  return result  }  Какое выражение верно для вывода числа 30?   * convertNumber {it \* 5 \* 3} * convertNumber {it \* 2 \* 5} * convertNumber {3 \* it \* 5} * convertNumber {it \* it \* 3}  1. Верно ли прописан код?   abstract class Client (var login: String, var password: String, var email: String) {  abstract fun getInfo()  }  class Data(login: String, password: String, email: String)  : Client(login, password, email) {  fun getInfo() {  *println*("Print")  }  }   * К классу Client нужно добавить модификатор open * В классе Data в конструкторе не прописаны слова var или val для параметров * В функции getInfo() нужно добавить модификатор override * Все ответы верны  1. Что будет выводом следующего фрагмента кода:   fun main() {  val ivan = Student("Ivan", 85.5f, 8)  val petr = ivan.copy(name = "Petr", exams = 9)  val anna = petr.copy(name = "Anna")  *println*(ivan.exams - petr.exams + anna.exams + 7)  }   * 12 * 15 (правильный ответ) * 17 * 24 * 33  1. Где правильно создан дочерний класс?  * class Bird () : Colibri (name: String, color: String) * class Bird (name: String, color: String, type: String) : Colibri (name: String, color: String) * class Bird (name: String, color: String, val type: String) : Colibri (name, color) * class Bird (name, color, val type: String) : Colibri (name, color)  1. Верно ли прописан интерфейс?   interface Program {  var number: Int  fun change(value: String): Boolean  fun print(text: Any): Boolean {  *println*("Result: $text")  return if(text == null) false else true  }  }   * В интерфейсе есть ошибки * В интерфейсе не хватает модификатора доступа * В интерфейсе все прописано правильно * Нет правильного ответа  1. Дан следующий код:   open class Samsung(val model: String) {  open fun getInfo(): String = "$model model"  }  open class Galaxy(model: String, val id: Int) : Samsung(model) {  open fun getInfo(): String = "$id id"  }   * Данный код будет работать нормально. * В IDE будет предупреждение, но код скомпилируется. * Данный код не скомпилируется. * Нет правильного ответа.  1. Дан следующий код:   class Machine(var drive: Boolean)  fun Machine.engineOff() {  this.drive = false  }  fun Machine.engineOn() {  this.drive = true  }  fun main() {  val car = Machine(false)  car.*engineOn*()  *println*(car.drive)  }  Данный код:   * Выведет true * Выведет false * Выдаст ошибку * Нет правильного ответа  1. Где неправильно передаются параметры в функцию:   fun div(first: Int = 0, second: Int) {  *print*("Result: ${first / second}")  }   * div(5) * div(second = 5, first = 2) * div(second = 5) * div(2, 5)  1. Основной конструктор -  * Объявляется внутри тела класса * Объявляется вне тела класса * Объявляется после вызова функции * Нет правильного ответа  1. Может ли быть несколько блоков инициализации в одном классе?  * Нет, должен быть только один блок * Нет, не должно быть ни одного блока * Да, может быть несколько блоков * Нет правильного ответа  1. Дан следующий код:   fun outter(outterF: String) {  val outter\_1 = "label\_3"  fun inner(innerF: String) {  *println*(outterF)  *println*(innerF)  *println*(outter\_1)  }  inner("label\_1")  }  fun main() {  *outter*("label\_2")  }  Что будет выведено на консоль?   * label\_2   label\_3  label\_1   * label\_1   label\_2  label\_3   * label\_2   label\_1  label\_3   * Будет выдана ошибка  1. Выберите правильный способ объявления функции, которая вычисляет деление двух чисел.  * div(a: Int, b: Int): Int = a / b * fun div(a: Int, b: Int): Int = a / b * def div(a: Int, b: Int): Int = a / b * function div(a: Int, b: Int): Int = a / b  1. Дан следующий код:   val printAll = {text: String -> *println*(text)}  printAll("Kotlin")  printAll("Java")  printAll("Kotlin" + "Java")  printAll("15")  printAll("15 + 18")  Что напечатает данный код?   * Kotlin   Java  KotlinJava  15  33   * Kotlin   Java  KotlinJava  15  15 + 18   * Kotlin   Java  Kotlin + Java  15  33   * Будет выдана ошибка  1. Дан следующий код:   fun Int.sum(): Int = this + this  Что нужно написать, чтобы вывести сумму чисел?   * (sum(5)) * (sum(), 5) * (5, sum()) * (5.sum())  1. Представьте, что есть функция с именем selectNumber(), которая принимает один аргумент. Как вызвать эту функцию?  * fun arg.selectNumber * fun selectNumber * fun selectNumber [arg] * fun selectNumber (arg) |

