

# Алгоритмы и программирование

## Модуль 1

## Занятие 2

# Типы данных и отладка

Гринчишин Михаил Александрович

Преподаватель

Клуб программистов 1С



# Задачи Java

<http://informatics.mccme.ru/>

Дистанционная подготовка / ► 1С / ►

Занятие 2. Задачи: ЗАДАЧА А.

**Сумма от 1 до N**

**Входные данные**

Одно целое число  $N$ .

**Выходные данные**

Одно число – сумма всех целых чисел от 1 до  $N$ .

**Примеры:** входные данные **100**; выходные данные **5050**.



# Создание первого проекта Java

## Задача:

### Первая задача:

```
import java.io.*; // библиотека для PrintWriter
import java.util.*; // библиотека для Scanner
public class prog1 {
public static void main(String[] args) {
Scanner in= new Scanner(System.in);
PrintWriter out = new PrintWriter(System.out);
long S = 0; // результат
long N = in.nextInt(); // ввод a
//решение
out.println(S); //ВЫВОД ОТВЕТА
out.flush(); in.close;}}//
```



## Задачи Java

**Очевидное решение:** Сумма от 1 до N

```
for (int i = 1; i <= N; i++) {  
    S += i;  
}
```

Результат: Test 3: Wrong answer

«А! ЦЕЛОЕ, ЭТО МОЖЕТ БЫТЬ И ОТРИЦАТЕЛЬНЫЕ!»



## Задачи Java

**Очевидное решение:** Сумма от 1 до N

```
if (N < 0) {K = -N; }
```

```
for (int i = 1; i <= K; i++) { s += i; }
```

```
if (N < 0)
```

Результат: Test 4: Time Limit

```
{ s = -s + 1; }
```

«Тип long, это же больше  $10^{18}$ , это же очень долго.

Надо не циклом, а по формуле!»



## Задачи Java

**Очевидное решение:** Сумма от 1 до N

```
int N = in.nextInt();
```

```
int s = 0;
```

```
s = N * (N + 1) / 2;
```

```
if (N < 0)
```

Результат: Test 5: Wrong Answer

```
{s = -s + 1;}
```

«Надо не int, а long!»



## Задачи Java

**Очевидное решение:** Сумма от 1 до N

```
long N = in.nextLong(); long s = 0;
```

```
s = N * (N + 1) / 2;
```

```
if (N < 0)
```

```
{s = -s + 1;}
```

Результат: Test 6: Wrong Answer

«Сначала делить на 2, потом умножать!»



## Задачи Java

**Очевидное решение:** Сумма от 1 до N

```
long N = in.nextLong();
```

```
long s = 0;
```

```
s = N * ((N + 1) / 2);
```

```
if (N < 0)
```

Результат: Test 8: Wrong Answer

```
{s = -s + 1;}
```

«Делить надо то, что делится!»

N – Если четное, N-1, если нечетное



## Задачи Java

**Очевидное решение:** Сумма от 1 до N

```
long N = in.nextLong();
```

```
long s = 0;
```

```
if (N % 2 == 0)
```

**Результат: ОК**

```
{s = (N / 2) * (N + 1);}
```

```
else {s = N * ((N + 1) / 2); }
```

```
if (N < 0) {s = -s + 1; }
```



# Создание первого проекта Java

Создание первого проекта – работа с тестирующей системой (результаты работы)

## 1. Первая задача: $A+B=?$

- **OK** — программа прошла все тесты, решение верное.
- **Wrong Answer** (Неправильный ответ, WA) — программа прошла не все тесты, то есть работает не во всех случаях. В этом случае в графе «Ошибка на тесте» показывается номер теста, на котором программа выдаёт неверный ответ.



# Создание первого проекта Java

Создание первого проекта – работа с тестирующей системой (результаты работы)

## 1. Первая задача: $A+B=?$

- **OK** — программа прошла все тесты, решение верное.
- **Wrong Answer** (Неправильный ответ, WA) — программа прошла не все тесты, то есть работает не во всех случаях. В этом случае в графе «Ошибка на тесте» показывается номер теста, на котором программа выдаёт неверный ответ.



