**1 Basics of software code development**

Этот список вопросов предназначен для самостоятельного контроля глубины изучаемого материала. Попробуйте выучить материал так, чтобы вы могли ответить (хоть пару слов😊) по каждому вопросу.

1. Объясните, что имеется в виду, когда говорится: Java-язык программирования и Java-платформа.

Ответ :

1.Java как язык программирования является высокоуровневым, статически-типизированным и объектно-ориентированным.

2.Java как платформа — это программное обеспечение, представляющее собой рабочую среду для работы программ, написанных на Java (и не только). Она состоит из Java API и Java виртуальной машины (JVM).

2. Поясните, как связаны имя java-файла и классы, которые в этом файле объявляются.

Ответ:

1.Имя java-файла должно быть таким же как имя класса верхнего уровнясодержащегося в этом файле.

3. Расшифруйте аббревиатуры JVM, JDK и JRE; покажите, где “они находятся” и что собой представляют.

Ответ:

1.JVM (Java Virtual Machine) - это виртуальная машина Java

2.JRE (Java Runtime Environment) - это минимальная реализация виртуальной машины, необходимая для исполнения Java приложений, без компилятора и других средств разработки.

3.JDK (Java Development Kit) - это комплект разработчика приложений на языке Java, включающий в себя компилятор, стандартные библиотеки классов Java.

4.Находятся в корневом каталоге.Проедстовляет из себя набор библиотек.

4. Объясните, как скомпилировать и запустить приложение из командной строки, а также зачем в переменных среды окружения прописывать пути к установленному jdk.

Ответ:

1.Скомпилировать можно командной JAVAC !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

5. Перечислите атрибуты доступа, объясните их действие.

Ответ:

1.Public - публичный, общедоступный класс или член класса. Поля и методы, объявленные с модификатором public, видны другим классам из текущего пакета и из внешних пакетов.

2.Protected - такой класс или член класса доступен из любого места в текущем классе или пакете или в производных классах, даже если они находятся в других пакетах

3.Private - закрытый класс или член класса, противоположность модификатору public. Закрытый класс или член класса доступен только из кода в том же классе.

4.Default (package visible) - Отсутствие модификатора у поля или метода класса предполагает применение к нему модификатора по умолчанию. Такие поля или методы видны всем классам в текущем пакете.

6. Что такое пакеты в java-программе, что представляют собой пакеты на диске? Каково соглашение по именованию пакетов? Как создать пакет?

Ответ:

1.Пакеты в java-программе разграничивают область видимости для классов.

2.Пакеты на диске представляют из себя папки.

3.Пакеты именуются в lowerСase чтобы не создавать конфликт с названиями классов.

4.Пакеты должны именоваться в трёх уровнях:

1.Домен

2.Имя компании

3.Название проекта

Пример : by.comName.myProject.myPackage

7. Объясните, какие классы, интерфейсы, перечисления необходимо импортировать в вашу программу, как это сделать. Влияет ли импорт пакета на импорт классов и др., лежащего в подпакетах? Какой пакет в Java импортируется по умолчанию?

Ответ:

1.Классы, интерфейсы, перечисления необходимо импортировать в вашу программу те которые используются в используются в данной программе.

2. Импортировать можно с помощью команды import.

3. Импортированный пакет подтягивает все вложенные в него подпакеты и методы.

4.По умолчанию импортируется java.lang

8. Объясните различия между терминами “объект” и “ссылка на объект”.

Ответ:

1.Объект это - физическая сущность имеющая свою ячейку памяти в heap(это отдел динамической памяти Java).

Создание : Object object = new Object (); - это объект.

2. Ссылка на объект это - это переменная хранящаяся в стеке памяти(отдел памяти JVM), содержащая адрес ячейки памяти, в которой хранится объект.

Создание: Object object; - это ссылка на объект.

9. Какие примитивные типы Java вы знаете, как создать переменные примитивных типов? Объясните процедуру, по которой переменные примитивных типов передаются в методы как параметры.

Ответ:

1. Примитивные типы Java это - byte,short,int,long,float,double,boolean,char.

