

Базовые классы Java

Пакеты

Как правило, в Java классы объединяются в пакеты. Пакеты позволяют организовать классы логически в наборы. По умолчанию java уже имеет ряд встроенных пакетов, например, `java.lang`, `java.util`, `java.io` и т.д. Кроме того, пакеты могут иметь вложенные пакеты.

Организация классов в виде пакетов позволяет избежать конфликта имен между классами. Ведь нередки ситуации, когда разработчики называют свои классы одинаковыми именами. Принадлежность к пакету позволяет гарантировать однозначность имен.

Чтобы указать, что класс принадлежит определенному пакету, надо использовать директиву `package`, после которой указывается имя пакета:

```
1 package название_пакета;
```

— Файлы в компьютере группируются по папкам. Классы в Java (а каждый класс лежит в отдельном файле) группируются **по пакетам, которые являются папками на диске**. Ничего принципиально нового. Но есть два замечания

— **Первое**. «**Полным уникальным именем класса**» является «**имя пакета**» + «**имя класса**». Примеры:

Полное уникальное имя	Имя пакета	Имя класса
java.io.FileInputStream	java.io	FileInputStream
java.lang.String	java.lang	String
java.util.ArrayList	java.util	ArrayList
org.apache.tomcat.Servlet	org.apache.tomcat	Servlet
Cat	отсутствует	Cat

Пакет JAVA.LANG

- Пакет `java.lang` представляет собой сердце языка Java. Этот пакет включает не только классы-оболочки для основных типов данных, но и средства обработки ошибок. Более того, некоторые классы дают определенную возможность управления системой клиента, хотя бы через командную строку или консоль Java в Netscape.

Классы:

Boolean

Класс Boolean работает как класс-оболочка, предоставляя программисту удобные возможности для работы с логическими значениями.

String

Класс `String` описывает природу строковой переменной и основные операции со строками.

Character

Класс `Character` - это класс-оболочка для обработки значений символов.

Class

Класс `Class` позволяет получить информацию о коде, использованном при создании класса.

Double

Класс Double расширяет класс Number, это класс-оболочка для работы с числами с плавающей точкой с двойной точностью.

Float

Класс Float расширяет класс Number, это класс-оболочка для работы с числами с плавающей точкой с одинарной точностью.

Integer

Класс Integer расширяет класс Number, это класс-оболочка для работы с целыми числами.

long

Класс Long расширяет класс Number, это класс-оболочка для работы с длинными целыми числами.

Math

Класс Math предоставляет разнообразный набор методов для математических вычислений, а также полезные константы.

Number

Класс Number является абстрактным и служит основой всех классов-оболочек для работы с числовыми типами данных.

Object

Класс Object - это основа всех классов языка программирования Java.

System

Класс `System` позволяет управлять некоторыми системными функциями, хотя и в обычной машинно-независимой манере.

Пример использования класса `System`

```
public class SystemExample {
    public static void main(String argv[]) {

        System.out.print("Это пример использования");
        System.out.println("стандартного потока вывода.");
        String s=System.getProperty("java.version");
        System.out.println(
            "Я использую язык Java версии"+s+".");
    }
}
```

Пример

```
Build Run Tools VCS Window Help OOPFor2Course - Main.java
ain
Main (4)
Main.java x
1 package PD1_19R.Lessons;
2 // Importirovanie CLassa iz ipredelennogo paketa
3
4 import java.util.Scanner;
5
6 //class nash Main
7 public class Main {
8     // nash glavny metod, to4ka xzapuska
9     public static void main(String[] args) {
10
11         System.out.println("Hello world");
12         String s = System.getProperty("java.version");
13         System.out.println(" The version of Java is " + s);
14
15         Scanner scanner = new Scanner(System.in); // Sozдание object classa scanner
16         System.out.println("Vvedite c: ");
17         int c = scanner.nextInt();
18         System.out.println("Vvedite x: ");
19         int x = scanner.nextInt();
20         int y = (int) (Math.pow(c, 2) + Math.sqrt(x));
21         System.out.println("y =" + y);
22         //perevonnaya s primitivnym tipom
23         int a = 20;
24         //ssylochnaya peremennaya s obertkoi
25         Integer u = 15;
26
27         char symbol = 'a';
28         // Obyavlenie peremennyh String
29         String name = "Oleg";
30         System.out.println(name.charAt(2));
31         char t= name.charAt(1);
32     }
```