

Список вопросов к экзамену по дисциплине

Программирование на языке Джава

зима 2023-2024 год

Тема 1. Особенности платформы Java. Синтаксис языка Java

1. Парадигма объектно-ориентированного программирования. Основные принципы ООП и их реализация в языке программирования Java и C++
2. Организация программы на Java. Основные структурные единицы. Процесс интерпретации и компиляции. Роль JVM
3. Структурирование Java приложения, пакеты. Уровни доступа и видимости
4. Примитивные и ссылочные типы данных. Использование механизмов автоупаковки и автораспаковки. Операция приведения типов. Понижающее и повышающее приведение.
5. Этапы проектирования, разработки и отладки ООП программ. Понятие конвенции кода языка и стиля программирования.
6. Массивы в Java. Способы создания массивов. Индексы. Размерность массивов. Доступ к элементам массива и примеры использования
7. Класс Scanner и его использование для чтения стандартного потока ввода, конструктор класса Scanner
8. Методы класса Scanner `nextLine()`, `nextInt()`, `hasNextInt()`, `hasNextLine()` и их использование для чтения ввода пользователя с клавиатуры
9. Виды типов данных в Джава. Примитивные типы данных, объявление и присваивание переменных. Константы в Джава: объявление и использование константы
10. Виды типов данных в Джава. Объектные типы данных
11. Объявление и использование бестиповых переменных в Джава
12. Объявление переменных типа класс и их инициализация
13. Массивы в Джава, как объектные типы данных, контроль доступа за пределы массива. Объявление и инициализация массивов, длина массива, получение доступа к элементу массива
14. Способы объявления массивов в Джава, использование операции `new` для выделения памяти для элементов массива. Объявление с инициализацией, объявление массива определенного размера без инициализации.
15. Инициализация полей класса и локальных переменных (отличие), инициализатор и статический инициализатор (когда вызывается).
16. Циклические конструкции в Java. Использование циклов для работы с массивами. Использование итераторов для обработки массивов. Использование итераторов для работы с коллекциями
17. Статические поля и методы. Класс Math, его основные методы.

18. Понятие перечисления. Состав и приемы использования в ООП программах на Java

Тема 2. Реализация ООП в Java.

29. Понятие класса. Определение, инициализация. Модификаторы доступа. Константы и переменные. Объявление классов.
30. Получение информации о типе. Создание экземпляров классов. Вызов методов класса Объявление класса на Джава, пример объявления
31. ООП в Java. Понятие объекта. Что представляет собой Java приложение с точки зрения ООП. Основные характеристики объектов в Java.
32. Конструкторы, назначение и использование. Конструктор с параметром, конструктор по умолчанию.
33. Конструкторы, назначение и использование. Вызов конструктора родительского класса, неявный вызов конструктора родительского класса, порядок инициализации экземпляра Java класса.
34. Использование языка UML для проектирования и документирования объектно-ориентированных программ. Основные UML диаграммы для отображения отношений между классами в ООП программах
35. Управление памятью в Java и C++, процесс освобождения памяти, занимаемой объектом. Метод finalize.
36. Понятие рекурсии, виды рекурсии и ее использование. Реализация Рекурсивных алгоритмов в ООП программах
37. Оператор new. Понятие ссылки и указателя на объект. Реализация в C++ и Java. Время жизни объекта
38. Переопределение методов в Java, абстрактные методы.
39. Преобразование ссылочных типов в Java, instanceof (экземпляр класса).
40. Графическая подсистема. Основы AWT, Swing components. Событийная модель при программировании GUI в ООП программах
41. Использование языка UML для проектирования и документирования объектно-ориентированных программ. Основные UML диаграммы для отображения отношений между классами в ООП программах
42. ООП в Java. Понятие объекта. Что представляет собой Java приложение с точки зрения ООП. Основные характеристики объектов в Java.
43. Модификатор доступа или видимости в Джава, виды и использование. Использования this для доступа к компонентам класса.
44. Чем отличаются static-метод класса от обычного метода класса. Можно ли вызвать static-метод внутри обычного метода?
45. Объявление и использование методов, объявленных с модификатором public static. Как вызвать обычный метод класса внутри static-метода?