# Создание первого проекта Java

Создание первого проекта – работа с тестирующей системой (результаты работы)

## 1. Первая задача: A+B=?

- **Presentation Error** (Неправильный формат вывода, PE) — означает, что на каком-то тесте программа выводит ответ не в том формате, как это требуется в условии задачи (например, выводит несколько чисел, когда требуется одно, или выводит слово, когда требуется число).
- **Runtime Error** (Ошибка выполнения RE) — означает, что на каком-то тесте программа выполняет недопустимую операцию (например, происходит деление на 0, выход за пределы массива или иная ошибка, которая может привести к аварийному завершению программы)



# Создание первого проекта Java

Создание первого проекта – работа с тестирующей системой (результаты работы)

## 1. Первая задача: A+B=?

- **Memory Limit** (Превышен предел по памяти, ML) — превышен максимальный объём памяти, выделяемый для программы по условию.
- **Time Limit** (Превышен предел времени выполнения программы, TL) — программа работает слишком долго.
- **Compile Error** (Ошибка компиляции, CE) — означает, что программа содержит синтаксические ошибки из-за чего тестирующая система не способна её откомпилировать и запустить на проверку.



# Создание первого проекта Java

## Основные правила оформления решений

1. Программа-решение должна чётко соблюдать формат входных и выходных данных.
2. Программа-решение должна отработать (закончить выполнение) быстро.
3. Входные данные всегда корректны

**Ограничения на входные данные учитываются при разработке программы, но никак не записываются в программу**



# Задачи Java

<http://informatics.mccme.ru/>

**Задачи для тех, кто сдал контест  
первого занятия:**

Дистанционная подготовка по информатике

Избранное:

Кои  
По  
так  
не  
или

207 241 174 1440

1693

Оси

Мои группы ☆  
Мои настройки ☆  
Сообщения об ошибках ☆  
Топ идеальных решений ☆  
Рейтинг пользователей ☆

Особенности языков программирования ☆  
Составление тестов к задачам ☆  
Учим python (в стадии разработки) ☆

Авторские курсы (5) 📁

Поиск

Google™ Пользовате: Поиск ×  
технология Google™

К задаче №  
 Перейти

Новостной форум

27 May 20:28: Машинное обучение в летней школе ☆  
3 May 22:40: Всероссийская олимпиада школьников 2016 ☆  
2 Apr 17:13: Московская



# Простые типы данных Java

## Переменные целого типа

`int a = 0;` // целая переменная a, значение 0

Имя	Разрядность (байт)	Диапазон
long	64	-9, 223, 372, 036, 854, 775, 808.. 9, 223, 372, 036, 854, 775, 807
int	32	-2, 147, 483, 648.. 2, 147, 483, 647
short	16	-32, 768.. 32, 767
byte	8	-128.. 127

`long b = in.nextInt();` - **неправильно!!**

`long b = in.nextLong();` - **правильно**



Что если число больше типа long?



# Простые типы данных Java

## Переменные вещественного типа

```
float b = 0.5; // вещественная b, значение
```

Имя	Разрядность (байт)	Диапазон
double	64	$1.7e-308.. 1.7e+308$
float	32	$3.4e-038.. 3.4e+038$

```
in.useLocale(Locale.US);
```

```
float b = in.nextFloat();
```



# Простые типы данных Java

## Переменные символьного типа

`char c = 'A';` //символ, кавычки одинарные

`String st = "st";` //строка, кавычки двойные

Имя	Разрядность (байт)	Диапазон
Char	1	Символы
String	класс!!!	Набор символов (нельзя сравнивать)

`Char c = in.nextChar();`



# Арифметические операции Java

Оператор	Результат	Оператор	Результат
+	Сложение	+ =	сложение с присваиванием
-	Вычитание (также унарный минус)	--	вычитание с присваиванием
*	Умножение	* =	умножение с присваиванием
/	Деление	/=	деление с присваиванием
%	Деление (остаток)	%=	деление по модулю с присваиванием
++	Инкремент	--	декремент



