Лекция 7

- 13 октября

Раздел 3. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ JAVA

1. ВВЕДЕНИЕ В JAVA

- Java – объектно-ориентированный, интерпретируемый, надёжный, безопасный, архитектурно-нейтральный, переносимый, многопотоковый и динамический язык.

1.1. Простейшие программы

- Традиционно первая программа в С-подобных языках выводит строку *Hello, World!*
-

```
•class HelloProg{
•  public static void  main(String[] args){
•    System.out.println(“Hello, World!”);
•  }
•}
```

-
-
-

Вторая программа

- Ещё одна простая программа:
-
- class SimpleProg{
- public static void main(String[] args){
- int a,b,c;
- a=17;
- b=34;
- c=a+b;
- System.out.println(“c=”+c);
- }
- }
-
-

1.2. Типы данных

- Рассмотрим сначала простые типы данных.
-
- ***Целочисленные типы:***
- byte — занимает 8 разрядов — хранит целое со знаком
- short — занимает 16 разрядов — хранит целое со знаком
- int — занимает 32 разряда — хранит целое со знаком
- long — занимает 64 разряда — хранит целое со знаком
-
- ***Типы с плавающей точкой:***
- float — занимает 32 разряда — хранит число с плавающей точкой
- double — занимает 64 разряда — хранит число с плавающей точкой
-
- ***Символьный тип:***
- char — занимает 16 разрядов — хранит символ Unicode
-
- ***Логический тип:***
- boolean — хранит true или false

Константы

- Целые константы записываются как обычно:
- 14, -234.
- Они имеют тип *int*.
- Константы типа ***long*** имеют суффикс ***l*** или ***L***:
- 156l, -657L.
- Константы с плавающей точкой могут содержать суффикс ***f*** или ***F*** (для типа ***float***), или ***d*** или ***D*** (для типа ***double***). По умолчанию константа имеет тип ***double***.
- Величины с плавающей точкой могут принимать специальные значения: положительная бесконечность ***POSITIVE_INFINITY***, отрицательная бесконечность ***NEGATIVE_INFINITY***, не число ***NaN***.
- Символьные константы записываются в одиночных кавычках:
- ‘а’, ‘Б’, ‘7’, ‘+’.
- Логические константы: ***true, false***.
- Строковые константы записываются в двойных кавычках:
- “строка”, “ф123”.

Переменные

- Объявления переменных выглядят следующим образом:
-
- *<тип> <список переменных>;*
-
- Например, *float x,y,z;*

- При этом переменные могут быть инициализированы:
-
- int a=4,b,c=-2;

1.3. Операции

- ***Арифметические операции:***

- \+ сложение
- \- вычитание
- * умножение
- / деление (получение частного)
- % деление по модулю (получение остатка)
- ++ увеличение на 1
- -- уменьшение на 1

Операции (продолжение)

- ***Операции отношения:***
- > больше
- < меньше
- >= больше или равно
- <= меньше или равно
- == равно
- != не равно
- Результат выполнения операции имеет тип ***boolean***.

- ***Логические операции:***
- & логическое И
- | логическое ИЛИ
- ^ исключающее ИЛИ
- ! логическое НЕ

Операции (продолжение)

- ***Сокращённые логические операции:***
- && сокращённое И
- || сокращённое ИЛИ

Операции (продолжение)

- ***Поразрядные операции:***
- & поразрядное И
- | поразрядное ИЛИ
- ^ поразрядное исключающее ИЛИ
- >> поразрядный сдвиг вправо
- >>> беззнаковый поразрядный сдвиг вправо
- << поразрядный сдвиг влево
- ~ поразрядная инверсия

Операции (продолжение)

- Возможно объединение операций с присваиванием: +=, >>=, *= и т.д.
- При выполнении операций с данными типа ***byte*** или ***short*** производится их расширение до типа ***int***. Поэтому следующий фрагмент кода содержит ошибку:
-
- short a=43, b, c=22;
- b=a*c;
-
- При выполнении умножения переменные будут расширены до типа ***int***, и результат будет иметь тип ***int***, поэтому его нельзя присвоить переменной ***b***, имеющей тип ***short*** (более короткий). Здесь необходимо использовать явное приведение типа:
-
- short a=43, b, c=22;
- b = (short) a*c;

1.4. Массивы

Для создания одномерного массива используется код:

<тип> <имя массива>[ ] = new <тип> [<размер>];

