**1.Перечислите состав класса.**

Поля, методы, конструкторы, логические блоки(инициализаторы), внутренние классы

**2. Где и как могут использоваться [static] [abstract] [final] в контексте**

**класса?**

Перед именем класса, методами

**4. Для чего используется модификатор native?**

сообщает компилятору, что метод реализован в другом месте(на другом языке, например на с++)

*public native int loadFreeDiskSpace();*

можно переопределять обычными методами в подклассах

**5. Что такое логический и статический блок?**

Логический блок:

{

        name = "Undefined";

        age = 18;

}

Статический блок: Статические блоки и статическая инициализация используются для инициализации статических полей в Java. Мы объявляем статический блок в тот момент, когда хотим проинициализировать статические поля в классе.Статический блок выполняется только один раз, при загрузке класса загрузчиком.Статический блок выполняется до вызова конструктора.Также можно сказать что статические поля и методы принадлежат классу, нестатические — объекту.

static

{

 int a = 3;

            int b = 10;

            if (b < a)

                System.out.println(a);

            else

                System.out.println(b);

}

**6. Определите параметризованный**(generic) **класс.**

**public** **class** Subject <T1, T2>

{

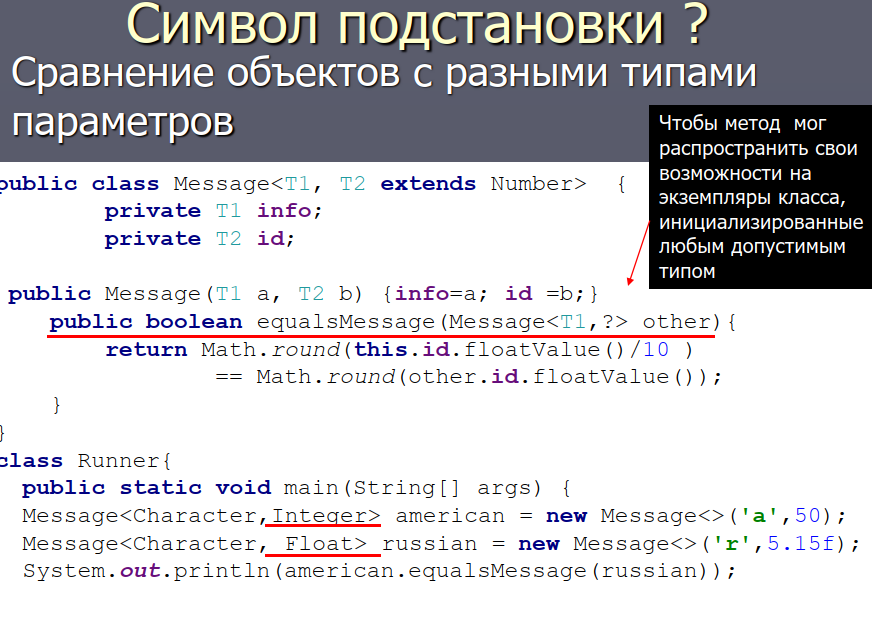
**private** T1 name;

