**Внутренние классы**

Нестатические вложенные классы называют также внутренними классами. Внутренний класс имеет доступ ко всем переменным и методам своего внешнего класса и может непосредственно ссылаться на них. Внутренние классы создаются внутри окружающего класса :

// внешний класс

class Outer {

int x1 = 0;

int x2 = 0;

void summa(final int x1, final int x2) {

this.x1 = x1;

this.x2 = x2;

Inner inner = new Inner();

inner.display();

}

// внутренний класс

class Inner {

void display() {

System.out.println ("summa = " + String.valueOf(x1 + x2));

}

}

}

...

class MainActivity {

Outer outer = new Outer();

outer.summa(12, 11);

}

**Статические внутренние классы**

Статические внутренние классы декларируются внутри основного класса и обозначаются ключевым словом **static**. Они не имеют доступа к членам внешнего класса за исключением статических.

Статический внутренний класс может содержать статические поля, методы и классы, в отличие от других типов внутренних классов. Пример :

class OuterClass

{

private int outerField;

static int staticOuterField;

public OuterClass() {}

static class InnerClass

{

int getOuterField()

{

return OuterClass.this.outerField; // Эта линия кода вызывает ошибку при компиляции

}

int getStaticOuterField()

{

return OuterClass.staticOuterField; // Эта линия кода синтаксически корректна

}

}

}

**Локальные классы**

Локальные классы объявляются внутри методов основного класса и могут быть использованы только внутри этих методов. Они имеют доступ к членам внешнего класса, а также как к локальным переменным, так и к параметрам метода при одном условии - переменные и параметры используемые локальным классом должны быть задекларированы *final*. **Локальные классы** не могут содержать определение (но могут наследовать) статических полей, методов и классов (кроме констант). Пример :

class OuterClass

{

public OuterClass(){}

private int outerField;

InnerClass inner; // Эта линия кода вызывает ошибку при компиляции

void methodWithLocalClass (final int parameter)

{

InnerClass innerInsideMehod; // Эта линия кода синтаксически корректна

int notFinal = 0;

class InnerClass

{

int getOuterField()

{

return OuterClass.this.outerField; // Эта линия кода синтаксически корректна

}

notFinal++; // Эта линия кода вызывает ошибку при компиляции

int getParameter()

{

return parameter; // Эта линия кода синтаксически корректна

}

};

}

};