или

<тип>[ ] <имя массива> = new <тип> [<размер>];

```
int a[] = new int[10];
```

или

```
int[] a = new int[10];
```

Обращение к элементу массива выглядит следующим образом:

```
a[i]
```

Массивы (продолжение)

```
int[] b;
. . .
b = new int[10];

int[] c = {4,-2,5,23,-6,0,7};

int[] a;
        . . .
        a = new int[]{3,2,0,1,-2,7};
```

Массивы (продолжение)

- Массивы содержат специальную переменную ***length*** – длину массива.
-
- for(int i=0;i<a.length;i++) a[i]=i+1;
-
-
- Многомерные массивы создаются аналогично одномерным, например:
-
- int[][] q = new int[3,4];
- int y[][] = {{1,2,3},{4,5,6}};
-
-
-
- Обращение к элементу массива выглядит следующим образом:
-
- q[i][j] y[i][j]

Массивы (продолжение)

```
int[][] a = new int[4][];
        a[0] = new int[3];
        a[1] = new int[4];
        a[2] = new int[1];
        a[3] = new int[3];
```

Обращение к элементу массива выглядит как обычно:

```
a[i][j]
```

```
int m[][] = {{1},{2,3},{4,5,6},{6,7,8,9}};
```

1.5. Строки

- Строки – это объекты класса ***String.***
-
- String s = “Строка 1”;
- s = “Строка 2”;
-
- Для строк определена операция сцепления +.
- Поле ***length*** строки задаёт её длину.

1.6. Математические функции

- Математические функции в языке Java реализованы как статические методы класса ***Math***. Некоторые функции:
-
- *static double sin(double x)*
- *static double cos(double x)*
- *static double tan(double x)*
- *static double sqrt(double x)*
- *static double log(double x)*
- *static double exp(double x)*
- *static double asin(double x)*
- *static double acos(double x)*
- *static double atan(double x)*
- *static double pow(double x, double y)*
- *static double abs(double x)*
-
- Тригонометрические функции ***sin, cos, tan*** в качестве аргумента получают угол в радианах.

Математические функции (продолжение)

- Обращения к методам выглядят следующим образом:
-
- double y = Math.sqrt(8.2);
- float z = (float)Math.cos(1.34);
-
- Во втором операторе используется явное преобразование типа, поскольку метод ***cos*** возвращает тип ***double***.
- В классе также определены константы:
-
- *double PI*
- *double E*
-
- Они задают значения пи и ***e***.

1.7. Пакеты

- Пакеты – это средство группирования классов. Они позволяют избежать конфликта имён.
- Для доступа к пакету используется оператор ***import***, например:
-
- import java.util.Date;
-
- Здесь импортируется класс ***Date*** для работы с датой и временем. Можно импортировать все классы пакета, например:
-
- import java.util.*;
-
- Пакеты имеют иерархическую структуру: пакет ***util*** входит в состав пакета ***java***. Имена классов в Java пишутся с большой буквы, а имена пакетов – с маленькой.
- По умолчанию во все программы автоматически включается пакет ***java.lang***, содержащий наиболее употребительные классы.
- В частности в него входят классы: ***System, String, Math***. Поэтому в предыдущих примерах оператор ***import*** не использовался.

2. ОПЕРАТОРЫ
2.1. Оператор присваивания

- Общая форма записи обычного оператора присваивания выглядит следующим образом:
-
- *<переменная> = <выражение>;*
-
- Например:
- int x,y,z;
- x=y=z=100;
-
- Могут быть составные операторы присваивания:
- *<переменная> <знак операции> = <выражение>;*
-
- Например:
- int a=3, b=4;
- a += b+2;

2.2. Условный оператор

- Условный оператор имеет две формы: краткую и полную.
-
- *if (<логическое выражение>) <оператор>*
- *if (<логическое выражение>) < оператор1 > else < оператор2 >*
-
-
-
-
- Например:
-
- int a=3, b=4, c=5;
- if(a>b) c=6; else c=7;
-
-
-

2.3. Составной оператор

Структура составного оператора выглядит следующим образом:

{

<операторы>

}

Во внутреннем блоке нельзя объявлять переменную с тем же именем, что и во внешнем блоке.

Пример составного оператора:

```
int a=3, b=4, c=5;
if(a<=b)
  {
    b=2;
    c=6;
   }
  else c=7;
```