- 46. Синтаксис объявления методов в Джава, тип возвращаемого значения, формальные параметры и аргументы. Методы с пустым списком параметров
- 47. Стандартные методы класса сеттеры и геттеры, синтаксис и их назначение?
- 48. Может ли быть поле данных класса объявлено как с модификатором `static` и `final` одновременно и что это означает?

Тема 3. Реализация наследования в программах на Джава

- 49. Наследование, виды наследования и его реализация в Java и C++
- 50. Расширение классов. Порядок создания экземпляра дочернего класса.
- 51. Наследование в Джава. Вид наследования и синтаксис Ключевое слово `extends`
- 52. Что означает перегрузка метода в Java (`overload`) и переопределение метода в Java (`override`)? В чем разница?
- 53. Абстрактные классы в Джава и абстрактные методы класса. Вложенные и анонимные классы.
- 54. Виды наследования в Джава, использование интерфейсов для реализации наследования
- 55. Что наследуется при реализации наследования в Джава (какие компоненты класса), а что нет?
- 56. К каким методам и полям базового класса производный класс имеет доступ (даже если базовый класс находится в другом пакете), а каким нет? Область видимости полей и данных из производного класса
- 57. Класс `Object`, его методы, их назначение. Иерархия классов в Java.
- 58. Наследование. Использование ключевых слов `this` и `super`. Пример использования в языках Си++ и Java
- 59. Паттерны проектирования программ. Паттерн Фабрика.
- 60. Паттерны проектирования программ. Паттерн Фабричный метод.
- 61. Расширение классов в Джава. Переопределение методов. Соккрытие полей данных.
- 62. Паттерны проектирования программ. Паттерн `Observer` и модель `MVC`

Тема 4. Полиморфизм в Джава. Работа со строками. Интерфейсы.

- 63. Интерфейсы. Общий синтаксис и расширение. Пустые интерфейсы. Реализация и применение. Сравнение с абстрактными классами.
- 64. Обработка строк в Java. Класс `StringBuffer`. Класс `StringBuilder`
- 65. Работа со строками в Java, строковый кэш. Операция конкатенации строк
- 66. Интерфейс `Comparable` и `Comparator`. Использование интерфейсных ссылок для написания обобщенных алгоритмов

67. Понятие сортировки массивов. Сортировка пузырьком. Сортировка вставками. Использование полиморфизма (ООП) для программирования алгоритмов сортировок в массивах и коллекциях
68. Понятие поиска в массивах. Последовательный поиск. Сортировка методом прямого выбора. Использование полиморфизма (ООП) для программирования алгоритмов поиска в массивах и коллекциях
69. Объявление и инициализация переменных типа String. Операция конкатенации строк и ее использование
70. При создании объектов строк с помощью класса StringBuffer, например `StringBuffer strBuffer = new StringBuffer(str)` можно ли использовать операцию конкатенации строк или необходимо использовать методы класса StringBuffer
71. Объявление и инициализация массива строк. Организация просмотра элементов массива
72. Понятие и объявление интерфейсов в Джава. Может ли один класс реализовывать несколько интерфейсов?
73. Что входит в состав интерфейса (какие компоненты может содержать интерфейс)? Может ли интерфейс наследоваться от другого интерфейса?
74. Интерфейсные ссылки и их использование в Джава
75. Интерфейс Comparable, назначение, его методы и использование в Джава
76. Какое значение возвращает вызов метода `object1.compareTo(object2)`, который сравнивает 2 объекта `obj1` и `obj2` в зависимости от объектов?

Тема 5. Основные принципы и типы исключительных ситуаций.

