



Язык программирования Kotlin.фип_БАК_МОиАИС_н/с

- 1 Где можно скачать Java Development Kit (JDK)?
- 2 Как можно проверить корректность установки JDK?
- 3 Какой шаг идет первым при установке IntelliJ IDEA?
- 4 В IntelliJ IDEA вы можете ...
- 5 Как задать неизменяемую переменную?
- 6 Неверно, что ... существуют в kotlin
- 7 Цикл ... позволяет перебирать элементы в заданном диапазоне или коллекции
- 8 Сортировка ... — это алгоритм сортировки сравнением на месте, который создает окончательный отсортированный массив по одному элементу за раз
- 9 Операция ..., позволяет прервать выполнение цикла и перейти к следующей инструкции после цикла
- 10 Если требуется выполнить блок кода хотя бы один раз, можно использовать цикл
- 11 ... — это концепция, означающая невозможность изменить состояние объекта после его создания
- 12 Операции ... позволяют повторять блок кода несколько раз, пока выполняется определенное условие
- 13 ... — это простой алгоритм поиска, который последовательно проверяет каждый элемент коллекции до тех пор, пока не будет найдено совпадение или не будет достигнут конец коллекции
- 14 ... - мощный инструмент автоматизации сборки и управления зависимостями в проектах Java. Он позволяет разработчикам легко настраивать и собирать проекты, добавлять и обновлять зависимости, а также выполнять различные задачи автоматически.
- 15 ... — это процесс преобразования переменной из одного типа данных в другой.





- 16) ... разрешают выполнять различные действия в зависимости от того, достигнут ли прогресс в соблюдении условий или нет.
- 17) Расставьте шаги настройки IntelliJ IDEA в правильном порядке:
- 18) Расставьте шаги алгоритма пузырьковой сортировки в правильном порядке:
- 19) Сопоставьте тип переменной и её значение:
- 20) Цикл ... будет выводить числа от 1 до 10
- 21) Сопоставьте операторы условия с их функцией:
- 22) ... классы используются для представления ограниченных иерархий классов, когда значение может иметь один из типов из ограниченного набора, но не может иметь никакой другой тип
- 23) В Kotlin классы и методы закрыты по умолчанию, то есть они объявляются как заключительные. Для того чтобы класс стал суперклассом и мог быть унаследован, мы должны его объявить с ключевым словом "..."
- 24) Методы и свойства, которые переопределены в подклассе, должны быть снабжены ключевым словом "..."
- 25) Интерфейс в Kotlin работает подобно классу, но интерфейс не может содержать ...: в нем могут быть объявлены абстрактные методы и реализованные методы
- 26) Чтобы класс реализовывал интерфейс, мы используем ключевое слово "..."
- 27) В Kotlin класс ... реализовывать более одного интерфейса
- 28) ... - это архитектурный паттерн проектирования, который разделяет приложение на три основных компонента
- 29) Когда мы объявляем класс как data, компилятор автоматически выводит следующие функции
- 30) За наследование отвечает символ
- 31) Интерфейс объявляется с использованием ключевого слова "..."
- 32) ... - это передача объекта в другой объект





- 33) Kotlin использует ключевое слово ``...`` для объявления синглтона, которое означает, что можно создать только один экземпляр класса
- 34) ... - это экземпляры классов. Они представляют собой воплощение классов, с которыми можно работать
- 35) В Kotlin модификатор ``...`` используется для указания, что соответствующий член доступен везде в том же модуле
- 36) Обычные классы объявляются с помощью ключевого слова ``...``
- 37) В Kotlin ... могут содержать объявления абстрактных методов, а также реализации методов. Что отличает их от абстрактных классов, так это то, что ... не могут хранить состояние
- 38) В Kotlin свойства, объявленные в суперклассе, могут быть переопределены в ...
- 39) Сопоставьте компонент и его функцию:
- 40) Сопоставьте разновидности класса в kotlin и их функции:
- 41) Расставьте по порядку строки объектного класса:
- 42) Базовый класс в kotlin может выглядеть как ...
- 43) Верно, что модульное тестирование ...
- 44) Класс ... - это стандартный класс, предоставляемый средами тестирования, такими как JUnit и KotlinTest Он предлагает различные методы утверждения для сравнения значений и проверки ожидаемых условий в модульных тестах.
- 45) Хорошие практики модульного тестирования с классами эквивалентности включают ...
- 46) Рекомендации по тестированию классов включают ...
- 47) Расшифровка аббревиатуры шаблона AAA – это ...
- 48) При написании параметризованных тестов, входные параметры указываются, используя аннотации ``@Value...`` или ``@Method...``
- 49) В отладчике Kotlin «Шаг в» (...) — выполнение этой строки кода с входом всех вызываемых методов





