

A background image showing a complex network of interconnected nodes and lines, resembling a web or a neural network, rendered in shades of blue.

# Лекция 15. Встроенные классы

**NetCracker**®

- **статические** (static) вложенные классы и интерфейсы;
- **нестатические** (non-static) или внутренние (inner) классы
  - *внутренние классы общего вида*
  - *локальные (local) внутренние классы*
  - *анонимные (anonymous) внутренние классы*

## Объявление класса внутри другого класса

```
class ИмяВнешнегоКласса{  
    тело внешнего класса  
        static class ИмяВложенногоКласса{  
            тело вложенного класса  
        }  
    продолжение тела внешнего класса  
}
```

Экземпляры вложенного класса, а также *методы класса* и *поля класса* получают в имени квалификатор – *имя класса* верхнего уровня.

*Доступ к полю*

**ИмяВнешнегоКласса.ИмяВложенногоКласса.имяПоля,**

*Обращение к методу класса*

**ИмяВнешнегоКласса.ИмяВложенногоКласса.имяМетода(список параметров).**

# Компиляция вложенных классов

## Компиляция вложенных классов

создаются самостоятельные классы `.class`, имеющие имя `имяВнешнегоКласса$имяВложенногоКласса.class`.

Точно такое же имя выдается в методах

*объектВложенногоКласса* `.toString()` или  
*объектВложенногоКласса* `.getClass().getName()`.

А *объектВложенногоКласса* `.getClass().getCanonicalName()` возвращает имя вложенного класса через точку.

Внутренний *класс* задается так же, как вложенный, но только без модификатора *static* перед именем этого класса:

```
class ИмяВнешнегоКласса{  
    тело внешнего класса  
        class ИмяВнутреннегоКласса{  
            тело внутреннего класса  
        }  
    продолжение тела внешнего класса }
```

Для внутренних классов экземпляры создаются через *имя объекта внешнего класса*, что принципиально отличает их от обычных и вложенных классов.

**ИмяВнешнегоКласса** **имяОбъекта** = new **ИмяВнешнегоКласса**  
(параметры);

**ИмяВнешнегоКласса.ИмяВнутреннегоКласса** **имя1** = **имяОбъекта.new**  
**ИмяВнутреннегоКласса**(параметры);

**ИмяВнешнегоКласса.ИмяВнутреннегоКласса** **имя2** = **имяОбъекта.new**  
**ИмяВнутреннегоКласса**(параметры);

Из внутреннего класса можно обратиться к объекту внешнего класса:

**ИмяВнешнегоКласса.this**

Доступ к полю или методу внешнего объекта

**ИмяВнешнегоКласса.this.имяПоля**

**ИмяВнешнегоКласса.this.имяМетода**(список параметров)

**Локальные классы** – классы, в которых область существования их и их экземпляров ограничена тем блоком, в котором они заданы

```
class LocalClass1 {  
    public LocalClass1(){  
        System.out.println("LocalClass1 object created");  
    }  
};  
  
LocalClass1 local1=new LocalClass1();
```

*Анонимный (безымянный) класс* объявляется без задания имени класса и переменных данного безымянного типа – задается только *конструктор* класса вместе с его реализацией.

```
new ИмяПрародителя(список параметров конструктора) {  
    тело класса  
}
```