2. Создание переменной – ссылка на переменную : int example; - это объявление переменной,

int example = 1; - это инициализация переменной.

3.Примитивные типы данных передаются в метод по значению.Метод создаёт копию передаваемого ему значения и не как не изменяет исходное значение.

10. Каков размер примитивных типов, как размер примитивных типов зависит от разрядности платформы, что такое преобразование (приведение) типов и зачем оно необходимо? Какие примитивные типы не приводятся ни к какому другому типу.

Ответ:

1.Размер примитивных типов:

boolean = 0 byte = true or false

byte = 1 byte

short = 2 byte

int = 4 byte

long = 8 byte

char = 2 byte

float = 4 byte

double = 8 byte

2.Зависимость от разрядности платформы : Размер примитивных типов не зависит от разрядности платформы.

3.Значение приведение типов:

4.Необходимость приведение типов:

5.Неприводимые примитивные типы: boolean.

11. Объясните, что такое явное и неявное приведение типов, приведите примеры, когда такое преобразование имеет место.

Ответ:

1.Явное приведение типов : с указанием приводимого типа.

long a = 1;

int b = ( int ) a;

2.Неявное приведение типов : происходит автоматически.

long a = 1;

int b = a;

3.Преобразование имеет место : например при присвоении целого числа к числу с плавающей точкой и наоборот.

12. Что такое литералы в Java-программе, какую классификацию литералов вы знаете, как записываются литералы различных видов и типов в Java-программе?

Ответ:

1.Литералы : это явно заданные значения в коде программы.

2.Классифицируются по типам:

1.Числовые : 1,2,3….

2.Стоковые : «Hi»

3.Символьные «\n»

4.Логические true or false

13. Как осуществляется работа с типами при вычислении арифметически выражений в Java?

Ответ:

1.

14. Что такое классы-оболочки, для чего они предназначены? Объясните, что значит: объект класса оболочки – константный объект.

Ответ:

1.Wrapper предназначен для представления примитивного типа в объект.

2. Объект класса оболочки – константный объект: означает что объект является иммутабельным (неизменяемый, immutable) .

15. Объясните разницу между примитивными и ссылочными типами данных. Поясните существующие различия, при передаче параметров примитивных и ссылочных типов в методы. Объясните, как константные объекты ведут себя при передаче в метод.

Ответ:

1. Разница между примитивными и ссылочными типами данных: примитивные типы хранят в себе значение а ссылочные ссылку на объект.

2. Передача в метод осуществляется по значению для примитивного типа само его значение

для ссылочных это ссылка на объект в памяти который может быть изменён в случае наличия прав доступа.

3. Константным объектам не может быть присвоено новое значение но может быть изменено его состояние в случае наличия прав доступа.

16. Поясните, что такое автоупаковка и автораспаковка.

Ответ:

1. Автоупаковка : преобразование примитивных типов в объект.

2. Автораспаковка : преобразование объекта в примитивный тип данных.

17. Перечислите известные вам арифметические, логические и битовые операторы, определите случаи их употребления. Что такое приоритет оператора, как определить, в какой последовательности будут выполняться операции в выражении, если несколько из них имеют одинаковый приоритет.

Ответ:

1. Арифметические :

+ - сложение.

- - вычитание.

\* - умножение.

/ - деление.

% - остаток от деления.

++ - инкремент ( увеличение на 1).

-- - декремент ( уменьшение на 1).

2.Сравнение :

== - сравнение (для примитивных типов сравнивает значение для объектов сравнивает ссылки на объекты. НЕ САМИ ОБЪЕКТЫ!Для сравнения объектов используется equals

Пример :

Примитив - a == b;

Объект — a.equals( b );

!= - не равно ( true если не равно)

> - больше.

< - меньше.

>= - больше или равно.

<= - меньше или равно.

3. Логические:

! - логическое НЕ.

& - логическое И . True если два операнда true. Проверяет все оперенты.

&& - логическое И сокращённое . То же самое что & но вернёт false если первое (левый) операнд false и второй операнд проверять не будет.

| - логическое ИЛИ. True если есть хоть один true.

|| - логическое ИЛИ сокращённое. То же самое что | но вернёт true если первое (левый) операнд true и второй операнд проверять не будет.