- 50 В отладчике Kotlin "Выход" (...) - выполнение кода до выхода из текущего метода
- 51 При настройке тестовых сред, советуется использовать аннотации Kotlin `@...`` или `@...Each`` (в зависимости от среды тестирования) для настройки тестовой среды перед каждым тестовым примером
- 52 ... — это отдельные тесты, охватывающие различные сценарии или функциональные возможности класса
- 53 ``assert... (expected, fact)``: проверяет, равны ли ожидаемое и фактическое значения
- 54 `Assert...(condition)``: проверяет, истинно ли условие
- 55 Тестовые примеры должны охватывать различные сценарии, включая нормальные входные ..., крайние случаи и ошибочные условия
- 56 Чистые функции легче тестировать, поскольку их ... определяются исключительно входными данными
- 57 Отладчик в среде разработки позволяет устанавливать ... в вашем коде, чтобы приостановить выполнение программы в определенных точках
- 58 ... — это группы или наборы входных данных, которые, как ожидается, будут обеспечивать сходное поведение или результаты тестируемого модуля
- 59 ... функции — это функции, которые производят один и тот же результат для одного и того же ввода без каких-либо побочных эффектов
- 60 Расставьте по порядку этапы тестирования чистых функций:
- 61 Расставьте по порядку этапы создания модульного теста в kotlin:
- 62 Сопоставьте вызовы класса-валидатора и то, какой с их помощью осуществляется функционал:
- 63 Хорошие практики модульного тестирования включают ...
- 64 "... " используется для преобразования одного значения в другое. Принимает лямбда-выражение в качестве параметра и возвращает результат этого лямбда-выражения
- 65 В Kotlin вы можете создавать собственные исключения, наследуя их от класса "..." или его подклассов





- 66 Блок "...", который является необязательным, содержит код, который будет выполнен, независимо от того, было ли сгенерировано исключение
- 67 В Kotlin вы можете наложить ограничения на поведение типа, которое может быть использовано в качестве параметра. Для этого используется ключевое слово "..."
- 68 В Kotlin ...- это общий интерфейс, представленный в стандартной библиотеке, который предоставляет функциональность заполненного элементами, упорядоченного набора данных
- 69 Если в обобщенном параметре используется "...", это означает, что мы не знаем, что это за тип, и нам это безразлично. Так можно использовать обобщенные типы, когда информация о конкретных параметрах типа недоступна или не важна.
- 70 Базовый синтаксис лямбда-выражений выглядит как ...
- 71 Обобщения позволяют определить ... с параметрами типов, что предоставляет большую гибкость при работе с коллекциями и другими типами данных
- 72 "..." позволяет выполнять некоторые операции над объектом внутри лямбда-выражения и затем возвращает результат последнего выражения
- 73 ... - это событие, которое происходит во время выполнения программы и нарушает нормальное выполнение инструкций
- 74 ... в обобщенном программировании - это концепция, которая позволяет сделать вывод о том, как один тип относится к другому исходя из их параметров
- 75 Котлин предоставляет функцию "...()", которую можно использовать для создания изменяемого "Set"
- 76 "..." определяет что обобщенный тип может быть базовым классом для заданного типа (contravariant). Это ключевое слово используется, когда тип параметра "вводится" в тело функции
- 77 ...— это функция без имени, которую можно использовать для создания функций налету
- 78 ... — это концепция, используемая в программировании, которая включает передачу функции или метода в качестве аргумента другой функции или метода
- 79 В Kotlin, ... - это принцип, позволяющий добавлять новые функции в существующие классы без их модификации





