



Вторая лекция
Классы

На этом уроке мы научимся

Писать команды на языке Java.

Создавать класс в Java.

Объявлять атрибуты и методы класса.

Узнаем, как методы могут возвращать или не возвращать значения, иметь или не иметь входные параметры.

Узнаем что такое метод **main()**.

Узнаем какие бывают переменные.

Научимся создавать экземпляр класса и вызывать его методы, то есть работать с созданным объектом.

Научимся делать комментарии к нашему коду.

Команды в Java

В языке программирования Java каждую команду принято писать с новой строки. В конце команды ставится точка с запятой.

```
System.out.println("Java это круто!");  
System.out.println("Java это круто!");  
System.out.println("Java это круто!");
```

Программа не может состоять только из команд в свою очередь команда тоже не может быть сама по себе, она – часть функции (в Java функции еще называют методами). А функции являются частью класса.

Классы в Java

Java — объектно-ориентированный язык программирования. Поэтому в Java методы не могут существовать отдельно, они могут быть только частью класса.

Классы можно представить контейнерами, в которых заключена логика программы, определяющая работу приложения, так же Классы позволяют разбить поведение сложных систем на простое взаимодействие взаимосвязанных объектов.

Java позволяет создавать классы, которые представляют объекты из реального мира. Например, можно создать класс Car (автомобиль) или Animal (животное).

Класс объединяет в себе методы и поля, которые описывают поведение и состояние объекта этого класса.

Класс — это шаблон, или проект, по которому будет сделан объект.

Для каждого класса создается отдельный файл, имя которого совпадает с именем класса.

Стандартная библиотека языка Java уже содержит несколько тысяч классов, предназначенных для решения разных задач.

Допустим, нужно создать класс, который будет описывать дом . Тогда нам нужно создать класс Home, который будет содержаться в файле Home.java

```
public class Home {  
    «Тело класса»  
}
```

public – модификатор доступа к классу, в данном случае он нам говорит, что этот класс будет доступен другим классам. В java существуют и другие модификаторы(private, protected, "default"), но об этом позднее.

class – ключевое слово, говорящее о том, что это класс.

Home - имя класса. Имена классов принято писать с заглавной буквы.

{ } – фигурные скобки, между которыми разместится тело нашего класса.

```
public class Home {  
    «переменная»  
    «переменная»  
    «метод»  
    «метод»  
}
```

Методы в Java

Класс может содержать - один, два, три и больше методов в зависимости от сложности класса.

Название метода всегда завершается двумя круглыми скобками (), после которых идет блок кода, обрамлённый фигурными скобками {}.

```
public void sayMeow() {  
    System.out.print("Meow");  
}
```

public void sayMeow() – сигнатура метода, его определение. В данном случае она нам говорит о том, что:

public - метод **sayMeow()** доступен для других классов;

void - метод **sayMeow()** не возвращает никаких данных.

sayMeow – имя метода. Имена методов, в отличие от имен классов, принято писать с маленькой буквы. На конце каждого метода после имени всегда идут круглые скобки (), пустые или содержащие параметры.

```
public String speak(String words){
    String phrase = words + "...may";
    return phrase;
}
```

В отличие от предыдущего метода этот метод возвращает значение и имеет входные параметры. Рассмотрим сигнатуру метода

public String speak(String words):

public - метод `speak()` доступен для других классов;

String —тип значения, которое возвращает метод. В предыдущих случаях ключевое слово **void** указывало на то, что метод ничего не возвращает. В данном случае

String указывает на то, что метод возвращает значение типа строка.

Возвращающие методы должны содержать в своем теле ключевое слово **return**.

Метод `main` в Java

Минимальная программа должна состоять минимум из одного класса, который должен содержать минимум один метод, с которого начинается выполнение программы. Такой метод должен иметь имя `main`. Минимальная программа выглядит вот так:

```
public class Home {  
  
    public static void main (String[] args) {  
  
    }  
}
```

Класс, с которого начинается программа, может иметь любое имя, но метод `main`, с которого начинает выполняться программа, всегда имеет один и тот же вид:

```
public static void main (String[] args)
```

Пакеты package

Программа на Java состоит из классов. Каждый класс хранится в отдельном файле.

Когда у нас много файлов с классами, мы группируем их в папки и подпапки. Классы при этом группируются по пакетам и подпакетам.

Обычно в пакеты объединяют классы одной и той же категории, либо предоставляющие сходную функциональность.

Имена пакетов и подпакетов нужно указывать в коде класса, и они должны совпадать с именами папок и подпапок на диске.

Все встроенные в Java классы, которые входят в комплект поставки, хранятся в пакете с именем `java`. Базовые функции языка хранятся во вложенном пакете `java.lang`. Компилятор Java знает, где найти эти классы.

После оператора `package`, но до любого определения классов в исходном Java-файле, может присутствовать список операторов `import`.

```
package myProject;  
public class Home {  
public static void main (String[] args) {  
  
    }  
}
```

Переменные в Java

Переменные – предназначены для хранения данных. Все данные в Java хранятся с помощью переменных.