**private** T2 id;

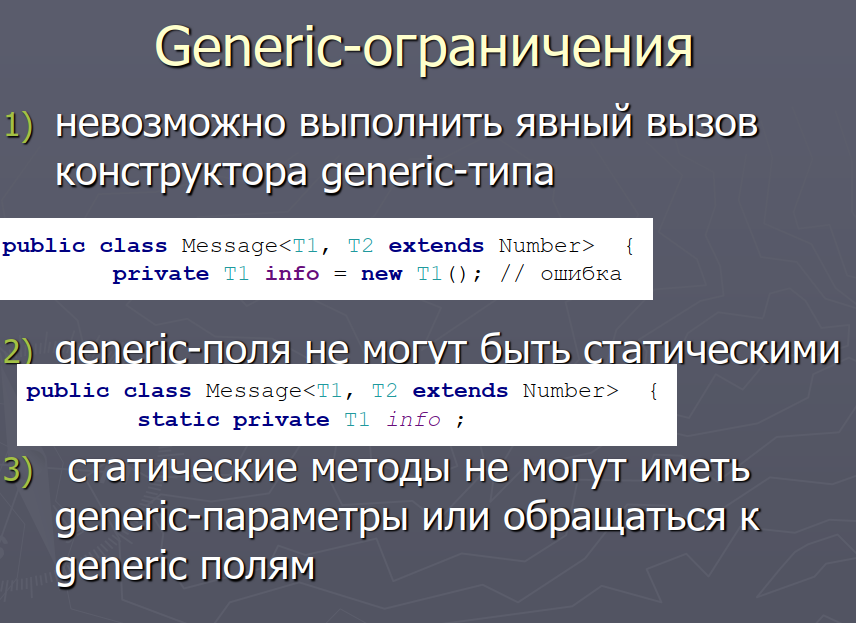
**public** Subject() {

}

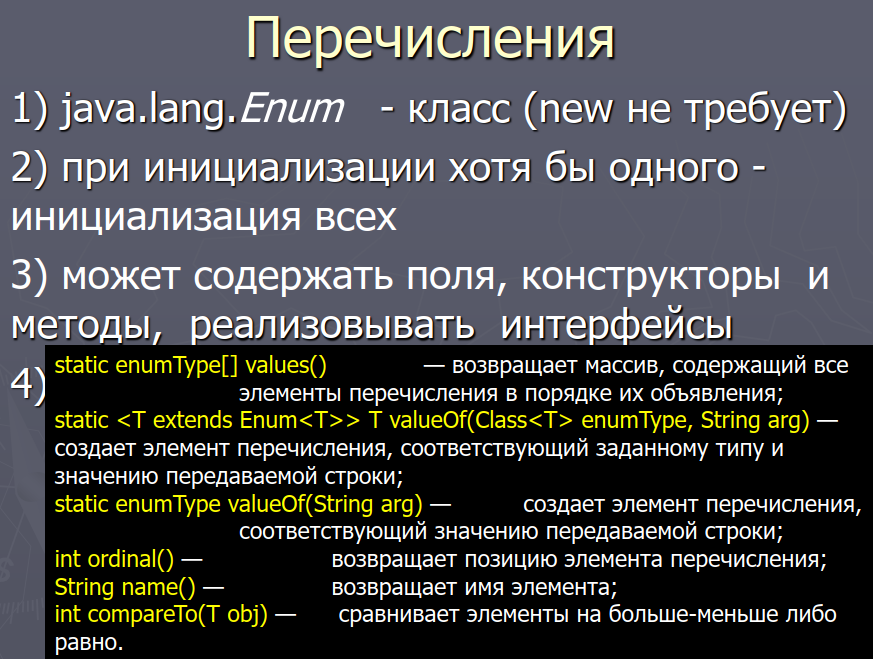
**7. Как используется метасимвол «?»**



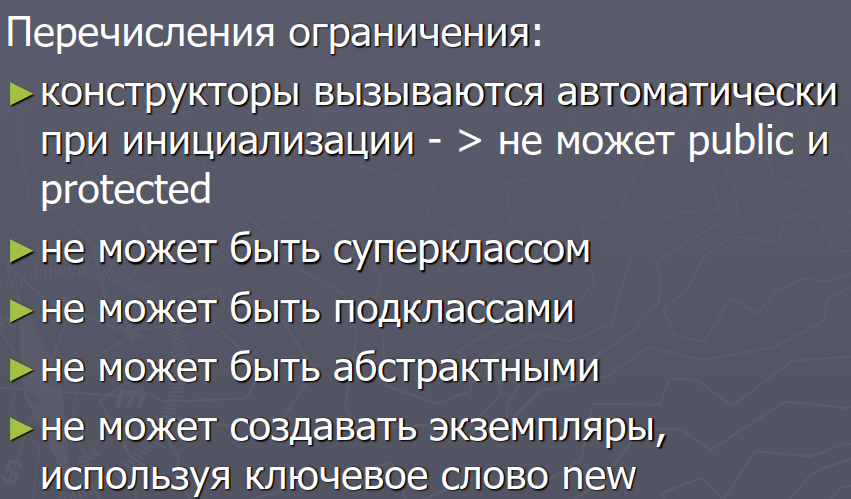
**8. Какие существуют generic-ограничения?**