- 80) перегрузить '+', вам нужно определить функцию "... ()"
- 81) Сопоставьте тип коллекции и его описание:
- 82) Сопоставьте функцию и её назначение:
- 83) Расположите по порядку строки примера выполнения функции apply:
- 84) ... - является корректным пример обобщённого класса
- 85) ... — это процесс преобразования переменной из одного типа данных в другой.
- 86) ... разрешают выполнять различные действия в зависимости от того, достигнут ли прогресс в соблюдении условий или нет.
- 87) Расставьте шаги настройки IntelliJ IDEA в правильном порядке:
- 88) Расставьте шаги алгоритма пузырьковой сортировки в правильном порядке:
- 89) Сопоставьте тип переменной и её значение:
- 90) ... - это дополнительные модули или расширения, которые добавляют функциональность и возможности в вашу среду разработки. Они предлагают различные инструменты и функции, которые помогают упростить и ускорить процесс разработки.
- 91) ... — это блоки кода, которые выполняют определенную задачу и могут повторно использоваться в программе
- 92) Чтобы использовать функцию, требуется написать ее имя, а затем скобки и необходимые
- 93) Функции ..., принимают другие функции в качестве параметров или возвращают их
- 94) Когда вызывается ... функция, ее код копируется непосредственно в вызывающий код, что снижает накладные расходы на вызовы функций
- 95) Сортировка ... — это алгоритм сортировки сравнением на месте, который делит входной список на две части: отсортированную часть и неотсортированную часть





- 96) ... сортировка — это алгоритм сортировки «разделяй и властвуй», который выбирает опорный элемент и разделяет другие элементы на два подмассива в зависимости от того, меньше или больше они опорного элемента
- 97) Операция ... прерывает текущую итерацию цикла и переходит к следующей
- 98) Цикл ... выполняет блок кода до тех пор, пока условие истинно
- 99) ... поиск — это алгоритм поиска «разделяй и властвуй», используемый в отсортированных коллекциях
- 100) ... — это метод поиска, который использует хеш-функцию для сопоставления элементов с определенным местом в структуре данных, называемой хеш-таблицей
- 101) Какой шаг идет первым при установке IntelliJ IDEA?
- 102) В IntelliJ IDEA вы можете ...
- 103) Как задать неизменяемую переменную?
- 104) ... представляют собой шаблоны или описания объектов, которые определяют их состояние (поля) и поведение (методы)
- 105) Преимущество использования композиции состоит в том, что она дает возможность динамически менять поведение объектов за счет замены ...
- 106) Одно из преимуществ использования композиции состоит в уменьшении зависимостей между ... и повышении гибкости кода: при необходимости любую компоненту можно легко заменить на другую
- 107) ... служит связующим звеном между моделью и представлением
- 108) ООП расшифровывается как ...
- 109) ... классы используются для представления ограниченных иерархий классов, когда значение может иметь один из типов из ограниченного набора, но не может иметь никакой другой тип
- 110) В Kotlin классы и методы закрыты по умолчанию, то есть они объявляются как заключительные. Для того чтобы класс стал суперклассом и мог быть унаследован, мы должны его объявить с ключевым словом "..."





- 111 Методы и свойства, которые переопределены в подклассе, должны быть снабжены ключевым словом "..."
- 112 Интерфейс в Kotlin работает подобно классу, но интерфейс не может содержать ...: в нем могут быть объявлены абстрактные методы и реализованные методы
- 113 Чтобы класс реализовывал интерфейс, мы используем ключевое слово "..."
- 114 В Kotlin модификатор `...` используется для указания, что соответствующий член доступен везде в том же модуле
- 115 ... - это архитектурный паттерн проектирования, который разделяет приложение на три основных компонента
- 116 Когда мы объявляем класс как data, компилятор автоматически выводит следующие функции
- 117 За наследование отвечает символ
- 118 Интерфейс объявляется с использованием ключевого слова "..."
- 119 ... - это передача объекта в другой объект
- 120 `assert...(condition)`: проверяет, является ли условие ложным
- 121 `assert...(actual)`: проверяет, является ли фактическое значение нулевым
- 122 `assert...(actual)`: проверяет, не равно ли фактическое значение нулю
- 123 `assert...(expected, fact)`: проверяет, являются ли ожидаемый и фактический объекты одним и тем же экземпляром
- 124 `assert...(expected, fact)`: проверяет, не являются ли ожидаемый и фактический объекты одним и тем же экземпляром
- 125 ... — популярная среда тестирования для Java и Kotlin, предоставляющая аннотации и утверждения для написания модульных тестов
- 126 ... — это среда макетирования, которая позволяет создавать макеты объектов и моделировать поведение в модульных тестах
- 127 ... тесты позволяют нам написать один тестовый пример, который можно выполнять несколько раз с разными входными данными