* 1. **описание показателей и критериев оценивания, шкалы оценивания.**

**Шкалы оценивания**

* 1. Входное тестирование

**Шкала критериев оценивания по входному тестированию**

|  |  |  |
| --- | --- | --- |
| **Шкала оценивания** | **Критерии** | **Описание** |
| 70%-100% верных ответов на вопросы входного теста | **зачтено** | Потенциальный обучающийся успешно выполнил входной тест и может быть допущен к обучению по программе |
| 69% и менее верных ответов на вопросы входного теста | **не зачтено** | Потенциальный обучающийся не выполнил успешно входной тест и не может быть допущен к обучению по программе |

* 1. Промежуточное тестирование по темам/модулям

**Шкала критериев оценивания по промежуточному тестированию**

|  |  |  |  |
| --- | --- | --- | --- |
| **Шкала** | **Критерии** | **Описание** | **Уровень сформированности компетенции** |
| 80%-100% верных ответов на вопросы промежуточного теста | зачтено | Обучающийся успешно выполнил промежуточный тест по теме/модулю | Продвинутый уровень |
| 70%-79% | зачтено | Обучающийся успешно выполнил промежуточный тест по теме/модулю | Базовый уровень |
| 50%-69% верных ответов на вопросы промежуточного теста | не зачтено | Обучающийся не выполнил успешно промежуточный тест, необходимо повторное назначение теста и консультации преподавателя | Начальный уровень |
| Менее 50% верных ответов на вопросы промежуточного теста | не зачтено | Обучающийся не выполнил успешно промежуточный тест, необходимо повторное назначение теста и консультации преподавателя | Не владеет, компетенция не сформирована |

* 1. Итоговое тестирование по программе

**Шкала критериев оценивания по итоговому тестированию**

|  |  |  |  |
| --- | --- | --- | --- |
| **Шкала** | **Критерии** | **Описание** | **Уровень сформированности компетенции** |
| 80%-100% верных ответов на вопросы промежуточного теста | зачтено | Обучающийся успешно выполнил итоговый тест по программе | Продвинутый уровень |
| 70%-79% | зачтено | Обучающийся успешно выполнил итоговый тест по программе | Базовый уровень |
| 50%-69% верных ответов на вопросы промежуточного теста | не зачтено | Обучающийся не выполнил успешно итоговый тест по программе | Начальный уровень |
| Менее 50% верных ответов на вопросы промежуточного теста | не зачтено | Обучающийся не выполнил успешно итоговый тест по программе | Не владеет, компетенция не сформирована |

* 1. **примеры контрольных заданий по модулям или всей образовательной программе.**

**Примеры контрольных заданий по программе**

Создайте класс «Airplane» со значениями:

* поля: имя компании, год основания, количество эксплуатируемых моделей самолета на сегодняшний день;
* конструктор через который устанавливаются все поля;
* метод для вывода имени компании.

Создайте класс-наследник «Airbus» <https://ru.wikipedia.org/wiki/Airbus> со значениями:

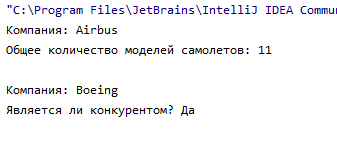
* поля: количество моделей самолетов;
* конструктор для установки всех переменных в том числе переменных что находятся в родительском классе;
* метод для вывода всех переменных из данного класса и класса родителя.