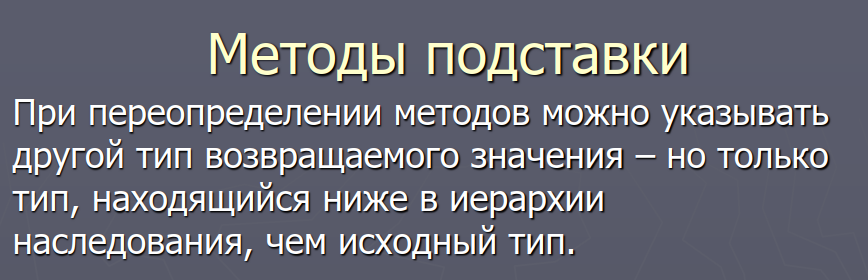
**9. Что могут содержать перечисления? Приведите пример**

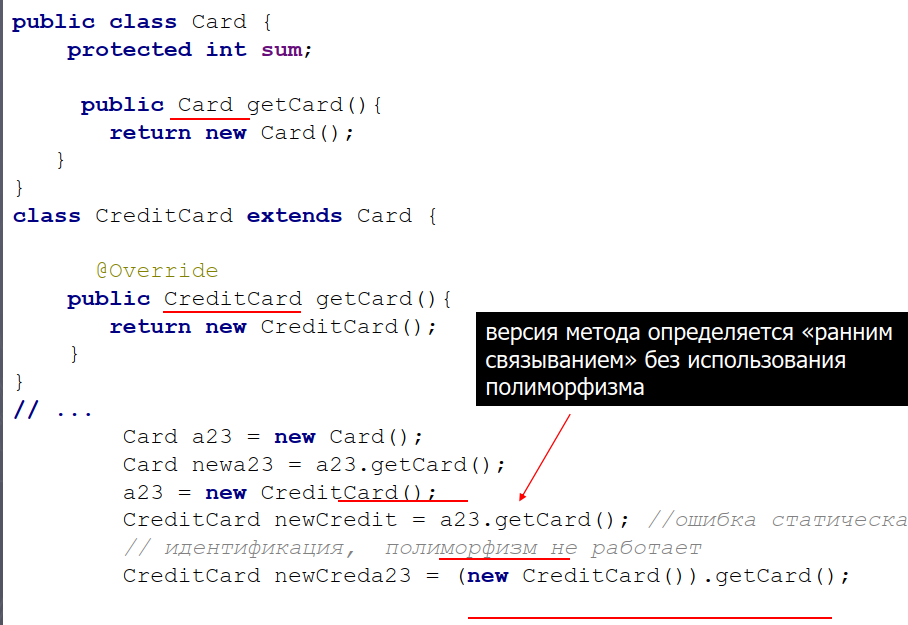


**10. Какие существуют ограничения для перечислений?**



**11. Что такое методы подставки?**





**12. Состав класса Object.**

1)toString

2)Метод hashCode,

### 3)Получение типа объекта и метод getClass

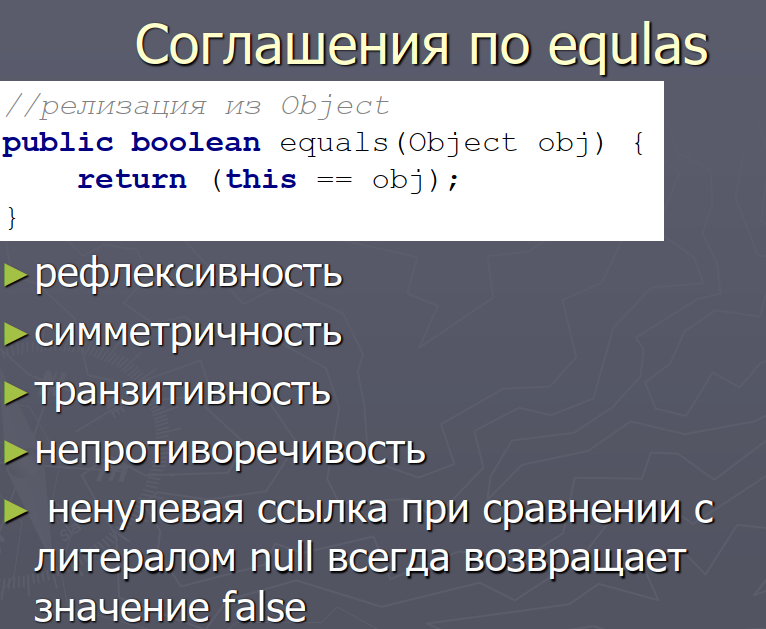
Метод getClass позволяет получить тип данного объекта:

|  |  |
| --- | --- |
| 1  2 | Person tom = new Person("Tom");  System.out.println(tom.getClass()); // class Person |