^ - Логическое исключающее ИЛИ (XOR).Возвращает true, если операнды разные и false если оба одинаковые.

4. Побитовые :

~ - Побитовый унарный оператор NOT.

& - Побитовый И. Даёт 1 если оба бита равны 1, в остальных случаях даёт 0.

| - Побитовый ИЛИ. Даёт 1, если любой из бит или оба равны 1. 0 - если оба равны 0.

^ - Побитовый исключающее ИЛИ. Даёт 1 если один из бит равен 1. даёт 0, если оба бита равны 1 или 0.

>> - Сдвиг вправо. Значение левого операнда сдвигается вправо на количество бит правого операнда, старшие биты заполняются битом знака (0 - плюс, 1 — минус).

>>> - Сдвиг вправо с заполнением нулями. Аналогично сдвигу вправо, но старший бит всегда заполняется нулями.

<< - Сдвиг влево. Значение левого операнда сдвигается влево на количество бит правого операнда (пример 0001 << 2 = 0100 ).

18. Укажите правила выполнения операций с плавающий точкой в Java (согласно стандарту IEEE754). Как определить, что результатом вычисления стала бесконечность или нечисло?

Ответ:

1.

19. Что такое статический импорт, какие элементы можно импортировать при статическом импорте.

Ответ:

1. Статический импорт позволяет вызывать методы класса на прямую.

import static java.lang.System.out;  
  
 public class Main {  
 public static void main(String[] args) {  
 out.println("Hi");  
 }  
}

20. Объясните работу операторов if, switch, while, do-while, for, for-each. Напишите корректные примеры работы этих операторов.

Ответ:

1.Оператор if : if (ЕСЛИ условия выполняются то переходим к выполнению блока ){

out.println (“Hi“);

};

2.Оператор switch :

3.Оператор while : while ( ПОКА условия положительное блок срабатывает ){

out.println (“Hi“);

};

4.Оператор do-while : do-while ( Тоже что и while с отличием что оно сработает один раз даже если условие ложное){

out.println (“Hi“);

}

5.Оператор for : for

6.Оператор for-each :

21. Объясните работу оператора instanceof. Что будет результатом работы оператора, если слева от него будет стоять ссылка, равная null?

Ответ:

**2\_Algorithmization**

Этот список вопросов предназначен для самостоятельного контроля глубины изучаемого материала. Попробуйте выучить материал так, чтобы вы могли ответить (хоть пару слов😊) по каждому вопросу.

1. Дайте определение массиву. Как осуществляется индексация элементов массива. Как необходимо обращаться к i-му элементу массива?

Ответ:

2. Приведите способы объявления и инициализации одномерных и двумерных массивов примитивных и ссылочных типов. Укажите разницу, между массивами примитивных и ссылочных типов.

Ответ:

3. Объясните, что значит клонирование массива, как в Java можно клонировать массив, в чем состоит разница в клонировании массивов примитивных и ссылочных типов.

Ответ:

4. Объясните, что представляет собой двумерных массив в Java, что такое “рваный массив”. Как узнать количество строк и количество элементов в каждой строке для “рваного” массива?

Ответ:

5. Объясните ситуации, когда в java-коде могут возникнуть следующие исключительные ситуации java.lang.ArrayIndexOutOfBoundsException и java.lang.ArrayStoreException.

Ответ:

6. Объясните, зачем при кодировании разделять решаемую задачу на методы. Поясните, как вы понимаете выражение: “Один метод не должен выполнять две задачи”.

Ответ:

7. Объясните, как в Java передаются параметры в методы, в чем особенность передачи в метод значения примитивного типа, а в чем ссылочного.

Ответ:

8. Объясните, как в метод передать массив. И как массив вернуть из метода. Можно ли в методе изменить размер переданного массива.

Ответ:

9. Поясните, что означает выражение ‘вернуть значение из метода’. Как можно вернуть значение из метода. Есть ли разница при возврате значений примитивного и ссылочного типов.