Переменные могут быть атрибутами класса, параметрами метода или могут использоваться в программе для краткосрочного хранения данных.

Каждая переменная в Java имеет три важных свойства: тип, имя и значение.

Тип переменной определяет тип данных, которые в ней можно хранить. Переменная может хранить значения только того же типа, что и она сама.

Типы данных делятся на:

- примитивные типы данных,
- ссылочные типы данных,

Правила именования переменных в java

Имя переменной должно начинаться с буквы (маленькой) и состоять из букв (Unicode) цифр и символа подчеркивания. В соответствии с соглашением имя переменной должно начинаться именно с маленькой буквы (с заглавной буквы начинаются имена классов). Пробелы при именовании переменных не допускаются.

- Имя переменной не должно быть ключевым или зарезервированным словом языка Java.
- Имя переменной чувствительно к регистру. `newVariable` и `newvariable` — разные имена.
- При выборе имени переменных, следует использовать полные слова вместо загадочных аббревиатур. Это сделает ваш код более удобным для чтения и понимания.
- Если выбранное вами имя переменной состоит только из одного слова — запишите его маленькими буквами. Если оно состоит из более чем одного слова, то отделяйте каждое последующее слово в имени переменной заглавной буквой. **Например:** `superCounter`, `myDomesticAnimal`

Если переменная сохраняет постоянное значение, то каждое слово следует писать заглавными буквами и отделять при помощи символа подчеркивания.

Пример: `static final int NUMBER_OF_HOURS_IN_A_DAY = 24`

Инициализация переменных

Оператором присваивания является одиночный знак равенства =

переменная = выражение;

```
int y = 10;
```

```
int x;
```

При объявлении переменной, в следующей последовательности указываются:

- тип данных (в данном примере — int — переменная содержит целое число),

- имя переменной (в данном примере имена — x и y),

- начальное значение переменной или, другими словами, инициализация переменной. Однако, это не является обязательным условием при объявлении переменной.

Если нужно объявить несколько переменных одного типа, то это также можно сделать одной строкой, указав имена переменных через запятую.

Примитивные типы данных

Тип	Размер (бит)	Диапазон
byte	1 байт	от -128 до 127
short	2 байт	от -32'768 до 32'767
char	2 байт	от 0 до 65535
int	4 байта	от -2'147'483'648 до 2'147'483'647
long	8 байта	от -9223372036854775808 до 9'223'372'036'854'775'807
float	4 байта	от $\sim 1,4 \cdot 10^{-45}$ до $\sim 3,4 \cdot 10^{38}$
double	8 байта	$\sim 4,9 \cdot 10^{-324}$ до $\sim 1,8 \cdot 10^{308}$
boolean	1 байт	Булевы значения true или false

Также существуют классы-оболочки: В отличие от примитивных типов, они пишутся с заглавной буквы. Эти типы соответствуют примитивным типам, однако являются ссылочными. **Integer, Double, Boolean.** char может хранить только одну букву, но в языке Java существует класс String для работы с более длинным текстом:

```
String myString = "The weather was fine";
```

В итоге класс `Cat` может выглядеть следующим образом:

```
package myProject;
public class Cat {
    private int age;
    private String name;
    private String color;

    public void eat(){
        System.out.print("Eating...\n");
    }

    public void sleep(){
        System.out.print("Sleeping zz-z-z-z...\n");
    }

    public String speak(String words){
        String phrase = words + "...mauu...\n";
        return phrase;
    }
}
```

Экземпляр класса

Прежде чем использовать созданные нами методы в классе нужно создать экземпляр класса.

```
Cat ourCat = new Cat();
```

Новый объект (или экземпляр) создаётся из существующего класса при помощи ключевого слова **new**.

Вы можете создать несколько объектов на основе класса.

Переменную `ourCat` теперь можно использовать для вызова методов класса `Cat`, например: **`ourCat.eat();`**

Добавим в наш метод **`main`** который находится в классе `Home` такие команды.

```
Cat ourCat = new Cat();  
ourCat.eat();  
ourCat.sleep();  
String say = ourCat.speak("Play with me");  
System.out.println(say);
```

```
package myProject;

public class Home {

    public static void main (String[] args) {

        Cat ourCat = new Cat();

        ourCat.eat();
        ourCat.sleep();
        String say = ourCat.speak("Play with me");
        System.out.println(say);

    }
}
```

Запустим нашу программу на исполнение.

// Комментарии

В языке Java ты можешь писать не только команды, но и комментарии к ним прямо в коде. Такие комментарии игнорируются компилятором, будто и нет их вовсе. При исполнении программы все комментарии пропускаются!

Начало комментария обозначается парой символов «/*», а конец — «*/». Когда программа будет компилироваться, компилятор пропустит все, что находится между символами /* и */

Есть еще один способ задать комментарий в коде — с помощью символов //

```
/*Первая программа  
создана 28.10.2015*/
```

```
public class Home  
{  
    public static void main (String[] args)  
    {  
        System.out.print("Java ");  
        System.out.print("это "); //вот тут еще один комментарий  
        System.out.print("Круто ");  
    }  
}
```