Создайте еще один класс-наследник «Boeing» <https://ru.wikipedia.org/wiki/Boeing> со значениями:

* поля: является ли конкурентом;
* конструктор для установки всех переменных в том числе переменных что находятся в родительском классе;
* метод для вывода всех переменных из данного класса и класса родителя. В методе создайте проверку на поле «является ли конкурентом». Выводить слово «Да» в случае если является конкурентом и слово «Нет», если значение этой переменной false. [https://ru.wikipedia.org/wiki/Конкуренция\_между\_Airbus\_и\_Boeing](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D0%BA%D1%83%D1%80%D0%B5%D0%BD%D1%86%D0%B8%D1%8F_%D0%BC%D0%B5%D0%B6%D0%B4%D1%83_Airbus_%D0%B8_Boeing)

Создайте массив и поместите в него два объекта на основе классов-наследников. При помощи цикла выведите на экран все данные про объекты, используя метод из класса.

Пример вывода информации из двух объектов:



* 1. **тесты и обучающие задачи (кейсы), иные практико-ориентированные формы заданий.**

Примеры практических заданий представлены в таблице 7.1.

Примеры тестов представлены в таблице 8.1.

В программе используются практические задания со свободным ответом (с последующей проверкой преподавателем), которые являются обязательными для выполнения по каждой теме. Дополнительно для организации самостоятельной работы используются задания со свободным ответом (в том числе проблемные), которые слушатели выполняют самостоятельно и которые также проверяются преподавателем.

**Примеры заданий для самостоятельной работы**

**Модуль 1.**

Самостоятельная работа – 1 ак.час

* + - 1. **Установка IntelliJ IDEA**

Для установки Intellij IDEA, необходимо перейти на официальный веб-сайт компании JetBrains:  <https://www.jetbrains.com/idea/>

Скачать JDK: [https://www.oracle.com/technetwork/java/javase/downloads/ındex.html](https://www.oracle.com/technetwork/java/javase/downloads/%C4%B1ndex.html)

* + - 1. **Отличие между Java и Kotlin**

Познакомьтесь с официальной документацией языка Kotlin. Для этого воспользуйтесь ссылками:

Официальная документация языка: <https://kotlinlang.org/docs/reference/>

Русскоязычный сайт с документацией: <https://kotlinlang.ru>

Обратите внимание, что сайт Kotlin поддерживается энтузиастами-разработчиками, поэтому на этом сайте можно найти полезные практические рекомендации. Найдите отличия между Java и Kotlin

Составьте краткий отчет о выполнении задания

**Модуль 2.**

Самостоятельная работа – 2 ак.час

1. Познакомьтесь самостоятельно на сайте документации Kotlin с разделом «Именование переменных»: Coding conventions <https://kotlinlang.org/docs/reference/coding-conventions.html>

Составьте краткий отчет о выполнении задания

**Модуль 3.**

Самостоятельная работа – 2 ак.час

1. Познакомьтесь самостоятельно на сайте документации Kotlin с разделом «Типы данных. Базовые типы»: <https://kotlinlang.org/docs/reference/basic-types.html> и <https://kotlinlang.ru/docs/reference/null-safety.html>Составьте краткий отчет о выполнении задания

**Модуль 4.**

Самостоятельная работа – 2 ак.час

1. Познакомьтесь самостоятельно на сайте документации Kotlin с разделом «Управляющие конструкции»: https://kotlinlang.ru/docs/reference/control-flow.html . Составьте краткий отчет о выполнении задания

**Модуль 5.**

Самостоятельная работа – 1 ак.час

1. Введение в коллекции. Познакомьтесь с полным списком интерфейсов и классов, которые представляют коллекции. Для этого пройдите по ссылке на страницу: https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/index.html
2. Работа с коллекцией Познакомьтесь с коллекциями:

https://kotlinlang.ru/docs/reference/collections.html

<https://kotlinlang.org/docs/reference/collections-overview.html>

Составьте краткий отчет о выполнении задания

**Модуль 6.**

Самостоятельная работа – 3 ак.час

1. Познакомьтесь с полным списком функций, которые представляют функциональное программирование. Для этого пройдите по ссылке на страницы:

<https://kotlinlang.ru/docs/reference/functions.html>

<https://kotlinlang.ru/docs/reference/lambdas.html>

https://kotlinlang.ru/docs/reference/inline-functions.html

Составьте краткий отчет о выполнении задания

**Модуль 7.**

Самостоятельная работа – 8 ак.час

1. Познакомьтесь с полным списком классов и объектов, которые представляют объектно-ориентированное программирование. Для этого пройдите по ссылке на следующие страницы:

<https://kotlinlang.ru/docs/reference/classes.html>

<https://kotlinlang.ru/docs/reference/properties.html>

<https://kotlinlang.ru/docs/reference/interfaces.html>

<https://kotlinlang.ru/docs/reference/visibility-modifiers.html>

<https://kotlinlang.ru/docs/reference/extensions.html>

<https://kotlinlang.ru/docs/reference/data-classes.html>

<https://kotlinlang.ru/docs/reference/sealed-classes.html>

<https://kotlinlang.ru/docs/reference/generics.html>

<https://kotlinlang.ru/docs/reference/object-declarations.html>

<https://kotlinlang.ru/docs/reference/nested-classes.html>

<https://kotlinlang.ru/docs/reference/enum-classes.html>

Составьте отчет в шаблоне задания

**Модуль 8.**

Самостоятельная работа – 2 ак.час

1. Познакомьтесь и поработайте сайтом Startandroid (<https://startandroid.ru/ru/>). Найдите еще не менее трех сайтов по введению в Android-разработку на Kotlin. Составьте отчет о проделанной работе (по схеме в задании)

**8.5.**  **описание процедуры оценивания результатов обучения.**

В процессе обучения слушатели должны выполнить практические задания со свободным ответом, которые проверяются преподавателем (в рамках практических занятий) и пройти промежуточную аттестацию по теме или модулю – электронное тестирование. Промежуточное тестирование включает 8-20 вопросов по теме или модулю. Промежуточное тестирование считается успешно завершённым, если слушателем даны верные ответы на не менее чем 70% вопросов теста. Каждый слушатель имеет три попытки на выполнение промежуточного теста, время тестирования – 0,5 часа -1 час (в зависимости от количества вопросов по теме/модулю).

К итоговой аттестации допускаются слушатели, **выполнившие все практические задания (в рамках практических занятий) и промежуточные тесты по темам/модулям (не менее 70% верных ответов по каждому тесту по теме/модулю)**, предусмотренные учебной программной. В ходе итогового электронного тестирования слушателю предлагается ответить на 20 вопросов. Тестирование считается успешно завершённым, если слушателем даны верные ответы на не менее чем 70% вопросов теста. Каждый слушатель имеет одну попытку на выполнение итогового теста, время тестирования – 1 час.

**9.Организационно-педагогические условия реализации программы**

**9.1. Кадровое обеспечение программы**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№**  **п/п** | **Фамилия, имя, отчество (при наличии)** | **Место основной работы и должность, ученая степень и ученое звание (при наличии)** | **Ссылки на веб-страницы с портфолио (при наличии)** | **Фото в формате jpeg** | **Отметка о полученном согласии на обработку персональных данных** |
| **1** | **Романова**  **Ассель** | Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО».  Университет ИТМО, старший преподаватель  Разработчик, JetBrains Academy |  |  | **+** |