77. Понятие исключительной ситуации, причины возникновения, механизм обработки. Классификация исключений. Исключения, классификация и использование исключений. Генерация (порождение исключений).
78. Служебное слово `throw` и его использование при определении методов. В каком случае программа должна использовать оператор `throw`?
79. Создание собственных классов исключений
80. Блок `try/catch/finally`, его предназначение и особенности
81. В Java все исключения делятся на два основных типа. Что это за типы и какие виды ошибок ни обрабатывают?
82. Код ниже вызовет ошибку: `Exception <...> java.lang.ArrayIndexOutOfBoundsException: 4`. Что она означает?
83. Контролируемые исключения (`checked`) и неконтролируемые исключения (`unchecked`) и ошибки, которые они обрабатывают
84. Как реализуется принципы ООП в Java при создании исключений? Порядок выполнения операторов при обработке блока `try...catch`

Тема 6. Абстрактные типы данных Дженирики и использование контейнерных классов в Джава

85. Абстрактный тип данных Stack (стек) в Джава
86. Абстрактный тип данных Queue (очередь) в Джава
87. Универсальные типы или обобщенные типы данных, для чего создаются?
88. Объявление обобщённого класса коллекции с параметризованным методом для обработки массива элементов коллекции на основе цикла for each (определение общего метода для отображения элементов массива)
89. Что представляет из себя класс ArrayList и в каком случае используется
90. Обобщенное программирование. Понятие и использование дженериков в Java.
91. Параметризованные классы и методы. Их определение и использование
92. Стирание типов.
93. Понятие структуры данных список. Линейный список. Виды списков и их реализация на Java. Доступ к элементу структуры данных список. Использование списков. Трудоемкость операций со списками.
94. Односвязный и двусвязный список. Способы реализации на языке Джава

Тема 7. Java Core. Дженерики (продолжение) и использование контейнерных классов Java Framework Collection

95. Возможности Java Framework Collection. Контейнер ArrayList и его основные методы.
96. Возможности Java Framework Collection. Контейнер LinkedList и его основные методы.
97. Возможности Java Framework Collection. Интерфейс Map и его основные методы.
98. Возможности Java Framework Collection. Контейнер HashMap и его основные методы.
99. Коллекция HashMap, принципы создания и методы работы с ней
100. Использование обобщенного класса HashMap, которая реализует интерфейс Map для хранения пар ключ-значение в разработке программ.
101. Возможности Java Framework Collection. Контейнер HashSet и его основные методы.
102. Обобщенный класс HashSet класс коллекция, наследует свой функционал от класса AbstractSet, а также реализует интерфейс Set. Что он себя представляет?
103. Регулярные выражения и организация работы с ними в Java. Примеры
104. Структура коллекций в Java Collection Framework. Иерархия интерфейсов
105. Одним из ключевых методов интерфейса Collection является метод `Iterator<E> iterator()`. Что возвращает этот метод?
106. Класс Pattern и его использование
107. Класс Math и его использование

- 108. Возможности Java Framework Collection. Интерфейс Map и его основные методы
- 109. Возможности Java Framework Collection. Контейнер Hashtable его основные методы.
- 110. Возможности Java Framework Collection. Интерфейс Iterator и Itarable.
- 111. Работа с Датой и временем в Java. Примеры использования

Тема 8. Стандартные потоки ввода-вывода. Сериализация.

- 112. Класс System. Стандартные поток ввода-вывода, предоставляемые Java. Работа со стандартами потоками вывода
- 113. Перегруженные методы out.println() класса System и их использование для вывода в консоль
- 114. Класс Scanner. Ввод и вывод данных. Стандартные потоки ввода и вывода.
- 115. Использование форматированного ввода-вывода в Java. Классы для форматирования вывода.
- 116. Понятие сериализации и ее использование в ООП программах. Использование интерфейса Serializable в программах на Джава
- 117. Какие объекты можно сериализовать?
- 118. Какие методы определяет интерфейс Serializable?
- 119. Что означает понятие десериализация?
- 120. Организация работы с файлами в Джава. Класс File, определенный в пакете java.io, не работает напрямую с потоками. В чем состоит его задача?
- 121. При работе с объектом класса FileOutputStream происходит вызов метода FileOutputStream.write(), что в результате этого происходит?
- 122. Иерархия классов ввода вывода. Работа с файлами в Java. Работа с файлами. Сериализация объектов