



Общие представления о языке Java

Курс «Технологии Java»

Java и другие языки программирования.

Системное и прикладное программирование

- Языки низкого и высокого уровня.
- Трансляция. Компиляция и интерпретация.
- Системное и прикладное программирование. C++, JavaScript, Java.
- Java - универсальный язык прикладного программирования.
- Java - компилируемый, платформонезависимый, объектно-ориентированный язык с C-образным синтаксисом.
- Версии Java 1.0 и 1.1 принято называть Java 1. Все версии Java начиная с 1.2 принято называть Java 2, Java5, Java 9 и т.п.

История

- 23 марта, 1995 День рождения Java.
James Goslings, Sun Microsystems
- Январь 1996 Выпуск JDK 1.0
- Декабрь 1998 Основание Java Community Process (JCP)
- 15 июня, 1999 Декларация трех основных направлений
Java платформы: J2SE, J2EE, J2ME
- Апрель 2001 Java 2 Platform, Enterprise Edition (J2EE) SDK 1.3
- 1 октября 2004 Выпуск Tiger (Java 5.0)
- 27 января 2010 Oracle купил активы Sun Microsystems
- 2019 появляется коммерческая лицензия Oracle

Виртуальная Java-машина, байт-код, JIT-компиляция

- Переносимость программ. Байт-код.
- Виртуальная Java-машина (JVM).
- Объектная ориентированность.
- Надёжность кода – уровень исходных кодов и уровень времени выполнения.
- JIT-компиляция (Just-In-Time).
- “Родной” код (native code)
- Многоязычное программирование на виртуальной машине Java (Groovy, Scala, Clojure)

Типы программ Java

- Приложение (application) – аналог “обычной” прикладной программы. Может иметь оконный интерфейс. Запускается под управлением java-машины.
- Апплет (applet) – специализированная программа, работающая в окне WWW-документа под управлением браузера.
- Сервлет (servlet) - специализированная программа, работающая в WWW на стороне сервера.
- Модуль EJB (Enterprise JavaBeans) – предназначен для многократного использования серверными приложениями Java.
- Пакет– библиотека классов Java предназначена для многократного использования программами Java. Может быть в виде архива (jar).

SDK - Software Development Kit

SDK – Комплект разработки программного обеспечения.

Имеется несколько технологий:

- **Java ME** – комплект Java Micro Edition, предназначенный для программирования устройств с малым объемом памяти и малой разрядностью процессора.
- **Java SE** – комплект Java Standard Edition, предназначенный для программирования приложений и апплетов.
- **Java EE** – комплект Java Enterprise Edition, предназначенный для написания серверного программного обеспечения, развертывания сложных систем.
- **Java Card** — технология предоставляет безопасную среду для приложений, работающих на смарт-картах и других устройствах с очень ограниченным объёмом памяти и возможностями обработки

Утилиты SDK

- `javac` -компилятор в режиме командной строки
- `java` - запуск виртуальной машины в режиме командной строки
- `appletviewer` - запуск апплетов
- `jdb` - отладчик
- `javadoc` - генератор документации
- `jar` - создание и управление архивами `jar`
- `javah` - генератор `h`-файлов `C/C++` интерфейса `JNI`
- `javap` - дизассемблер классов
- `extcheck` - обнаружение конфликтов между файлами архивов `jar`
- `native2ascii` - конвертация параметра, передаваемого в виде текста на национальном алфавите, в последовательность символов `UNICODE`
- ...???

Идентификаторы

- Язык Java является **регистро-чувствительным**.
- **Идентификаторы** - это имена переменных, классов, методов и т.д. В идентификаторах можно применять только буквы и цифры, причём первой всегда должна быть буква, а далее может идти произвольная комбинация букв и цифр. Длина идентификатора в Java любая.
- Традиционные соглашения стиля именования

Переменные и типы.

Примитивные и ссылочные типы

- **Переменная** – это именованная ячейка памяти, содержимое которой может изменяться. При объявлении переменной сначала указывается тип переменной, а затем идентификатор задаваемой переменной.
- **Типы** в Java делятся на **примитивные** и **ссылочные**. Существует несколько (8) predefined примитивных типов, все остальные – ссылочные. Все пользовательские типы кроме типов-перечислений являются ссылочными. Значение null соответствует ссылочной переменной, которой не назначен адрес ячейки с данными.