Ответ:

10. Перечислите известные вам алгоритмы сортировки значений, приведите код, реализующий это алгоритмы.

Ответ:

**3 Strings and basics of text processing**

Этот список вопросов предназначен для самостоятельного контроля глубины изучаемого материала. Попробуйте выучить материал так, чтобы вы могли ответить (хоть пару слов😊) по каждому вопросу.

**Strings**

1. Как создать объект класса String, какие конструкторы класса String вы знаете? Что такое строковый литерал? Объясните, что значит “упрощенное создание объекта String”?

Ответ:

2. Можно ли изменить состояние объекта типа String? Что происходит при попытке изменения состояния объекта типа String? Можно ли наследоваться от класса String? Как вы думаете, почему строковые объекты immutable?

Ответ:

3. Объясните, что такое кодировка? Какие кодировки вы знаете? Как создать строки в различной кодировке?

Ответ:

4. Что такое пул литералов? Как строки заносятся в пул литералов? Как занести строку в пул литералов и как получить ссылку на строку, хранящуюся в пуле литералов? Где хранится(в каком типе памяти) пул литералов в Java 1.6 и Java 1.7?

Ответ:

5. В чем отличие объектов классов StringBuilder и StringBuffer от объектов класса String? Какой из этих классов потокобезопасный? Как необходимо сравнивать на равенство объекты классов StringBuilder и StringBuffer и почему?

Ответ:

6. Что такое Unicode?

Ответ:

7. Какие методы класса String используются для работы с кодовыми точками? Как вы думаете, когда следует их использовать?

Ответ:

**Regular Expressions**

1. Расскажите, что представляет собой регулярное выражение? Что такое метасимволы регулярного выражения? Какие вы знаете классы символов регулярных выражений? Что такое квантификаторы? Какие логические операторы регулярных выражений вы знаете? Что значит “якорь” для регулярного выражения?

Ответ:

2. Какие java-классы работают с регулярными выражениями? В каком пакете они расположены? Приведите пример анализа текста с помощью регулярного выражения и поясните код примера.

Ответ:

3. Что такое группы в регулярных выражениях? Как нумеруются группы? Что представляет собой группа номер 0(ноль)? Приведите пример с использованием групп регулярного выражения.

Ответ:

**4. Programming with classes**

Этот список вопросов предназначен для самостоятельного контроля глубины изучаемого материала. Попробуйте выучить материал так, чтобы вы могли ответить (хоть пару слов😊) по каждому вопросу.

1. Опишите процедуру инициализации полей класса и полей экземпляра класса. Когда инициализируются поля класса, а когда – поля экземпляров класса. Какие значения присваиваются полям по умолчанию? Где еще в классе полям могут быть присвоены начальные значения?

Ответ:

2. Дайте определение перегрузке методов. Как вы думаете, чем удобна перегрузка методов? Укажите, какие методы могут перегружаться, и какими методами они могут быть перегружены? Можно ли перегрузить методы в базовом и производном классах? Можно ли private метод базового класса перегрузить public методов производного? Можно ли перегрузить конструкторы, и можно ли при перегрузке конструкторов менять атрибуты доступа у конструкторов?

Ответ:

3. Объясните, что такое раннее и позднее связывание? Перегрузка – это раннее или позднее связывание? Объясните правила, которым следует компилятор при разрешении перегрузки; в том числе, если методы перегружаются примитивными типами, между которыми возможно неявное приведение или ссылочными типами, состоящими в иерархической связи.

Ответ:

4. Объясните, как вы понимаете, что такое неявная ссылка this? В каких методах эта ссылка присутствует, а в каких – нет, и почему?

Ответ:

5. Что такое финальные поля, какие поля можно объявить со спецификатором final? Где можно инициализировать финальные поля?

Ответ:

6. Что такое статические поля, статические финальные поля и статические методы. К чему имеют доступ статические методы? Можно ли перегрузить и переопределить статические методы? Наследуются ли статические методы?

Ответ:

7. Что такое логические и статические блоки инициализации? Сколько их может быть в классе, в каком порядке они могут быть размещены и в каком порядке вызываются?