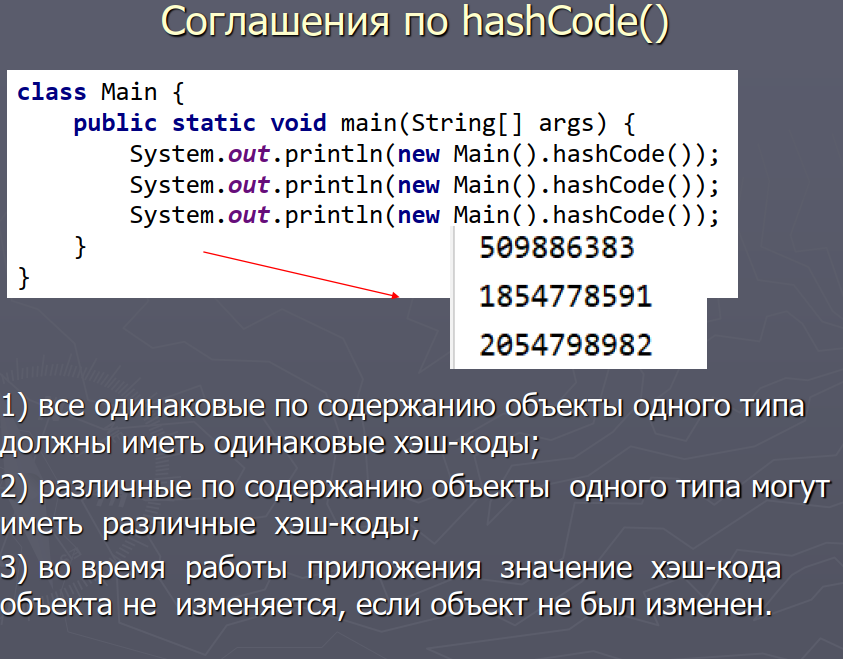
4) Метод equals

Метод equals сравнивает два объекта на равенство. Метод equals принимает в качестве параметр объект любого типа, который мы затем приводим к текущему, если они являются объектами одного класса.

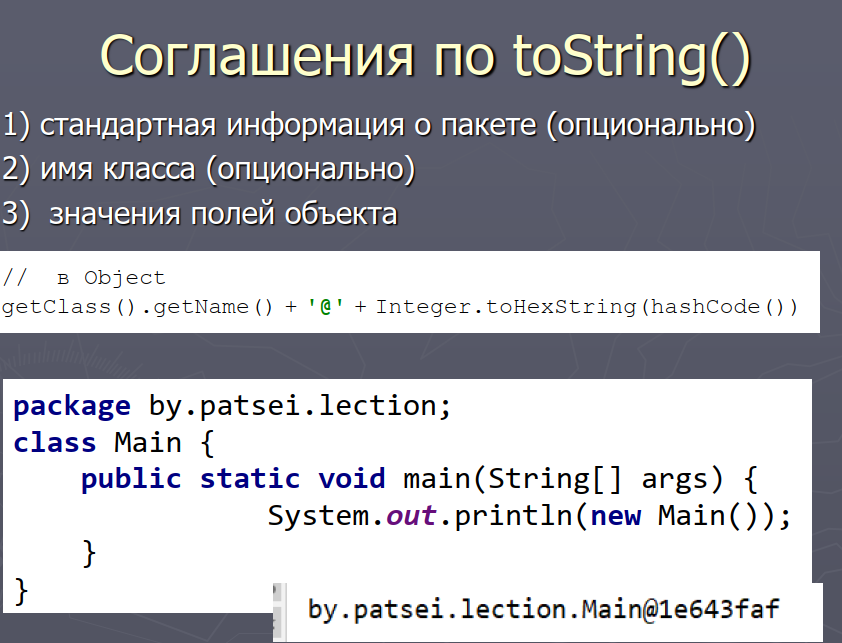
**13. Перечислите соглашения по equlas.**