Пример приложения

```
public class HelloWorld {  
  
    public static void main(String args[]){  
        System.out.println("Hello World!!! Java application!!!");  
        for (int i=0; i<args.length; i++)  
            System.out.println("Param "+i+" = "+args[i]);  
    }  
  
}
```

Пример апплета

```
import java.awt.*;
import java.applet.*;
/* пример файла hello.html для запуска апплета
   <applet code=HelloWorld.class width=200 height=60>
   </applet>
*/
public class HelloWorld extends Applet {
    public void paint( Graphics gc ) {
        gc.drawString("Hello World!!! I'm Java in Web!!!", 10, 20); }
}
```

Пример приложения и апплета

```
import java.awt.*;
import java.applet.*;
/* пример файла hello.html для запуска апплета
  <applet code=HelloWorld.class width=200 height=60>
  </applet>
*/
public class HelloWorld extends Applet {
    public void paint( Graphics gc ) {
        gc.drawString("Hello World!!! I'm Java in Web!!!", 10, 20); }
    public static void main(String args[]){
        System.out.println("Hello World!!! I'm Java application!!!");
        for (int i=0; i<args.length; i++)
            System.out.println("Param "+i+" = "+args[i]);
    }
}
```

Файл с текстом

HelloWorld.java

Имя (регистрозависимое)
совпадает с именем класса

Компиляция средствами JDK

```
>javac HelloWorld.java
```

Результат компиляции
HelloWorld.class

Выполнение

- Приложения

```
>java HelloWorld
```

- Апплета

- Просмотр файла hello.html в окне браузера
- Используя утилиту JDK

```
>appletviewer hello.html
```

Классы и объекты.

Работа со ссылочными переменными.

Сборка мусора

- Объект создаётся с помощью вызова *конструктора* - специальной подпрограммы-функции, задаваемой в классе.
- Методы делятся на *методы объектов* и *методы классов*. Метод объекта можно вызывать только из объекта соответствующего типа. А метод класса может работать и при отсутствии объекта, и вызываться из класса.
- Переменные ссылочного типа содержат адреса данных, а не сами данные. Поэтому присваивания для таких переменных меняют адреса, но не данные. Все объектные типы являются ссылочными.
- На один объект может существовать множество ссылок.
- Потеря последней ссылки на объект приводит к сборке мусора.

Начало работы

```
import java.util.*;
/**
 * This program demonstrates console input.
 * @version 1.10 2004-02-10
 * @author Cay Horstmann
 */
public class InputTest {
    public static void main(String[ ] args) {
        Scanner in = new Scanner(System.in);

        System.out.print("What is your name? "); // get first input
        String name = in.nextLine();

        System.out.print("How old are you? "); // get second input
        int age = in.nextInt();
        // display output on console
        System.out.println("Hello, " + name + ". Next year, you'll be " + (age + 1));
    }
}
```

```

import java.math.*;
import java.util.*;

public class BigIntegerTest {
    public static void main(String[ ] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("How many numbers do you need to draw? ");
        int k = in.nextInt();

        System.out.print("What is the highest number you can draw? ");
        int n = in.nextInt();
        /*
         * compute binomial coefficient  $n*(n-1)*(n-2)*...*(n-k+1)/(1*2*3*...*k)$ 
         */
        BigInteger lotteryOdds = BigInteger.valueOf(1);

        for (int i = 1; i <= k; i++)
            lotteryOdds = lotteryOdds.multiply(BigInteger.valueOf(n - i + 1)).divide(
                BigInteger.valueOf(i));

        System.out.println("Your odds are 1 in " + lotteryOdds + ". Good luck!");
    }
}

```

```

public class LotteryArray {
    public static void main(String[] args) {
        final int NMAX = 10;

        int[][] odds = new int[NMAX + 1][]; // allocate triangular array
        for (int n = 0; n <= NMAX; n++)
            odds[n] = new int[n + 1];

        for (int n = 0; n < odds.length; n++) // fill triangular array
            for (int k = 0; k < odds[n].length; k++) {
                /* compute binomial coefficient  $n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot (n-k+1) / (1 \cdot 2 \cdot 3 \cdot \dots \cdot k)$  */
                int lotteryOdds = 1;
                for (int i = 1; i <= k; i++)
                    lotteryOdds = lotteryOdds * (n - i + 1) / i;
                odds[n][k] = lotteryOdds;
            }

        for (int[] row : odds) { // print triangular array
            for (int odd : row)
                System.out.printf("%4d", odd);
            System.out.println();
        }
    }
}

```

Описание и тестирование класса

- Самотестирование

```
[public] class MyClass {  
    ....                //поля  
    ....                // методы  
    public static void main ( String [ ] args) {  
        MyClass mc = new MyClass( . . . );  
        // проверка методов класса  
    }  
}
```

Описание и тестирование класса

- Через класс - тестер

```
[public] class MyClassTest {  
    public static void main ( String [ ] args) {  
        MyClass mc = new MyClass( . . . );  
        // проверка методов класса  
    }  
}
```

- Через технологию Unit-тестирования