**9.2.Учебно-методическое обеспечение и информационное сопровождение**

|  |  |
| --- | --- |
| **Учебно-методические материалы** | |
| Методы, формы и технологии | Методические разработки,  материалы курса, учебная литература |
| Методы: проблемно-развивающие методы; исследовательский, объяснительно-иллюстративный, метод проблемного изложения, частично-поисковый, методы контроля и самоконтроля  Формы: самостоятельная работа по освоению теоретического материала; самоконтроль через ответы на вопросы по видеолекциям и выполнение обязательных практических заданий и заданий для самостоятельной работы; промежуточный контроль (тесты по темам/модулям для промежуточной аттестации), итоговый контроль (итоговое тестирование по курсу). | * + - 1. [Скин Джош](http://www.labirint.ru/authors/218412/?p=30658), [Гринхол Дэвид](http://www.labirint.ru/authors/218413/?p=30658). [Kotlin. Программирование для профессионалов. Из-во: Питер. 2020. 464 с.](http://www.labirint.ru/books/709995/?p=30658)       2. Жемеров Д. Б., Исакова С. С. [Kotlin в действии. Издательство: ДМК-Пресс. 2018. 402 с.](http://www.labirint.ru/books/612984/?p=30658)       3. [Симон Пьер-Ив](http://www.labirint.ru/authors/224411/?p=30658). [Волшебство Kotlin. Издательство: ДМК-Пресс. 2018. 536 с.](http://www.labirint.ru/books/731858/?p=30658)       4. [Гриффитс Д](http://www.labirint.ru/authors/127543/?p=30658)[.,](http://www.labirint.ru/books/731858/?p=30658) [Гриффитс Д.](http://www.labirint.ru/authors/127584/?p=30658)[Head First. Kotlin Из-во: Питер. 2020. 464 с.](http://www.labirint.ru/books/722612/?p=30658) |
|  |  |

|  |  |
| --- | --- |
| **Информационное сопровождение** | |
| Электронные  образовательные ресурсы | Электронные  информационные ресурсы |
| Котлин - быстрый старт [MOOC] Режим доступа: <https://www.udemy.com/course/kotlin_sumin/> | 1. Документация по языку Kotlin от JetBrains [Электронный ресурс]. Режим доступа: https://kotlinlang.org/docs/reference/ 2. Гайдлайн по разработке приложений от Google JetBrains [Электронный ресурс]. Режим доступа: https://codelabs.developers.google.com/kotlin-bootcamp/ |

**9.3.Материально-технические условия реализации программы**

|  |  |
| --- | --- |
| Вид занятий | Наименование оборудования,  программного обеспечения |
| Лекционные (видеолекции) | Операционная система семейств Linux Mac Os, Windows.  Браузер Google Chrome |
| Практические | Операционная система семейств Linux Mac Os, Windows.  Интегрированная среда разработки: IntelliJ IDEA, Android Studio |
| Самостоятельная работа | Операционная система семейств Linux Mac Os, Windows.  Интегрированная среда разработки: IntelliJ IDEA, Android Studio |

**III. Паспорт компетенций (Приложение 2)**

Автономная некоммерческая организация дополнительного профессионального образования

«СофтЛайн Эдюкейшн»

.