Ответ:

8. Что представляют собой методы с переменным числом параметров, как передаются параметры в такие методы и что представляет собой такой параметр в методе? Как осуществляется выбор подходящего метода, при использовании перегрузки для методов с переменным числом параметров?

Ответ:

9. Чем является класс Object? Перечислите известные вам методы класса Object, укажите их назначение.

Ответ:

10. Что такое хэш-значение? Объясните, почему два разных объекта могут сгенерировать одинаковые хэш-коды?

Ответ:

11. Что такое объект класса Class? Чем использование метода getClass() и последующего сравнения возвращенного значения с Type.class отличается от использования оператора instanceof?

Ответ:

12. Укажите правила переопределения методов equals(), hashCode() и toString().

Ответ:

**5. Basics of OOP**

Этот список вопросов предназначен для самостоятельного контроля глубины изучаемого материала. Попробуйте выучить материал так, чтобы вы могли ответить (хоть пару слов😊) по каждому вопросу.

**OOP, Classes and Objects, Inheritance, Class Object, Interfaces**

1. Дайте развернутое объяснение трем концепциям ООП.

Ответ:

2. Опишите процедуру инициализации полей класса и полей экземпляра класса. Когда инициализируются поля класса, а когда – поля экземпляров класса. Какие значения присваиваются полям по умолчанию? Где еще в классе полям могут быть присвоены начальные значения?

Ответ:

3. Приведите правила, которым должен следовать компонент java-bean.

Ответ:

4. Дайте определение перегрузке методов. Как вы думаете, чем удобна перегрузка методов? Укажите, какие методы могут перегружаться, и какими методами они могут быть перегружены? Можно ли перегрузить методы в базовом и производном классах? Можно ли private метод базового класса перегрузить public методов производного? Можно ли перегрузить конструкторы, и можно ли при перегрузке конструкторов менять атрибуты доступа у конструкторов?

Ответ:

5. Объясните, что такое раннее и позднее связывание? Перегрузка – это раннее или позднее связывание? Объясните правила, которым следует компилятор при разрешении перегрузки; в том числе, если методы перегружаются примитивными типами, между которыми возможно неявное приведение или ссылочными типами, состоящими в иерархической связи.

Ответ:

6. Объясните, как вы понимаете, что такое неявная ссылка this? В каких методах эта ссылка присутствует, а в каких – нет, и почему?

Ответ:

7. Что такое финальные поля, какие поля можно объявить со спецификатором final? Где можно инициализировать финальные поля?

Ответ:

8. Что такое статические поля, статические финальные поля и статические методы. К чему имеют доступ статические методы? Можно ли перегрузить и переопределить статические методы? Наследуются ли статические методы?

Ответ:

9. Что такое логические и статические блоки инициализации? Сколько их может быть в классе, в каком порядке они могут быть размещены и в каком порядке вызываются?

Ответ:

10. Что представляют собой методы с переменным числом параметров, как передаются параметры в такие методы и что представляет собой такой параметр в методе? Как осуществляется выбор подходящего метода, при использовании перегрузки для методов с переменным числом параметров?

Ответ:

11. Чем является класс Object? Перечислите известные вам методы класса Object, укажите их назначение.

Ответ:

12. Что такое хэш-значение? Объясните, почему два разных объекта могут сгенерировать одинаковые хэш-коды?

Ответ:

13. Как вы думаете, для чего используется наследование классов в java-программе? Приведите пример наследования. Как вы думаете, поля и методы, помеченными модификатором доступа private, наследуются?

Ответ:

14. Укажите, как вызываются конструкторы при создании объекта производного класса? Что в конструкторе класса делает оператор super()? Возможно ли в одном конструкторе использовать операторы super() и this()?

Ответ:

15. Объясните, как вы понимаете утверждения: “ссылка базового класса может ссылаться на объекты своих производных типов” и “объект производного класса может быть использован везде, где ожидается объект его базового типа”. Верно ли обратное и почему?

Ответ:

16. Что такое переопределение методов? Как вы думаете, зачем они нужны? Можно ли менять возвращаемый тип при переопределении методов? Можно ли менять атрибуты доступа при переопределении методов? Можно ли переопределить методы в рамках одного класса?

