Сlasses and Objects

**1. Дайте определение таким понятиям как ”класс” и “объект”. Приведите примеры объявления класса и создания объекта класса. Какие спецификаторы можно использовать при объявлении класса.**

**2. Как вы определяете, какие поля и методы необходимо определить в классе, приведите пример. Какие спецификаторы можно использовать с полями, а какие с методами (и что они значат)?**

|  |  |  |  |
| --- | --- | --- | --- |
| **Модификатор** | **Класс** | **Поле класса** | **Метод** |
| **Доступа** | **+** | **+** | **+** |
| **Static** | **+** | **+** | **+** |
| **final** | **+** | **+** | **+** |
| **synchronized** | **-** | **-** | **+** |
| **transient** | **-** | **+** | **-** |
| **volatile** | **-** | **+** | **-** |
| **Abstract** | **+** | **-** | **+** |

Public class Cat {}

Private static class Cat{} – создание классов.

New Cat(); - создание объекта класса Cat.

**3. Что такое конструктор? Как вы отличите конструктор от любого другого метода? Сколько конструкторов может быть в классе? Что такое конструктор по умолчанию, может ли в классе совсем не быть конструкторов? Объясните, что делает оператор this() в конструкторе?**

Конструктор - это специальный метод, который вызывается при создании нового объекта.

Конструкторов может быть сколько угодно. К-р по умолчанию = конструктору без параметров. Если переопределен другой конструктор, то по умолчанию не работает, нужно указывать явно. Не быть конструктора не может, но может быть приватный (паттерн синглтон)

This –означает обращаение к этому объекту, используется в след случаях:

Public User(String name, int age) {

This.name = name;

This.age = age;

}

**4. 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?**

Поля класса инициализуруются при запуске JVM, если поле – статическое; и при создании экземпляра класса, если поле не статическое.

Каждый тип имеет свое значение по умолчанию (Boolean = false, int=0; object=null и т.д.). Значение по умолчанию устанавливается при инициализации.

Статические переменные хранятся в особом блоке non-heap memory – metaspace.

Обычные переменные экземпляров класса хранятся в heap.

Иным способом можно инициализировать static блоками, которые вызываются перед вызовом даже конструктора класса.

**5. 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?**

В программе мы можем использовать методы с одним и тем же именем, но с разными типами и/или количеством параметров. Такой механизм называется **перегрузкой методов** (method overloading).

Стоит отметить, что на перегрузку методов влияет количество и типы параметров, но тип возвращаемого значения не играет никакой роли.

Перегружать и переопределять методы можно и в базовом и в производном классе. В производном можно перегружать/определять методы базового класса.

Перегружать конструктора также можно, менять артибуты доступа тоже можно.

**6. 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.**

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

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

9. 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?

10. 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?

11. 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?