Практическая работа №2

Переменные и константы

Для хранения данных в программе предназначены переменные. Переменная представляет именованную область памяти, которая хранит значение определенного типа. Каждая переменная имеет тип, имя и значение. Тип определяет, какую информацию может хранить переменная или диапазон допустимых значений.

Переменные объявляются следующим образом:

```
тип данных имя переменной;
```

Например, определим переменную, которая будет называться x и иметь тип int:

```
int x;
```

В качестве имени переменной может выступать любое произвольное название, которое удовлетворяет следующим требованиям:

- имя может содержать любые алфавитно-цифровые символы, а также знак подчеркивания, при этом первый символ в имени не должен быть цифрой;
- в имени не должно быть знаков пунктуации и пробелов;
- имя не может быть ключевым словом языка Java.

Кроме того, при объявлении и последующем использовании надо учитывать, что Java - регистрозависимый язык, поэтому объявления int num; и int NUM; будут представлять две разных переменных.

Объявив переменную, мы можем присвоить ей значение:

```
int x;      // объявление переменной
x = 10;     // присвоение значения
System.out.println(x);  // 10
```

Также можно присвоить значение переменной при ее объявлении. Этот процесс называется инициализацией:

```
int x = 10;     // объявление и инициализация переменной
System.out.println(x);  // 10
```

Если мы не присвоим переменной значение до ее использования, то мы можем получить ошибку, например, в следующем случае:

```
int x;
System.out.println(x);
```

Через запятую можно объявить сразу несколько переменных одного типа:

```
int x, y;
x = 10;
y = 25;
```

```
System.out.println(x);  // 10
System.out.println(y);  // 25
```

Также можно их сразу инициализировать:

```
int x = 8, y = 15;
System.out.println(x);  // 8
System.out.println(y);  // 15
```

Отличительной особенностью переменных является то, что мы можем в процессе работы программы изменять их значение:

```
int x = 10;
System.out.println(x);  // 10
x = 25;
System.out.println(x);  // 25
```

Кроме переменных, в Java для хранения данных можно использовать константы. В отличие от переменных константам можно присвоить значение только один раз. Константа объявляется так же, как и переменная, но перед типом данных указывается ключевое слово final:

```
final int LIMIT = 5;
System.out.println(LIMIT);  // 5
LIMIT=57; // вызовет ошибку, так как LIMIT - константа
```

Константы позволяют задать такие значения, которые не должны больше изменяться, и обычно имеют имена, состоящие из символов в верхнем регистре.

Начиная с версии 10, в язык Java было добавлено ключевое слово var, которое также позволяет определять переменную. Слово var указывается вместо типа данных, а сам тип переменной выводится из того значения, которое ей присваивается. Например, если переменной x присваивается число 10, то переменная будет представлять тип int:

```
var x = 10;
System.out.println(x);  // 10
```

Если переменная объявляется с помощью var, её необходимо проинициализировать, то есть предоставить ей начальное значение, иначе мы получим ошибку:

```
var x;      // ! Ошибка, переменная не инициализирована
x = 10;
```

Консольный вывод данных

Важной особенностью любой программы является умение вводить некоторые данные, которые она обрабатывает, и выводить результаты обработки этих данных.

Для вывода данных в консоль используется класс System. Этот класс располагается в пакете java.lang, который автоматически подключается к программе, поэтому дополнительно импортировать данный пакет и класс не требуется. Для создания потока вывода в классе System определен объект out. В этом объекте определен метод println, который позволяет вывести на консоль некоторое значение с последующим переводом консоли на следующую строку:

```
System.out.println("Hello, world!");
```

В метод println передается любое значение, как правило, строка, которое надо вывести на консоль. При необходимости можно и не переводить курсор на следующую строку. В этом случае можно использовать метод System.out.print(), который аналогичен println за тем исключением, что не осуществляет перевода на следующую строку:

```
System.out.print("Hello, world!");
```

Методы print() и println() выводят значения на экран без какого-либо форматирования. Элементарное форматирование приходится реализовывать при помощи добавления к строкам дополнительных пробелов или других символов. Например, есть два числа, значения которых нужно вывести на экран. В этом случае мы можем, например, написать так:

```
int x=5;
int y=6;
System.out.println("x=" + x + "; y=" + y); // x=5; y=6
```

В Java также есть функция для форматированного вывода – метод System.out.printf(). С ее помощью мы можем переписать предыдущий пример следующим образом:

```
System.out.printf("x=%d; y=%d \n", x, y); // x=5; y=6
```

В данном случае символы %d обозначают спецификатор, вместо которого подставляется один из аргументов. В данном случае вместо первого %d подставляется значение переменной x, а вместо второго - значение переменной y. Сама буква d означает, что данный спецификатор будет использоваться для вывода целочисленных значений.

Для каждого типа данных определён собственный спецификатор:

Спецификатор формата	Выполняемое форматирование
%a	Шестнадцатеричное значение с плавающей точкой
%b	Логическое (булево) значение аргумента
%c	Символьное представление аргумента
%d	Десятичное целое значение аргумента
%h	Хэш-код аргумента
%e	Экспоненциальное представление аргумента
%f	Десятичное значение с плавающей точкой
%g	Выбирает более короткое представление: %e или %f
%o	Восьмеричное целое значение аргумента
%n	Вставка символа новой строки
%s	Строковое представление аргумента
%t	Время и дата
%x	Шестнадцатеричное целое значение аргумента
%%	Вставка знака %

Также возможно использование спецификаторов с заглавными буквами: например, спецификатор %A является эквивалентом спецификатора %a. Форматирование с их помощью обеспечивает перевод символов в верхний регистр.

Помимо спецификаторов, в методе printf могут использоваться флаги:

Флаг формата	Выполняемое форматирование
-	Выравнивание влево
#	Изменяет формат преобразования
0	Выводит значение, дополненное нулями вместо пробелов
пробел	Положительные числа предваряются пробелом
+	Положительные числа предваряются знаком +
,	Числовые значения включают разделители групп
(	Отрицательные числовые значения заключаются в скобки

Рассмотрим пример использования флагов:

```
System.out.printf("%,.2f%n", 10000.0 / 3.0); // 3 333,33
System.out.printf("%, (.2f%n", -10000.0 / 3.0); // (3 333,33)
System.out.printf("%09.2f%n", 10000.0 / 3.0); // 003333,33
```

В строке, определяющей формат, может задаваться индекс форматируемого параметра. Индекс должен следовать непосредственно за символом % и завершаться знаком $:

```
System.out.printf("Hello %1$s!%n%1$s, how are you?%nWelcome to the site %2$s",
        "John", "www.site.com");
```

Таким образом, общий синтаксис формата вывода можно описать так:

%[индекс][флаги][ширина][.точность]спецификатор

Консольный ввод данных

Для ввода данных из консоли в классе System определен объект in. Однако работать непосредственно через объект System.in не очень удобно – как правило, в качестве посредника используют класс Scanner. Перед использованием данный класс необходимо подключить к программе при помощи директивы импорта:

```
import java.util.Scanner;
```

Рассмотрим примеры считывания данных примитивных типов:

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int a = in.nextInt();//считываем целое число
        byte b = in.nextByte();//считываем байтовое число
        String c = in.nextLine();//считываем одну строку целиком
        double d = in.nextDouble();//считываем вещественное число
        long  e = in.nextLong();//считываем длинное целое число
        short f = in.nextShort();//считываем короткое целое число
        String s = in.next();//считываем строку до первого пробела
    }
}
```

Рассмотрим программу для ввода информации о человеке:

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Введите имя: ");
        String name = in.nextLine();
        System.out.print("Введите возраст: ");
        int age = in.nextInt();
        System.out.println("Ваше имя: " + name + ", ваш возраст: " + age);
    }
}
```

Практические задания

1. Составить программу вывода на экран числа, вводимого с клавиатуры. Выводимому числу должно предшествовать сообщение "Вы ввели число ".
2. Составить программу вывода на экран числа, вводимого с клавиатуры. После выводимого числа должно следовать сообщение: " - вот какое число Вы ввели".
3. Вывести на одной строке числа 1, 13 и 49 с одним пробелом между ними.
4. Вывести на экран числа 50 и 10 одно под другим.
5. Составить программу вывода на экран следующей информации:

   а) 7 см

   б) x 25

   в) x y

   где x и y – переменные величины целого типа, значения которых вводятся с клавиатуры и должны быть выведены вместо имен величин.