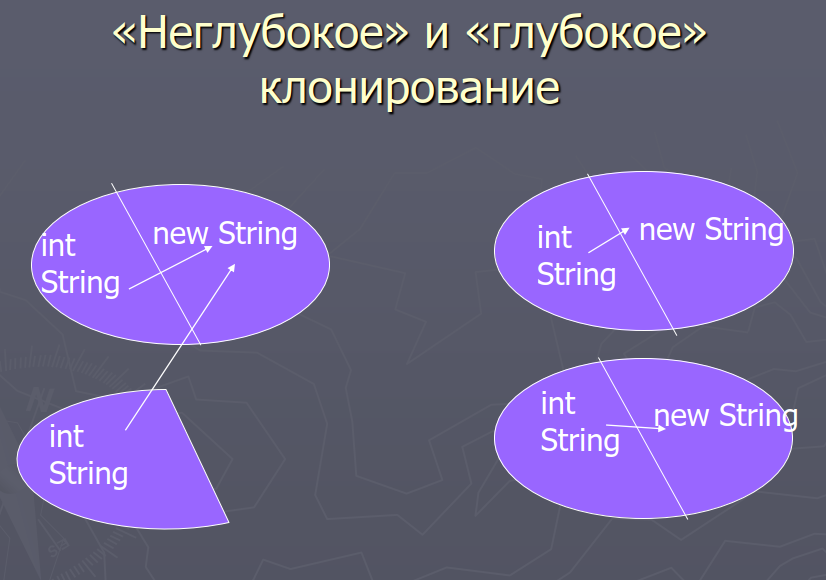
**14. Перечислите соглашения по hashCode() .**



**15. Перечислите соглашения по toString().**

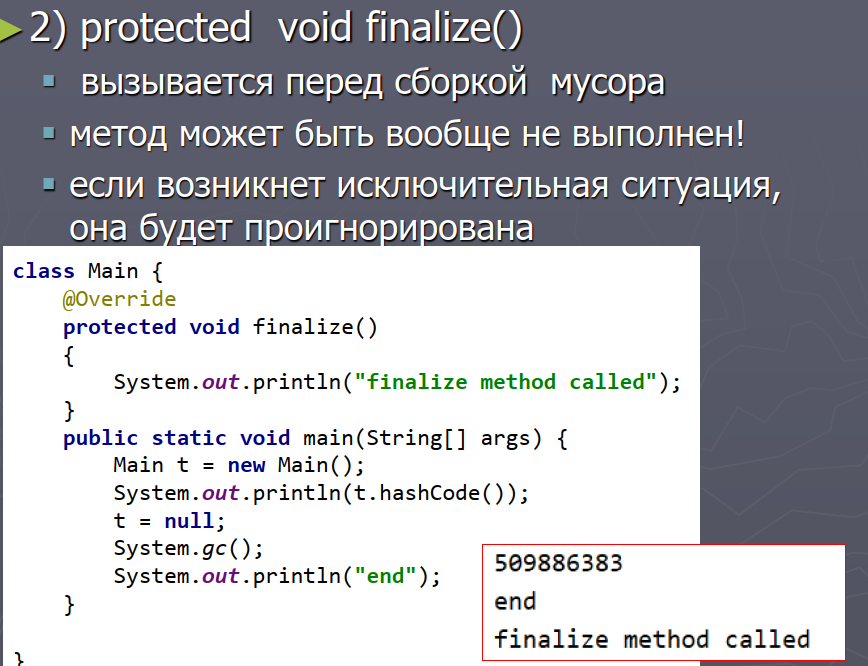


**16. Поясните разницу между «неглубким» и «глубоким» клонированием? Приведите пример.**

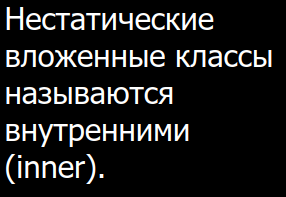


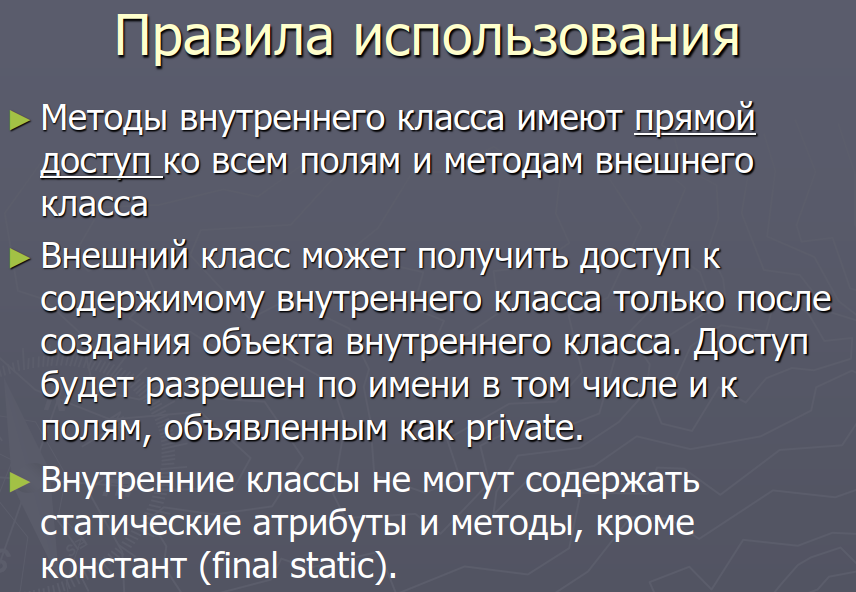
Глубокое- по значение, неглубокое- по ссылке

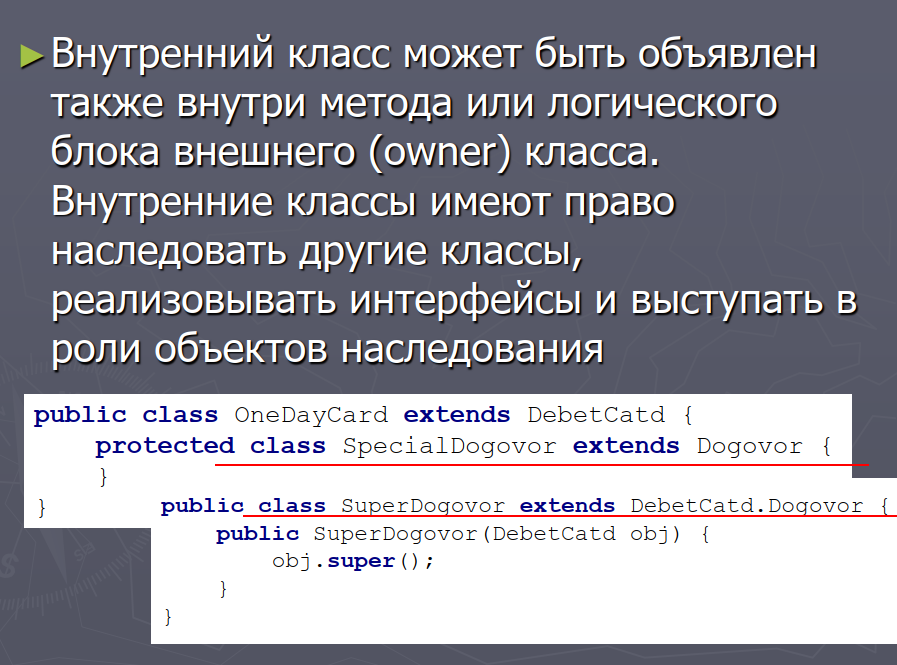
**17. Как можно использовать метод void finalize()?**



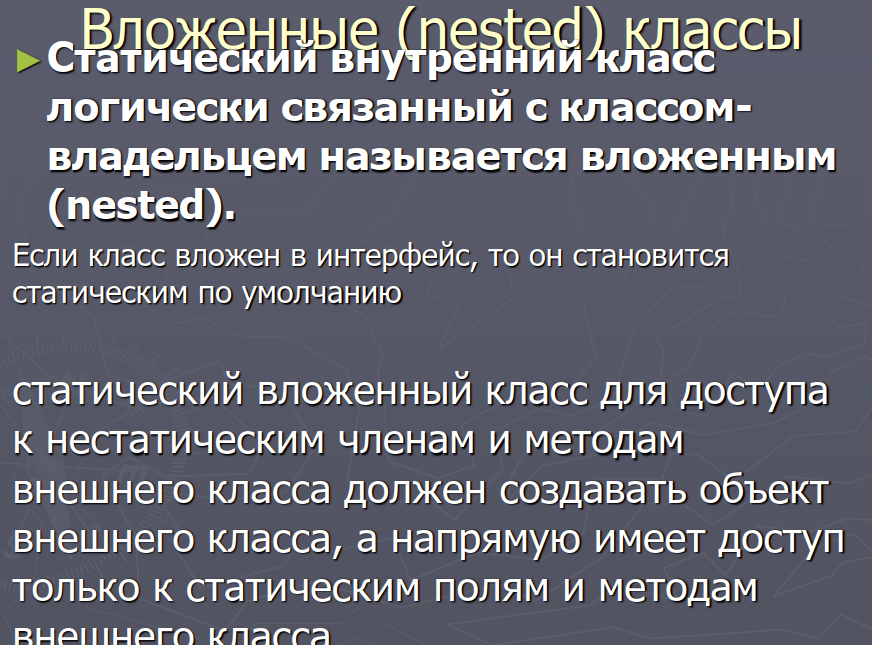
**18. Что такое внутренние классы (inner)? Привила использования.**