Ответ:

17. Определите правило вызова переопределенных методов. Можно ли статические методы переопределить нестатическими и наоборот?

Ответ:

18. Какие свойства имеют финальные методы и финальные классы? Как вы думаете, зачем их использовать?

Ответ:

19. Укажите правила приведения типов при наследовании. Напишите примеры явного и неявного преобразования ссылочных типов. Объясните, какие ошибки могут возникать при явном преобразовании ссылочных типов.

Ответ:

20. Что такое объект класса Class? Чем использование метода getClass() и последующего сравнения возвращенного значения с Type.class отличается от использования оператора instanceof?

Ответ:

21. Укажите правила переопределения методов equals(), hashCode() и toString().

Ответ:

22. Что такое абстрактные классы и методы? Зачем они нужны? Бывают ли случаи, когда абстрактные методы содержат тело? Можно ли в абстрактных классах определять конструкторы? Могут ли абстрактные классы содержать неабстрактные методы? Можно ли от абстрактных классов создавать объекты и почему?

Ответ:

23. Что такое интерфейсы? Как определить и реализовать интерфейс в java-программе? Укажите спецификаторы, которые приобретают методы и поля, определенные в интерфейсе. Можно ли описывать в интерфейсе конструкторы и создавать объекты? Можно ли создавать интерфейсные ссылки и если да, то на какие объекты они могут ссылаться?

Ответ:

24. Для чего служит интерфейс Clonable? Как правильно переопределить метод clone() класса Object, для того, что объект мог создавать свои адекватные копии?

Ответ:

25. Для чего служат интерфейсы Comparable и Comparator? В каких случаях предпочтительнее использовать первый, а когда – второй? Как их реализовать и использовать?

Ответ:

**Generic classes and Interfaces, Enums**

1. Что такое перечисления в Java. Как объявить перечисление? Чем являются элементы перечислений? Кто и когда создает экземпляры перечислений?

Ответ:

2. Можно ли самостоятельно создать экземпляр перечисления? А ссылку типа перечисления? Как сравнить, что в двух переменных содержится один и тот же элемент перечисления и почему именно так?

Ответ:

3. Что такое анонимные классы?

Ответ:

4. Что такое параметризованные классы? Для чего они необходимы? Приведите пример параметризованного класса и пример создания объекта параметризованного класса? Объясните, ссылки какого типа могут ссылаться на объекты параметризованных классов? Можно ли создать объект, параметризовав его примитивным типом данных?

Ответ:

**Exceptions and Errors**

1. Что для программы является исключительной ситуацией? Какие способы обработки ошибок в программах вы знаете?

Ответ:

2. Что такое исключение для Java-программы? Что значит “программа выбросила исключение”? Опишите ситуации, когда исключения выбрасываются виртуальной машиной(автоматически), и когда необходимо их выбрасывать вручную?

Ответ:

3. Приведите иерархию классов-исключений, делящую исключения на проверяемые и непроверяемые. В чем особенности проверяемых и непроверяемых исключений?

Ответ:

4. Объясните работу оператора try-catch-finally. Когда данный оператор следует использовать? Сколько блоков catch может соответствовать одному блоку try? Можно ли вкладывать блоки try друг в друга, можно ли вложить блок try в catch или finally? Как происходит обработка исключений, выброшенных внутренним блоком try, если среди его блоков catch нет подходящего? Что называют стеком операторов try? Как работает блок try с ресурсами.

Ответ:

5. Укажите правило расположения блоков catch в зависимости от типов перехватываемых исключений. Может ли перехваченной исключение быть сгенерировано снова, и, если да, то как и кто в этом случае будет обрабатывать повторно сгенерированное исключение? Может ли блок catch выбрасывать иные исключения, и если да, то опишите ситуацию, когда это может быть необходимо.

Ответ:

6. Когда происходит вызов блока finally? Существуют ли ситуации, когда блок finally не будет вызван? Может ли блок finally выбрасывать исключений? Может ли блок finally выполнится дважды?

Ответ: