**Тест №3. Объектно-ориентированные принципы Java**

1. Конструктор класса – это…

**- специальный метод, который не имеет возвращаемого значения и имеет то же самое название, что и класс;**

- метод, который имеет возвращаемое значение и имеет то же самое название, что и класс**;**

- метод, который не имеет возвращаемого значения и имя конструктора не совпадает с именем класса**;**

**- специальный метод, который вызывается при создании нового объекта.** Принципы объектно-ориентированного программирования. Определение методов.

1. Частный доступ, который доступен только из методов данного класса – это…

- public

**- private**

- protected

- friendly Объявление классов. Частный доступ.

1. Заполните пропуски:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Описание модификаторов доступа** | **private** | **friendly** | **protected** | **public** |
| Тот же самый класс | Да | 2. | Да | Да |
| Тот же самый пакет, другой класс | Нет | Да | Да | 4. |
| Тот же самый пакет подкласс | 1. | Да | Да | Да |
| Другой пакет другой класс | Нет | Нет | Нет | Да |
| Другой пакет подкласс | Нет | Нет | 3. | Да |

- 1 - нет; 2 - нет; 3 - да; 4 - нет;

- 1 - нет; 2 - да; 3 - нет; 4 - да;

- 1 - да; 2 - да; 3 - да; 4 - да;

**- 1 - нет; 2 - да; 3 - да; 4 - да.** Объявление классов. Модификаторы доступа.

1. Выберите верные утверждения о наследовании в Java:

**- суть наследования состоит в создании производного класса, который имеет возможности базового класса плюс свои собственные;**

- наследование не поддерживает концепцию иерархии классов;

- при использовании наследования можно описать объект путём определения того класса (или классов), от которого он наследуется. Принципы объектно-ориентированного программирования. Наследование классов в Java.

1. Объявление класса с данным спецификатором означает, что его нельзя расширить и породить от него подкласс:

- protected;

**- final;**

- void;

- new. Классы. Ключевое слово final.

1. Результатом выполнения программы

class FinalDemo {

public static void main (String []args) {

final int noChange = 20;

noChange = 30;

System.out.println(“Значение noChange = ” + noChange);

}

}

является:

- 20;

- 30;

- 50;

**- сообщение об ошибке**. Объявление классов. Выполнение программы.

1. Выберите утверждения, характеризующие перегруженные методы:

**- это методы, которые находятся в одном классе;**

**- это методы, которые имеют одинаковое имя, но различные списки параметров;**

- это методы, которые находятся в разных классах с одинаковым именем, но с различными списками параметров**.** Принципы объектно-ориентированного программирования. Перегруженные методы.

1. Выберите утверждения, характеризующие переопределенные методы:

**- это методы, которые находятся в суперклассе и в подклассе;**

- это методы, которые имеют одинаковое имя, но различные списки параметров;

**- это методы, которые имеют одинаковую сигнатуру;**

– все варианты. Принципы объектно-ориентированного программирования. Переопределенные методы.

1. Выберите верные утверждения:

**- абстрактным называется класс, который содержит хотя бы один абстрактный метод;**

**- в определении абстрактного класса нужно указывать ключевое слово abstract;**

**- с помощью абстрактных классов можно объявлять классы, реализованные лишь частично, поручив полную реализацию классам-расширениям;**

**-** ничего из перечисленного. Принципы объектно-ориентированного программирования. Абстрактные классы.

1. Заполните пропуски:

Класс может \_\_\_\_\_1\_\_\_\_\_\_ только один базовый класс, и\_\_\_2\_\_\_\_\_\_ несколько интерфейсов.

**- 1-расширять (extends); 2-раскрывать (implements);**

- 1-раскрывать (implements); 2-расширять (extends);

- 1-раскрывать (implements); 2-раскрывать (implements);

- 1-расширять (extends); 2-расширять (extends). Принципы объектно-ориентированного программирования. Наследование.