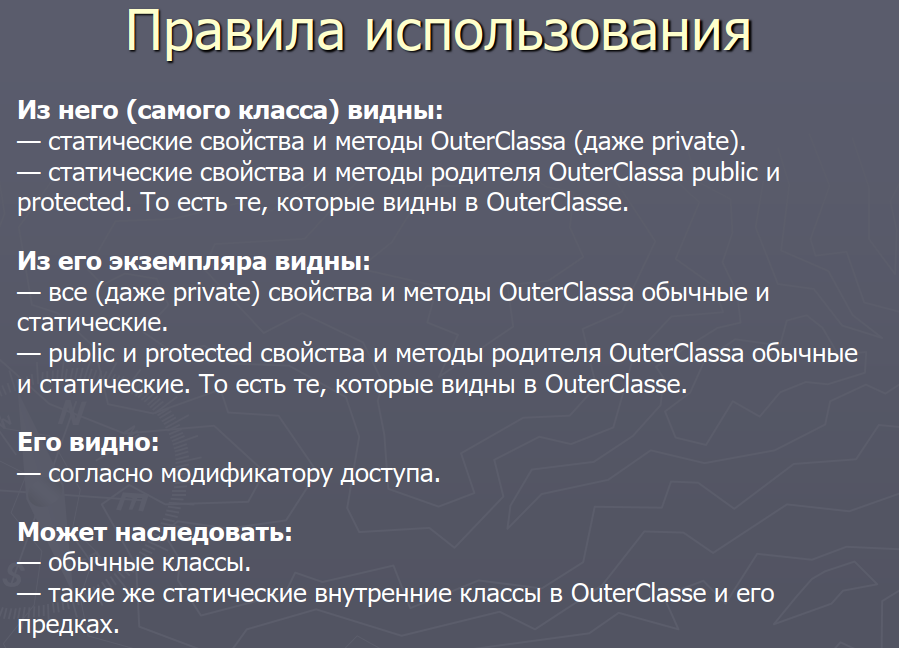


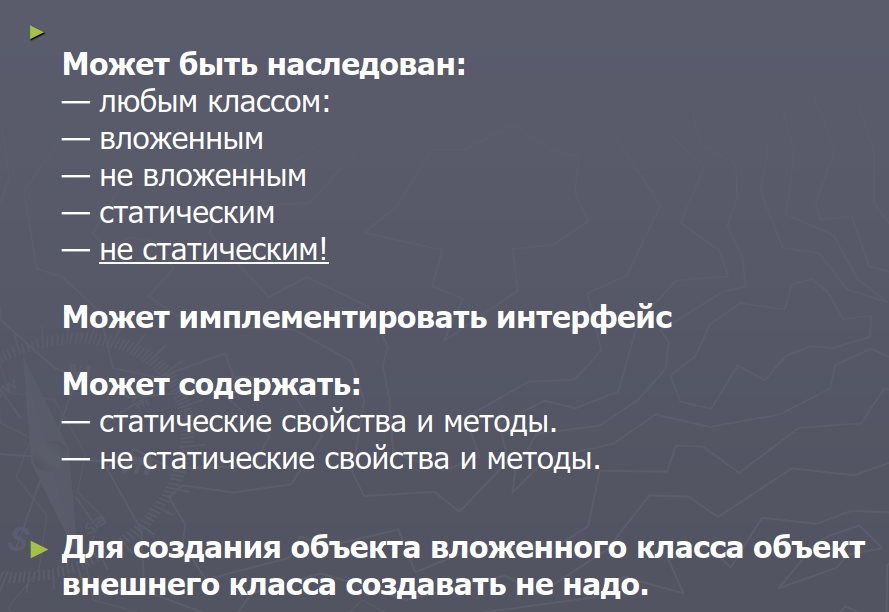




**19. Что такое вложенные (nested) классы? Привила использования.**







**20. Что такое анонимные (anonymous) классы?**