# Арифметические операции Java

## Деление

Результат деления целого на целое—**целое** число (остаток отбрасывается):

```
int a = 3, b = 4;
float x;
x = 3 / 4; // = 0
x = 3. / 4; // = 0.75
x = 3 / 4.; // = 0.75
x = a / 4; // = 0
x = a / 4.; // = 0.75
x = a / b; // = 0
x = float(a) / 4; // = 0.75
x = a / float(b); // = 0.75
```



# Арифметические операции Java

## Остаток от деления - %

```
int a, b, d;  
d = 85;  
b = d / 10;  
a = d % 10;  
d = a % b;  
d = b % a;
```

Для отрицательных чисел:

```
int a = -7;  
b = a / 2;  
d = a % 2;
```



В математике не  
так!

остаток  $\geq 0$

$$-7 = (-4) * 2 + 1$$



# Арифметические операции Java

## Сокращенная запись операций

```
int a, b;  
...  
a ++;      // a = a + 1;  
a --;      // a = a - 1;  
a += b;    // a = a + b;  
a -= b;    // a = a - b;  
a *= b;    // a = a * b;  
a /= b;    // a = a / b;  
a %= b;    // a = a % b;
```



# Операции объекта Math

Используется: **Math.операция(аргументы)**

**Math.abs (x)** – модуль числа

**Math.sqrt (x)** – квадратный корень

**Math.sin (x)** – синус угла, заданного **в радианах**

**Math.cos (x)** – косинус угла, заданного **в радианах**

**Math.exp (x)** – экспонента  $e^x$

**Math.ln (x)** – натуральный логарифм

**Math.pow (x, y)** –  $x^y$ : возведение числа  $x$  в степень  $y$

**Math.floor (x)** – округление «вниз»

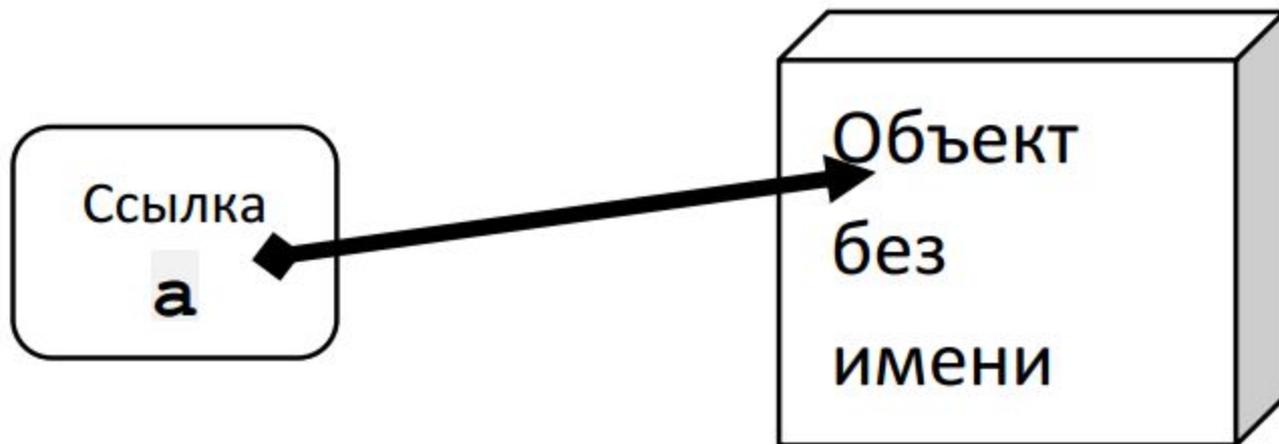
**Math.ceil (x)** – округление «вверх»



# Операции с объектами

## Используется: ссылка на объект

С любыми объектами в Java программисты работают при помощи ссылок.





