

Алексей Максимович
Македонский

alexeymakedonsky@gmail.com

Литература

- Патрик Ноутон, Герберт Шилдт «Java 2 в подлиннике»
- Брюс Эккель «Философия Java»
- Марк Гранд «Шаблоны проектирования в Java»
- Мартин Фаулер «Рефакторинг. Улучшение существующего кода»

ООП: объекты и классы

В рамках объектно ориентированного программирования было предложено объединять данные и использующие их подпрограммы в единую сущность: **объект**.

В свою очередь множество объектов идентичной структуры оказалось удобно декларировать с помощью **классов**.

Например: что такое «стол»?

ООП: объекты и классы

В рамках объектно ориентированного программирования было предложено объединять данные и использующие их подпрограммы в единую сущность: **объект**.

В свою очередь множество объектов идентичной структуры оказалось удобно декларировать с помощью **классов**.

Например: что такое «стол»?

Если говорить о столах вообще, то получится класс (набор объектов идентичной структуры).

Конкретный стол является объектом (экземпляром - instance) некоторого класса (имеет конструкцию определенного типа), но обладает некоторыми собственными характерными чертами.

Процедурное программирование:

```
public static void main(String [] args) {  
    System.out.println("Hello, World!");  
}
```

Java – полностью объектный язык. Поэтому даже в простейшей программе будет описание класса объектов:

```
public class HelloWorld {  
    public static void main(String [] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

Описание класса должно находиться в одноименном файле с расширением .java

Java – полностью объектный язык. Поэтому даже в простейшей программе будет описание класса объектов:

```
public class HelloWorld {  
    public static void main(String [] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

Описание класса должно находиться в одноименном файле с расширением .java

Для компиляции и запуска этой программы:

```
javac HelloWorld.java  
java HelloWorld
```

JDK – Java Development Kit

JVM – Java Virtual Machine

Язык Java разрешает использовать все основные конструкции структурного и процедурного программирования:

ветвление:

```
if (<условие>) {  
    <операторы>  
} else {  
    <операторы>  
}
```

составное ветвление:

```
switch (<выражение>) {  
    case <константа 1>:  
    {  
        <операторы 1>;  
    }  
    case <константа 2>:  
    {  
        <операторы 1>;  
        break;  
    }  
    default:  
    {  
        <операторы>;  
        break;  
    }  
}
```


Язык Java разрешает использовать все основные конструкции структурного и процедурного программирования:

```
while (<условие>) {  
    <тело цикла>  
}
```

```
for (<инициализация цикла>; <условие>; <шаг цикла>) {  
    <тело цикла>  
}
```

```
do {  
    <тело цикла>  
} while (<условие>)
```

Язык Java разрешает использовать все основные конструкции структурного и процедурного программирования:

подпрограмма:

```
<тип результата> <имя подпрограммы>(<список параметров>) {  
    <операторы>;  
    return <значение>;  
}
```

тип **void** – отсутствие результата

Язык Java предлагает программисту следующие predefined типы данных (так называемые примитивные типы):

- `int` (32-битное знаковое целое)
- `byte` (8-битное знаковое целое)
- `short` (16-битное знаковое целое)
- `long` (64-битное знаковое целое)
- `float` (32-битное вещественное число, стандарт IEEE 754)
- `double` (64-битное вещественное число, стандарт IEEE 754)
- `char` (16-битный код символа в кодировке Unicode)
- `boolean` (логическое значение `true` или `false`)

Класс animal

```
class animal {  
    protected Integer footsCount;  
    protected Integer eyesCount;  
  
    public animal() {  
        this.footsCount = 0;  
        this.eyesCount = 0;  
        this.haveFur = false;  
    }  
  
    public Integer getFoodsCount() {  
        return this.footsCount;  
    }  
  
    public void setFoodsCount(Integer footsCount) {  
        this.footsCount = footsCount;  
    }  
}
```

Класс dog

```
class dog extends animal {  
    protected String name;  
  
    public animal() {  
        super();  
        this.name = "";  
    }  
  
    public animal(String dogsName) {  
        super();  
        this.name = dogsName;  
    }  
  
    public static void say() {  
        System.out.println("woof-woof!");  
    }  
}
```

Класс vehicle