IT ШКОЛА SAMSUNG

Модуль 1. Основы программирования

Урок 1-2. Типы данных и операции.

SAMSUNG

Переменные

Все что делает компьютер – он обрабатывает данные. Любые формы информации, будь то звук, картинка, текст или видео, это лишь числа в памяти компьютера. **Тип данных** – это подсказка программе, как обрабатывать хранимые числа.

В Java различают типы данных двух групп:

- Простые типы
- Ссылки на объекты

Ячейки памяти для хранения данных, объявленные определенным образом – называются переменными.

Переменные

Данные это конечно цифры, и хранятся они в памяти компьютера, но хранение и использование отличается в зависимости от их типа. Это как с продуктами для приготовления. Они все являются продуктами и все хранятся в шкафу, но вот хранятся по разному и в разных емкостях.

Переменные

<тип> <имя переменной> [= <значение>];

Например:

- Int a2;
- double xx = .15;

Переменная может называться любой последовательностью латинских букв, цифр и знаков подчеркивания при этом не могут начинаться с цифры.

- int MyNewSize_2_Triangle; - правильно
- Int 2size; - неправильно

Переменные

<тип> <имя переменной> [= <значение>];

Переменная – именованная ячейка памяти, которая может менять своё значение. Переменная имеет: тип, имя, значения

В Java используется статическая типизация. Это значит, что переменная не может сменить свой тип, например, переменная int превратиться в переменную double не может. Также переменная не может менять своего названия.

Переменные

Вывод чисел выполняется так же, как и строк, при помощи PrintStream, например:

```
static Scanner in = new Scanner(System.in);
int x = in.nextInt();
in.useLocale(Locale.US);
double y = in.nextDouble();

static PrintStream out = new PrintStream(System.out)
out.println(x);
out.println(y);

double r = in.nextDouble();
out.println(3.1415 * r * r);
```

Программа A+B

```
import java.io.PrintStream;
import java.util.Scanner;

public class MyProgram
{
// public необходим, чтобы Android приложение могло изменить
значение переменных.
public static Scanner in = new Scanner(System.in);
public static PrintStream out = System.out;

public static void main(String[] args)
{
int a, b;
out.println("Введите два числа:");
a = in.nextInt();
b = in.nextInt();
c = a + b;
out.print("Сумма: ");
out.print(c)
}
}
```

Типы данных

	ТИП ДАННЫХ	ДИАПАЗОН ДОПУСТИМЫХ ЗНАЧЕНИЙ	ОБЪЁМ ЗАНИМАЕМОЙ ПАМЯТИ
целочисленные	byte	от -128 до 127	1 байт
	short	от -32768 до 32767	2 байта
	int	от -2147483648 до 2147483647	4 байта
	long	от -9223372036854775808 до 9223372036854775807	8 байт
с плавающей точкой	float	от -3.4E+38 до 3.4E+38	4 байта
	double	от -1.7E+308 до 1.7E+308	8 байт
символы	char	от 0 до 65536	2 байта
логические	boolean	true или false	Для хранения значения этого типа достаточно 1 бита, но в реальности память такими порциями не выделяется, поэтому переменные этого типа могут быть по-разному упакованы виртуальной машиной

Операторы

Операторы Java:

- Арифметические
- Сравнения
- Побитовые
- Логические
- Присваивания
- Тернарный

Класс Math

Арифметические операторы

Арифметические операторы
Арифметические операторы — используются в математических выражениях таким же образом, как они используются в алгебре. Предположим, целая переменная A равна 10, а переменная B равна 20.

Оператор	Описание	Пример
+	Складывает значения по обе стороны от оператора	A + B даст 30
-	Вычитает правый операнд из левого операнда	A - B даст -10
*	Умножает значения по обе стороны от оператора	A * B даст 200
/	Оператор деления делит левый операнд на правый операнд	B / A даст 2
%	Делит левый операнд на правый операнд и возвращает остаток	B % A даст 0
++	Инкремент - увеличивает значение операнда на 1	B++ даст 21
--	Декремент - уменьшает значение операнда на 1	B-- даст 19

Операторы сравнения

Есть следующие операторы сравнения, поддерживаемые на языке Java. Предположим, переменная A равна 10, а переменная B равна 20. В следующей таблице перечислены реляционные операторы или операторы сравнения в Java:

Оператор	Описание	Пример
==	Проверяет, равны или нет значения двух операндов, если да, то условие становится истинным	(A == B) — не верны
!=	Проверяет, равны или нет значения двух операндов, если значения не равны, то условие становится истинным	(A != B) — значение истинна
>	Проверяет, является ли значение левого операнда больше, чем значение правого операнда, если да, то условие становится истинным	(A > B) — не верны
<	Проверяет, является ли значение левого операнда меньше, чем значение правого операнда, если да, то условие становится истинным	(A < B) — значение истинна
>=	Проверяет, является ли значение левого операнда больше или равно значению правого операнда, если да, то условие становится истинным	(A >= B) — значение не верны
<=	Проверяет, если значение левого операнда меньше или равно значению правого операнда, если да, то условие становится истинным	(A <= B) — значение истинна

Побитовые операторы

Java определяет несколько побитовых операторов, которые могут быть применены для целочисленных типов: int, long, short, char и byte. В Java побитовый оператор работает над битами и выполняет операцию бит за битом. Предположим, если a = 60; и b = 13; то в двоичном формате они будут следующие:

a = 0011 1100
b = 0000 1101

Побитовые операторы

Оператор	Описание	Пример
& (побитовое и)	Бинарный оператор AND копирует бит в результат, если он существует в обоих операндах.	(A & B) даст 12, который является 0000 1100
| (побитовое или)	Бинарный оператор OR копирует бит, если он существует в любом из операндов.	(A | B) даст 61 который равен 0011 1101
^ (побитовое логическое или)	Бинарный оператор XOR копирует бит, если он установлен в одном операнде, но не в обоих.	(A ^ B) даст 49, которая является 0011 0001
~ (побитовое дополнение)	Бинарный оператор дополнения и имеет эффект «отражения» бит.	(~ A) даст -61, которая является формой дополнением 1100 0011 в двоичной записи
<< (сдвиг влево)	Бинарный оператор сдвига влево. Значение левых операндов перемещается влево на количество бит, заданных правым операндом.	A << 2 даст 240, который 1111 0000
>> (сдвиг вправо)	Бинарный оператор сдвига вправо. Значение правых операндов перемещается вправо на количество бит, заданных левых операндом.	A >> 2 даст 15, который является 1111
>>> (нулевой сдвиг вправо)	Нулевой оператор сдвига вправо. Значение левых операндов перемещается вправо на количество бит, заданных правым операндом, а сдвинутые значения заполняются нулями.	A >>> 2 даст 15, который является 0000 1111

Логические операторы

Предположим, логическая переменная A имеет значение true, а переменная B хранит false. В следующей таблице перечислены логические операторы в Java:

Оператор	Описание	Пример
&&	Называется логический оператор «И». Если оба операнда являются не равны нулю, то условие становится истинным	(A && B) — значение false
||	Называется логический оператор «ИЛИ». Если любой из двух операндов не равен нулю, то условие становится истинным	(A || B) — значение true
!	Называется логический оператор «НЕ». Использование меняет логическое состояние своего операнда. Если условие имеет значение true, то оператор логического «НЕ» будет делать false	!(A && B) — значение true

Операторы присваивания

Оператор	Описание	Пример
=	Простой оператор присваивания, присваивает значения из правой стороны операндов к левому операнду	C = A + B, присвоит значение A + B в C
+=	Оператор присваивания «Добавления», он присваивает левому операнду значения правого	C += A, эквивалентно C = C + A
-=	Оператор присваивания «Вычитания», он вычитает из правого операнда левый операнд	C -= A, эквивалентно C = C - A
*=	Оператор присваивания «Умножение», он умножает правый операнд на левый операнд	C * = A эквивалентно C = C * A
/=	Оператор присваивания «Деление», он делит левый операнд на правый операнд	C /= A эквивалентно C = C / A
%=	Оператор присваивания «Модуль», он принимает модуль, с помощью двух операндов и присваивает его результат левому операнду	C %= A, эквивалентно C = C % A
<<=	Оператор присваивания «Сдвиг влево»	C << = 2, это как C = C << 2
>>=	Оператор присваивания «Сдвиг вправо»	C >>= 2, это как C = C >> 2
&=	Оператор присваивания побитового «И» («AND»)	C &= 2, это как C = C & 2
^=	Оператор присваивания побитового исключающего «ИЛИ» («XOR»)	C ^= 2, это как C = C ^ 2
|=	Оператор присваивания побитового «ИЛИ» («OR»)	C |= 2, это как C = C | 2

Тернарный оператор

Тернарный оператор — оператор, который состоит из трех операндов и используется для оценки выражений типа boolean. Тернарный оператор в Java также известен как условный оператор. Цель тернарного оператора или условного оператора заключается в том, чтобы решить, какое значение должно быть присвоено переменной.

```
переменная x = (выражение) ? значение if true : значение if false
```

Программирование на языке Java

Математические функции

Для выполнения сложных математических вычислений в пакете java.lang имеется класс Math, который содержит полезные так называемые статические методы.

Метод	Описание
Math.acos(x)	Арккосинус x
Math.asin(x)	Арксинус x
Math.atan(x)	Арктангенс x
Math.cos(x)	Косинус x
Math.sin(x)	Синус x
Math.tan(x)	Тангенс x
Math.abs(x)	Модуль (абсолютное значение) x
Math.pow(x, y)	Возводит число x в степень y
Math.sqrt(x)	Квадратный корень из x

Спасибо!

SAMSUNG