## Операции с объектами

**Используется: ссылка на объект**

Чаще всего сравнение объектов при помощи операции равенства (==) является ошибкой!

```
String s1 = "abc";  
String s2 = "abcd";  
s2 = s2.substring(0, 3); // создается новая строка!  
if (s1 == s2) {out.println(s1 + " equals " + s2); }  
else {out.println(s1 + " does NOT equal " + s2); }
```

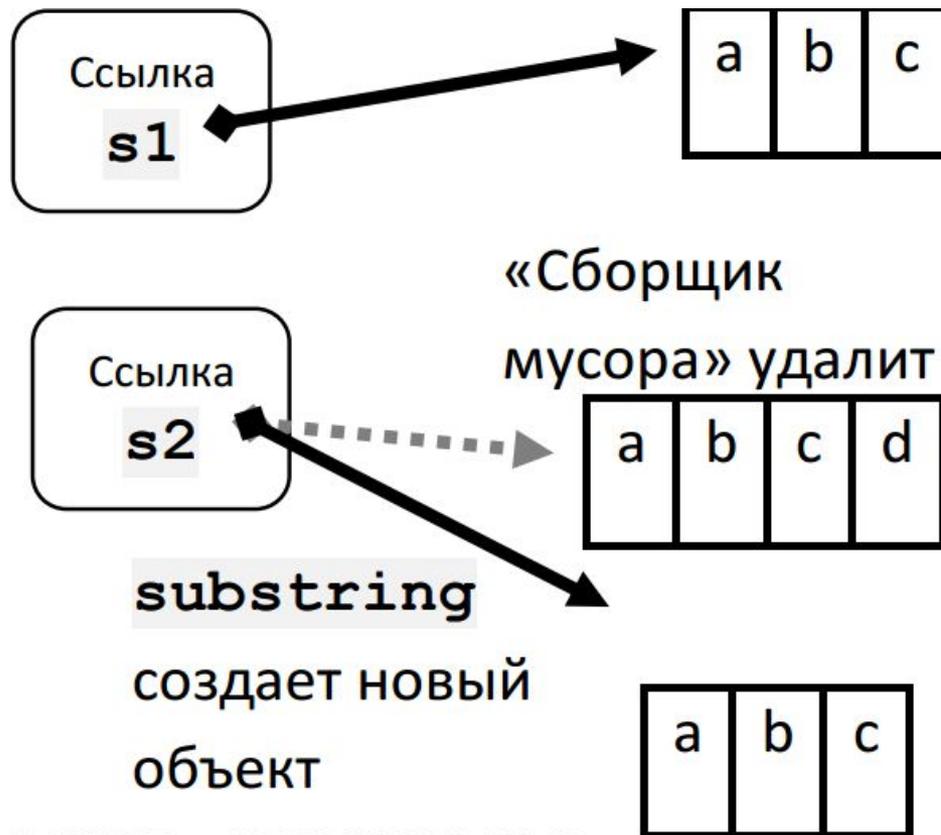
**Вывод: abc does NOT equal abc**



# Операции с объектами

## Используется: ссылка на объект

Для корректного сравнения объектов используют специальные функции.





## Операции с объектами Объекты **BigInteger** и **BigDecimal**

Во-первых, в Java не разрешена перегрузка операций.

Для того, чтобы, скажем, сложить два объекта **BigInteger**, нужно пользоваться не операцией «+», а функцией **add**.



## Операции с объектами Объекты `BigInteger` и `BigDecimal`

Во-вторых, в результате работы функций реализующих арифметические операции исходное число не меняется, а создается и возвращается новое число – результат операции.

Например, код:

```
BigInteger b = new BigInteger("1");  
b.add(b); // b плюс b out.println(b);
```

**Выведет 1**

Правильно

```
b = b.add(b);
```



## Операции с объектами Объекты `BigInteger` и `BigDecimal`

Во-вторых, в результате работы функций реализующих арифметические операции исходное число не меняется, а создается и возвращается новое число – результат операции.

Например, код:

```
BigInteger b = new BigInteger("1");  
b.add(b); // b плюс b out.println(b);
```

**Выведет 1**

Правильно

```
b = b.add(b);
```



# Создание первого проекта Java

Соедините любые три высказывания по итогам занятия

Буду  
применять

Можно, но  
трудно

Этому нет  
предела

Я приму к  
сведению

Всем желаю  
успеха

Это интересно

Это  
малоэффективно

Надо расти и  
совершенствоваться

У нас все  
получится

Это неактуально