ПАСПОРТ КОМПЕТЕНЦИИ

**Дополнительная профессиональная программа повышения квалификации**

**«Основы программирования на языке Kotlin (базовый курс)»**

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | Наименование компетенции | Способен ставить и решать прикладные задачи по созданию и совершенствованию программного кода с использованием языка программирования Kotlin | |
| 2. | Указание типа компетенции | профессиональная | |
| 3. | Определение, содержание и основные сущностные характеристики компетенции | Под компетенцией понимается способность постановки и нахождения путей решения прикладных задач по созданию и совершенствованию программного кода с использованием языка программирования Kotlin  Слушатель должен:  *знать*:  - синтаксис языка программирования Kotlin и особенности программирования на Kotlin;  - особенности выбранной среды программирования;  - место языка программирования Kotlin среди развивающихся языков программирования;  - технологии программирования на языке Kotlin;  - особенности Android-разработки;  *уметь***:**  - применять Kotlin для написания программного кода;  - использовать инструментальные средства разработки, доступные у платформы IntelliJ IDEA;  - осуществлять подготовку данных для реализации работы над приложением;  *владеть*:  - приемами создания программного кода на Kotlin в соответствии с спецификациями;  - приемами и методами реализации базовых алгоритмических конструкции на языке программирования Kotlin;  - способами определения и манипулирования данными на языках Java и Kotlin;  - приемами работы в интегрированной среде разработки, позволяющей реализовывать разработку проекта;  - приемами запуска и отладки программного кода в интегрированной среде разработки | |
| 4. | Дескриптор знаний, умений и навыков по уровням | Уровни сформированности компетенции  обучающегося | Индикаторы |
|  | Начальный уровень | *Знает:* синтаксис Kotlin.  *Умеет:* применять Kotlin для написания программного кода при решении простейших линейных алгоритмических задач.  *Владеет:* приемами разработки, запуска и отладки простейших сценариев на Kotlin в интегрированной среде разработки |
|  | Базовый уровень | *Знает*: особенности программирования Kotlin и выбранной среды программирования; место языка программирования Kotlin среди развивающихся языков программирования.  *Умеет*: применять Kotlin для написания программного кода при решении алгоритмических задач ветвления и повторения; использовать инструментальные средства разработки, доступные у платформы IntelliJ IDEA.  *Владеет*: приемами и методами реализации алгоритмических структур ветвления и повторения на Kotlin, приемами создания программного кода на Kotlin в соответствии с спецификациями |
|  | Продвинутый | *Знает*: технологии программирования на Kotlin; особенности Android-разработки.  *Умеет*: применять Kotlin для написания программного кода при решении прикладных задач с использованием адекватных алгоритмических структур; осуществлять подготовку данных для реализации работы над приложением  *Владеет*: способами определения и манипулирования данными на языках Java и Kotlin; приемами работы в интегрированной среде разработки, позволяющей реализовывать разработку проекта |
| 5. | Характеристика взаимосвязи данной компетенции с другими компетенциями/ необходимость владения другими компетенциями для формирования данной компетенции | Для формирования данной компетенции слушатель должен владеть:   * компетенциями цифровой грамотности, в том числе навыками владения ПК на уровне среднего или продвинутого пользователя; * компетенции в области алгоритмизации (знание основных алгоритмических структур, умения выражать их на псевдокоде и в виде блок-схем) и основ программирования, базовыми знаниями языка Java | |
| 6. | Средства и технологии оценки | Тесты, электронное промежуточное (по темам/модулям) и итоговое тестирование | |

**IV.Иная информация о качестве и востребованности образовательной программы** (результаты профессионально-общественной аккредитации образовательной программы, включение в системы рейтингования, призовые места по результатам проведения конкурсов образовательных программ и др.) (при наличии)

**V.Рекомендаций к программе от работодателей**: наличие не менее двух писем и/или подтверждения на цифровой платформе Государственной системы предоставления ПЦС от работодателей о рекомендации образовательной программы для реализации в рамках Государственной системы предоставления ПЦС на формирование у трудоспособного населения компетенций цифровой экономики с указанием востребованности результатов освоения программы в сфере деятельности соответствующих компаний и готовности к рассмотрению заявок наиболее успешно освоивших образовательную программу граждан на прохождение стажировки и (или) собеседования на предмет трудоустройства путем проставления отметки в профиле программы

**Загружено на платформу**

**VI.Указание на возможные сценарии профессиональной траектории граждан** по итогам освоения образовательной программы (в соответствии с приложением)

**СЦЕНАРИИ ПРОФЕССИОНАЛЬНОЙ ТРАЕКТОРИИ ГРАЖДАН**

|  |  |
| --- | --- |
| **Цели получения персонального цифрового сертификата** | |
| **текущий статус** | **цель** |
| **Развитие компетенций в текущей сфере занятости** | |
| работающий по найму в организации, на предприятии | сохранение текущего рабочего места |
| работающий по найму в организации, на предприятии | развитие профессиональных качеств |
| работающий по найму в организации, на предприятии | повышение заработной платы |
| работающий по найму в организации, на предприятии | смена работы без изменения сферы профессиональной деятельности |

**VII.Дополнительная информация**

**VIII.Приложенные Скан-копии**

**Генеральный директор**

**АНО ДПО «СофтЛайн Эдюкейшн» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.Э. Разуваев**

М.П.