

Типы данных с плавающей точкой: float и double — введение в Java 003

Числа с плавающей точкой

Для хранения дробных чисел в Java используют float и double. Правильное название — простые (примитивные) типы данных с плавающей точкой. Примитивные типы данных — byte, short, int, long, которые являются целыми числами, — мы рассматривали [здесь](#).

Float

Давайте сразу рассмотрим практический пример:

```
public class TestFloat {
    public static void main(String[] args) {
        float a = 3.5f;
        float b = 4.5F;
        float c = (float) 5.5;
        float d = a + b + c;
        System.out.println("a+b+c: " + d);
    }
}
```

Здесь использованы три способа объявления переменных типа float. Самое важное при работе с ними — избавиться от "запятых". Java — американский язык программирования, и в американской математике дробь отделяется точкой.

Double

Давайте отдельно рассмотрим пример деления с целыми и дробными числами. Сначала целые числа:

```
public class IntDiv {
    public static void main(String[] args) {
        int a, b, c;
        a = 1;
        b = 10;
        c = a / b;
        System.out.println(c);
    }
}
```

При делении единицы на десять мы получаем ноль целых и, при использовании типа данных **int**, после и... больше ничего не будет. Деление единицы на десять даёт ноль.

- Сколько айфонов ценой в 1000€ я могу купить за 10€?
- Ноль. За 10€ я куплю ноль айфонов. И за 999€ я куплю ровно ноль айфонов.

Результат деления целых чисел (тип данных int) округляется до меньшего целого числа.

```
public class IntDiv {
    public static void main(String[] args) {
        double a, b, c;
```

```

        a = 1;
        b = 10;
        c = a / b;
        System.out.println(c);
    }
}

```

Но, если мы при инициализации переменных изменим их тип на **double**, то получим привычное нам **0.1**.

Представьте, что вы купили десять метров ткани, и на костюм вам надо 7 метров. Сколько костюмов вы сошьёте из 10 метров? Один или полтора? В некоторых случаях нам абсолютно не нужны дробные (с плавающей точкой) числа, а в некоторых — целые.

На простейшем примере можно увидеть работу double ещё раз:

```

public class HalloDouble {
    public static void main(String[] args) {
        final double PI = 3.14;
        // final - используется для "финализации" переменной,
        // теперь переменной PI нельзя присвоить другое значение.
        double r = 5.5;
        double k = r * r;
        double s = PI * r * r;
        System.out.println(k);
        // промежуточный результат
        System.out.println(s);
    }
}

```

Тип	Максимальное позитивное значение	Минимальное позитивное значение	Размер (бит)
float	3.4028234663852886E+038	1.4012984643248171E-045	32
double	1.7976931348623157E+308	4.9406564584124654E-324	64

Экспоненциальная запись

Экспоненциальная запись — представление действительных чисел в виде мантиссы (дробной части логарифма числа) и порядка. Удобна при представлении очень больших и очень малых чисел, а также для унификации их написания.

1_000_000 - один миллион: 1,0E+6 соответствует один умножить на десять в шестой степени.

Что использовать нам в программе? Float или double

У меня два ответа на этот вопрос.

Вариант один. Пока вы учитесь, используйте **double**. Он проще не требует буковок после цифры или дополнительного кастинга.

Вариант два. Производительность программы с **float** ровно в два раза выше, чем с double. Если вам важно делать свой код изящнее, то вы всегда должны понимать, какие типы данных уместнее использовать.

Давайте попробуем сделать программу, которая считает степень двойки.

```

public class HalloDouble2 {
    public static void main(String[] args) {

```

```

        double a = 2;
        double b = 30;
        double c = Math.pow(a, b);
        System.out.println(c);
        // Выводит на экран 1.073741824E9
    }
}

```

Достаточно интересный вопрос: что происходит при пересечении разных типов данных? Что произойдёт, если тип переменных **int** встретится с **double**? Я подготовил программу без единой строчки комментария. Попробуйте предугадать то, что Java выведет на экран.

```

public class IntToDouble {
    public static void main(String[] args) {

        int var1 = 5;
        int var2 = 2;
        int var3 = var1 / var2;

        double var4 = 5.0;
        double var5 = 2.0;
        double var6 = var4 / var5;
        double var7 = var1 / var2;

        double var25 = var1 / var4;

        System.out.println("Int Var3 = " + var3);
        System.out.println("double Var6 = " + var6);
        System.out.println("double Var7 = " + var7);
        System.out.println("double Var25 = " + var25);

        int var11 = (int) (var1 / var4);

        System.out.println("Int Var11 = " + var11);
        System.out.println("Магия " + 5.0/var2);
    }
}

```

Результат работы IntToDouble:

1. Int Var3 = 2
2. double Var6 = 2.5
3. double Var7 = 2.0
4. double Var25 = 1.0
5. Int Var11 = 1
6. Магия = 2.5

Результат работы подтверждает, что **int** находится в пределах **double** и легко может использоваться в расчётах, где результатом будет число типа **double**. К сожалению, обратное невозможно. Мы должны преобразовать тип данных — произвести **Casting**. Пример кастования из нашего IntToDouble: `int var11 = (int) (var1 / var4)`, где `int` в скобках — и есть преобразование.

Помимо вычисления степени числа, математическая библиотека Java (Math) предлагает нам метод, который генерирует случайное число. Дотошный разработчик скажет, что это не случайное число, а псевдослучайное, но на нашем уровне обучения псевдослучайности достаточно для работы в Java.

Давайте, в качестве примера, сгенерируем число от 30 до 100:

```
public static void main(String[] args) {  
    int min = 30;  
    int max = 100;  
    int result = (int) (Math.random() * ((max - min) + 1)) + min;  
    System.out.println(result);  
}
```

Метод `Math.random()` представляет результат в виде дробных чисел типа `double`. Нам нужны целые числа, поэтому мы "кастуем" результат в нужный тип данных.

Метод `Math.random()` генерирует дробные числа между **0 и 1**. Поэтому мы с помощью арифметики приводим случайное число между нулём и единицей в случайное число между 30 и 100. Попробуйте получить число между 1 и 50.

Поэкспериментируйте с этим методом. Он нам понадобится не один раз.

Метод `Math.random()` является одним из многих подготовленных для вас методов, которые часто требуются при программировании. Возведение в степень, число π , логарифмы и многое другое. Вы можете ознакомиться с официальной документацией к библиотеке `Math` — [здесь](#)

Остался, может быть, вопрос, что использовать? Если нет особой надобности в учебных целях или для скорости разработки — используйте `double` и `int`. Остальные типы данных требуют более бережного отношения и внимания.

Домашнее задание

1. Написать программу для вычисления площади круга.
2. Найти длину окружности.
3. Написать формулу вычисления стоимости Евро, если курс Евро к Доллару 1,1.
4. Найти соотношение площади Беларуси к площади Украины.
5. Найти, во сколько раз площадь Москвы больше площади Берлина.
6. Если клиент положил в банк 1000€ под 3,5%, какая сумма будет его ждать через семь лет?

Дополнительные материалы

1. Полезная библиотека - <https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html>
2. Конвенция java (договор о правописании и синтаксисе) - <https://www.oracle.com/technetwork/articles/javaee/codeconventions-135099.html>
3. Округление чисел - https://geekbrains.ru/posts/java_round