- 128) Выбирая репрезентативные входные данные из каждого класса эквивалентности, мы можем обеспечить ... тестирование при минимизации избыточности
- 129) В отладчике Kotlin "Перешагнуть" (...) - выполнение строки кода без начала вызываемых методов
- 130) При настройке тестовых сред, советуется использовать аннотации Kotlin `@...` или `@...Each` (в зависимости от среды тестирования) для настройки тестовой среды перед каждым тестовым примером
- 131) ... — это отдельные тесты, охватывающие различные сценарии или функциональные возможности класса
- 132) ``assert... (expected, fact)``: проверяет, равны ли ожидаемое и фактическое значения
- 133) `Assert...(condition)``: проверяет, истинно ли условие
- 134) Расставьте по порядку этапы тестирования чистых функций:
- 135) Расставьте по порядку этапы создания модульного теста в kotlin:
- 136) Сопоставьте вызовы класса-валидатора и то, какой с их помощью осуществляется функционал:
- 137) Верно, что модульное тестирование ...
- 138) Класс ... - это стандартный класс, предоставляемый средами тестирования, такими как JUnit и KotlinTest Он предлагает различные методы утверждения для сравнения значений и проверки ожидаемых условий в модульных тестах.
- 139) Использование обобщений похоже на использование переменных. Вместо конкретного типа мы используем ... (часто обозначаемый "T"), который будет заменен на конкретный тип при использовании класса или функции
- 140) В Kotlin, как и в других языках, которые поддерживают ООП, для обработки исключений используются блоки "try", "...", "finally"
- 141) Есть несколько способов создания списка в Kotlin. Один из них - использование функции "listOf()". Это создаст ... список
- 142) Лямбда-выражения в Kotlin полезны для работы со списками. Например, можно легко отфильтровать список с помощью функции "...()"





- 143 В Kotlin "try" можно использовать как выражение, которое возвращает ...
- 144 Для перегрузки оператора "-" используется функция "... ()".
- 145 Перегруженные операторы не должны изменять свои операнды. Они должны быть определены как обычные функции или расширения с ключевым словом "..."
- 146 Оператор '<' (меньше) перегружает функция "... ()"
- 147 "... " определяет что обобщенный тип может быть производным классом для заданного типа (covariant). Это ключевое слово используется, когда тип параметра функции "выводится" из тела функции
- 148 Для создания изменяемого списка используйте функцию "...()". В таком списке вы можете добавлять, удалять и изменять элементы
- 149 Kotlin предоставляет различные функции для работы со списками, такие как ... и другие
- 150 Основное различие между "Set" и "List" заключается в том, что ...
- 151 Коллекция "Map" в Kotlin представляет собой ...
- 152 Верно, что колбэки обычно используются для того, чтобы ...
- 153 "with" похож на "run", но ...
- 154 ... — это концепция, используемая в программировании, которая включает передачу функции или метода в качестве аргумента другой функции или метода
- 155 В Kotlin, ... - это принцип, позволяющий добавлять новые функции в существующие классы без их модификации
- 156 перегрузить '+', вам нужно определить функцию "... ()"
- 157 Сопоставьте тип коллекции и его описание:
- 158 Сопоставьте функцию и её назначение:
- 159 Расположите по порядку строки примера выполнения функции apply:





- 160 ... - является корректным пример обобщённого класса
- 161 В Kotlin вы можете наложить ограничения на поведение типа, которое может быть использовано в качестве параметра. Для этого используется ключевое слово "..."
- 162 В Kotlin ...- это общий интерфейс, представленный в стандартной библиотеке, который предоставляет функциональность заполненного элементами, упорядоченного набора данных
- 163 Если в обобщенном параметре используется "...", это означает, что мы не знаем, что это за тип, и нам это безразлично. Так можно использовать обобщенные типы, когда информация о конкретных параметрах типа недоступна или не важна.
- 164 Базовый синтаксис лямбда-выражений выглядит как ...
- 165 ... - это событие, которое происходит во время выполнения программы и нарушает нормальное выполнение инструкций
- 166 ... в обобщенном программировании - это концепция, которая позволяет сделать вывод о том, как один тип относится к другому исходя из их параметров
- 167 Цикл ... позволяет перебирать элементы в заданном диапазоне или коллекции
- 168 Сортировка ... — это алгоритм сортировки сравнением на месте, который создает окончательный отсортированный массив по одному элементу за раз
- 169 Операция ..., позволяет прервать выполнение цикла и перейти к следующей инструкции после цикла
- 170 Если требуется выполнить блок кода хотя бы один раз, можно использовать цикл ...
- 171 ... — это концепция, означающая невозможность изменить состояние объекта после его создания
- 172 Kotlin использует ключевое слово `...` для объявления синглтона, которое означает, что можно создать только один экземпляр класса
- 173 Сопоставьте операторы условия с их функцией:

