



# Начальный курс Java

8 занятий



# Занятие 1

# Java

<http://prog.kiev.ua/forum>

- \* сюда надо выкладывать «домашки» на проверку
- \* тут можно скачать учебные материалы

Также можно писать в Skype:  
**prog.kiev.ua**

# Java

**ВНИМАНИЕ!!! ЗАПИШИТЕ!!!**

Пароль на презентацию курса Java Start:

**java-1-0626**

# Java

- \* <https://www.facebook.com/prog.kiev.ua>
- \* [https://vk.com/prog\\_kiev\\_ua](https://vk.com/prog_kiev_ua)
- \* <https://plus.google.com/+ProgKievUa>
- \* <https://twitter.com/KievProg>

Подписывайтесь. Тут можно найти:

- \* Новости и объявления по нашим курсам
- \* Ссылки на учебные материалы
- \* Вакансии

# Java

Superjob.ua:

“... Средние рыночные зарплатные предложения для Java-программистов в Киеве составляют 24 000 грн. Максимальный заработок программистов Java в столице составляет 40 000 грн.”

22.01.2014

# Java

Код = текст. Пример:

```
public class MyClass {  
    public static void main(String[] args) {  
        System.out.println("Куку!");  
    }  
}
```

Код  Компилятор  Готовая программа

# Java

- \* Java — объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (куплена Oracle).
- \* Дата выпуска – 23 мая 1995 г.
- \* Java 1 – 8

# Java

- \* Программы на Java транслируются в байт-код JVM
- \* Независимость байт-кода от операционной системы и оборудования (Windows, Mac OS X, Linux etc.)
- \* Автоматическое управление памятью (GC)
- \* Гибкая система безопасности
- \* Скорость

# Java

## Платформы:

- \* Java SE (J2SE)
- \* Java EE (J2EE)
- \* Java ME (J2ME)
- \* JavaFX
- \* Java Card
- \* Android
- \* Google App Engine (GAE/Java)

# Java

## Проекты:

- \* eBay
- \* LinkedIn
- \* Yahoo
- \* Privat24
- \* Одноклассники

# Java

## Дистрибутивы:

- \* JRE
- \* JDK

## Библиотеки:

- \* Стандартная библиотека
- \* Сторонние библиотеки (Spring, Hibernate... )

## Средства разработки:

- \* Eclipse
- \* NetBeans
- \* IntelliJ IDEA
- \* Блокнот 😊

# Java

## Структура:

- \* Class-ы
- \* Package - группа взаимосвязанных классов
- \* JAR – группа пакетов

## Проекты:

- \* .java файлы
- \* Иерархия каталогов

# Java

Компиляция программы:

\*.java -> компилятор -> \*.class -> JAR

# Первая программа

```
package test; // можно опустить
```

```
public class MyClass {  
    public static void main(String[] args) {  
        // этот текст появится на экране  
        System.out.println("Prog.kiev.ua is the best!");  
    }  
}
```

# КОМПИЛЯЦИЯ

1. Сохраняем в файл MyClass.java
2. `javac -d ./ *.java`
3. `jar -cf my.jar test`
4. Запускаем: `java -cp my.jar test.MyClass`
5. Открываем JAR, добавляем в файл META-INF/MANIFEST.MF строку “Main-Class: test.MyClass”
6. Запускаем: `java -jar my.jar`

# Усложняем...

```
package test;
```

```
public class MyClass {
```

```
    public static void main(String[] args) {
```

```
        Runtime r = Runtime.getRuntime();
```

```
        System.out.println(System.getProperty("os.name"));
```

```
        System.out.println(System.getProperty("os.version"));
```

```
        System.out.println(r.availableProcessors());
```

```
        // total amount of memory in the Java virtual machine.
```

```
        System.out.println(r.totalMemory());
```

```
    }
```

```
}
```

# IDE

Запускаем Eclipse -> создаем  
проект...

# Документация

[Overview](#) [Package](#) **[Class](#)** [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[Prev Class](#) [Next Class](#) [Frames](#) [No Frames](#) [All Classes](#)

Summary: [Nested](#) | [Field](#) | [Constr](#) | [Method](#)    Detail: [Field](#) | [Constr](#) | [Method](#)

`java.lang`

## Class System

`java.lang.Object`  
`java.lang.System`

---

```
public final class System
extends Object
```

The `System` class contains several useful class fields and methods. It cannot be instantiated.

Among the facilities provided by the `System` class are standard input, standard output, and error output streams; access to externally defined procedures for quickly copying a portion of an array.

### Since:

JDK1.0

# Лирическое отступление

Размерность данных:

- \* 1 бит : 0 или 1
- \* 1 байт = 8 бит (10101110)
- \* 1 килобайт = 1024 байт
- \* 1 мегабайт = 1024 килобайт
- \* 1 гигабайт = 1024 мегабайт
- \* 1 терабайт = 1024 гигабайт

# Hex

Hex = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

Пример 1:

- \*  $0x5A3 = 3 \cdot 16^0 + 10 \cdot 16^1 + 5 \cdot 16^2 = 1443$

Пример 2:

- \*  $A = 1443 / 16 = 90, B = 1443 \% 16 = 3$

- \*  $A = 90 / 16 = 5, B = 90 \% 16 = 10 = A$

- \*  $A = 5 / 16 = 0 \rightarrow \text{stop} \rightarrow 0x5A3$

# Binary

$V = \{0, 1\}$

Пример 1:

\*  $b101011 = 1 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^3 + 1 \cdot 2^5 = 43$

Пример 2:

\*  $A = 43 / 2 = 21, B = 43 \% 2 = 1$

\*  $A = 21 / 2 = 10, B = 21 \% 2 = 1$

\*  $A = 10 / 2 = 5, B = 10 \% 2 = 0$

\*  $A = 5 / 2 = 2, B = 5 \% 2 = 1$

\*  $A = 2 / 2 = 1, B = 2 \% 2 = 0$

\*  $A = 1 / 2 = 0 \rightarrow \text{STOP!!} \rightarrow b101011$

# Java

Домашнее задание:

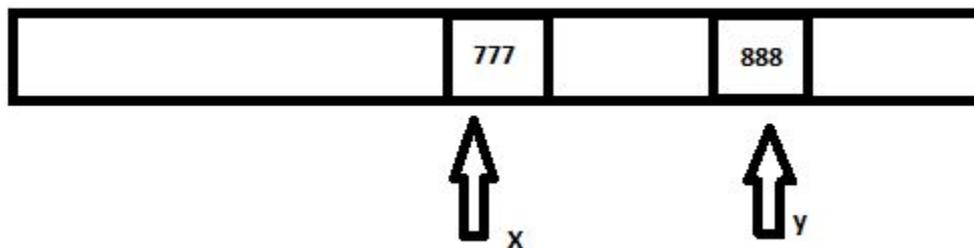
1. Разобраться с «ручной компиляцией» программ без IDE.
2. Разобраться с 16-й системой исчисления.



# Занятие 2

# Переменные

**Переменная** – именованная область памяти которую можно использовать для осуществления доступа к данным. Данные, находящиеся в переменной, называются значением этой переменной.



# Типы данных

- \* byte: 8 бит (от -128 до 127)
- \* short: 16 бит (от -32 768 до 32 767)
- \* int: 32 бита (от -2 147 483 648 до 2 147 483 647)
- \* long: 64 бита (от -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807)
- \* float: 32 бита ( $-(2 \cdot 2^{-23}) \cdot 2^{127} .. (2 \cdot 2^{-23}) \cdot 2^{127}$ , или примерно  $-3.4 \cdot 10^{38} .. 3.4 \cdot 10^{38}$ , а также  $-\infty$ ,  $+\infty$ , NaN)

Float.POSITIVE\_INFINITY, Float.NEGATIVE\_INFINITY, Float.NaN

- \* double: 64 бита ( $-(2 \cdot 2^{-52}) \cdot 2^{1023} .. (2 \cdot 2^{-52}) \cdot 2^{1023}$ , или примерно  $-1.8 \cdot 10^{308} .. 1.8 \cdot 10^{308}$ , а также  $-\infty$ ,  $+\infty$ , NaN)

Double.POSITIVE\_INFINITY, Double.NEGATIVE\_INFINITY, Double.NaN

(<http://docs.oracle.com/javase/specs/jls/se7/html/jls-4.html#jls-4.2.3>)

- \* char: 16 бит (0..65535)
- \* boolean: true/false

# Переменные

```
package test;
```

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

```
        int x;
```

```
        int a = 1;
```

```
        double b = 7.5;
```

```
        long c = 100000, d = 33333;
```

```
        char d1 = 'A';
```

```
        boolean e = true;
```

```
        short f = 500;
```

```
        int g = f + 300;
```

```
        x = 55;
```

```
    }
```

```
}
```

# Переменные

- \* `int i1 = 12;` // переменной `i` присвоить значение 12 в десятичной системе счисления
- \* `int i2 = 0x12;` // переменной `i` присвоить значение 12 в шестнадцатеричной системе счисления
- \* `int i3 = 012;` // переменной `i` присвоить значение 12 в восьмеричной системе счисления
- \* `int i4 = 0b1011;` // переменной `i` присвоить значение 1011 в двоичной системе счисления (с Java 7)

# Преобразования

Правила:

- \* Если один операнд имеет тип `double`, другой тоже преобразуется к типу `double`.
- \* Иначе, если один операнд имеет тип `float`, другой тоже преобразуется к типу `float`.
- \* Иначе, если один операнд имеет тип `long`, другой тоже преобразуется к типу `long`.
- \* Иначе оба операнда преобразуются к типу `int`.

# Преобразования

```
package test;
```

```
public class MyClass {  
    public static void main(String[] args) {  
        int x = 1; // превратится в double x = 1.0  
        double y = 2.3;  
        double z = x + y - 5.2;  
    }  
}
```

# Преобразования

```
public class MyClass {  
    public static void main(String[] args) {  
        byte a = (byte)(127 + 1);  
        short x = Byte.MAX_VALUE + 1; /* 127 + 1 = 128 */  
        byte y = (byte)x;  
        byte z = (byte)(x + 129);  
  
        System.out.println(a); // -128 !!! Переполнение  
        System.out.println(x); // 128 // Все ОК  
        System.out.println(y); // -128  
        System.out.println(z); // 1  
    }  
}
```

-128, -127, -126, -125, ... 0, 1, 2 ..., 127

# Типы данных

**String** – строки. Индексация – с нуля.

```
public class MyClass {
    public static void main(String[] args) {
        String s1 = "abCd";
        String s2 = "12345";
        String s3 = s1 + s2; // "abCd12345"
        String s4 = s1.substring(2); // "Cd"
        int x = s2.length(); // 5
        String s5 = s1.toUpperCase(); // "ABCD"
        char c = s1.charAt(0); // 'a'
        int pos = s1.indexOf("bC"); // 1
        String s6 = "aaacccddd".replace("ccc", "qqq"); // "aaaqqqddd"

        System.out.println(s3);
        System.out.println(s4);
        System.out.println(s5);
    }
}
```

# Типы данных

## Сравнение строк:

```
package test;
```

```
public class MyClass {  
    public static void main(String[] args) {  
        String s1 = "ABCD";  
        String s2 = "abcd";  
        String s3 = "abcd";  
  
        System.out.println(s3.equals(s1));  
        System.out.println(s1.equals(s2));  
        System.out.println(s2.equals(s1));  
        System.out.println(s1.equalsIgnoreCase(s2));  
    }  
}
```

# Преобразования

Классы:

- \* Integer, Long, Short, Boolean, Character, Byte.

Преобразование числа в строку:

- \* `String s1 = Integer.toString(777); "777"`
- \* `String s2 = Long.toString(888888);`
- \* `String s3 = Integer.toString(222, 16); // "DE"`

Преобразование строки в число:

- \* `int a = Integer.parseInt("4455");`
- \* `long b = Long.parseLong("2343554623453");`
- \* `int c = Integer.parseInt("DE", 16);`

# Типы данных

```
package test;
```

```
public class MyClass {
```

```
    public static void main(String[] args) {
```

```
        String s = "String";
```

```
        int x = 100;
```

```
        System.out.println(s + x); // "String 100"
```

```
        System.out.println(s + Integer.toString(x)); // "String 100"
```

```
    }
```

```
}
```

# Переменные

## Область видимости переменных:

```
public class MyClass {  
    public static void main(String[] args) {  
        short x;  
  
        if (true) {  
            short x;  
            int y;  
  
            y = 60;  
            x = 33;  
        }  
  
        y = 50; // ошибка!  
        x = 22;  
    }  
}
```

# Операторы

- \*  $+$  ( $+=$ ) - сложение
- \*  $-$  ( $-=$ ) - вычитание
- \*  $*$  ( $*=$ ) - умножение
- \*  $/$  ( $/=$ ) - деление
- \*  $\%$  ( $\%=$ ) - деление по модулю
- \*  $++$  - инкремент
- \*  $--$  - декремент

# Операторы

```
int a = 1 + 34;  
int b = a / 7; // 5  
int c = a * 2; // 70  
int d = 12 % 3; // 0  
int e = 13 % 3; // 1  
e++; // 2  
a--; // 34  
c += 2; // 72 // c = c + 2;  
c -= 3; // 69
```

# Операторы

В постфиксных операциях значение сначала используется а затем изменяется, а в префиксных - наоборот.

```
int i = 0;  
System.out.print(++i); // 1  
System.out.print(i++); // 1  
System.out.print(i); // 2
```

# Как прочитать ввод

```
import java.util.Scanner;

public class Main {
    public static void main (String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter string: ");
        String s = scanner.nextLine();
        System.out.println("You printed: " + s);

        System.out.print("Enter integer: ");
        int i = scanner.nextInt();
        System.out.println("You printed: " + i);
    }
}
```

# Задачи

1. Прочитать строку 1
2. Прочитать строку 2
3. Прочитать строку 3
4. Вывести на экран
  1.  $S_1 + s_3$
  2.  $S_3 + s_2 + s_1$
  3.  $S_1 + s_2 + s_3$

# Задачи

1. Прочитать с консоли число 1
2. Прочитать с консоли число 2
3. Вывести результат
4. И так для каждого оператора

```
1. import java.util.Scanner;
2. public class Main {
3.     public static void main (String[] args) {
4.         Scanner scanner = new Scanner(System.in);
5.
6.         int x = scanner.nextInt(), y = scanner.nextInt();
7.
8.         System.out.println(x + y);
9.         System.out.println(x - y);
10.        System.out.println(x * y);
11.        System.out.println(x / y);
12.    }
13. }
```



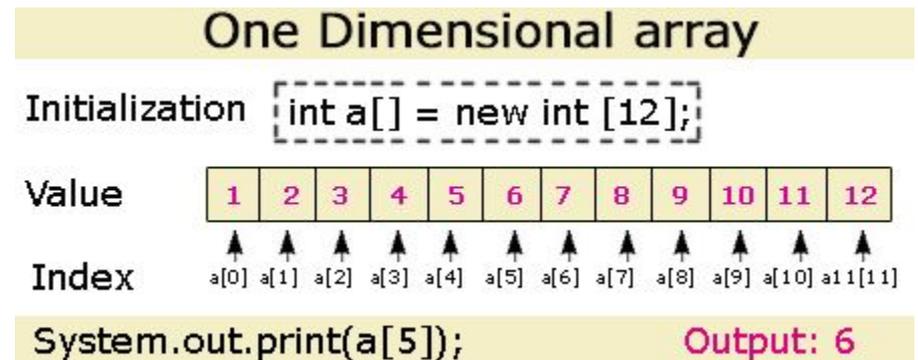
# Занятие 3

# Строки и массивы

**Массив** — это конечная последовательность упорядоченных элементов одного типа, доступ к каждому элементу в которой осуществляется по его индексу.

Варианты объявления массива:

- \* тип[] имя; // !!!
- \* тип имя[];



# Строки и массивы

Инициализация массивов:

```
int[] aro;
```

```
double[] ar1;
```

```
double ar2[] = {3.14, 2, 5.5}; // Массив из 3-х элементов типа
```

```
aro = new int[10]; // массив из 10 элементов типа int
```

```
int n = 5;
```

```
ar1 = new double[n]; // Массив из 5 элементов double
```

# Строки и массивы

```
int[] arr = new int[5];
```

## **Запись в массив:**

```
arr[0] = 0;
```

```
arr[1] = 15;
```

...

## **Чтение из массива:**

```
int x = arr[4]; // последний элемент
```

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

Длина массива:

```
int len = arr.length; // 5
```

# Строки и массивы

Если массив создается с помощью оператора `new`, то всем элементам массива автоматически присваиваются значения по умолчанию:

- \* Для числовых значений начальное значение будет 0.
- \* Для массива типа `boolean` начальное значение будет равно `false`.
- \* Для массива типа `char` - `'\u0000'`.
- \* Для массива типа класса - `null`.

# Строки и массивы

**Что за массив в main?**

```
public static void main(String[] args) {  
    System.out.println(args[0]); // MyClass.java  
    System.out.println(args[1]); // Class2.java  
}
```

```
java -jar my.jar MyClass.java Class2.java
```

# Строки и массивы

Массивы строк:

```
String[] a = new String[] {"Hello", "Java", "SE"};
```

```
System.out.println(a[0]); // "Hello"
```

```
System.out.println(a[1]); // "Java"
```

# Строки и массивы

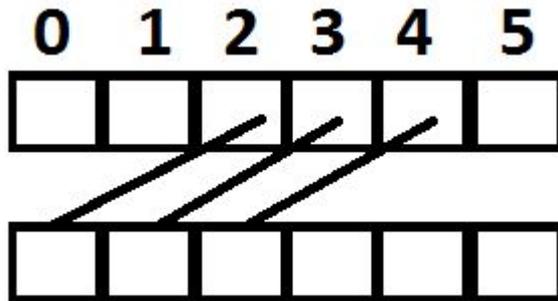
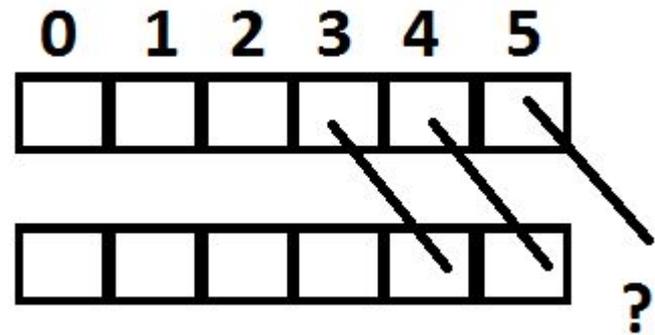
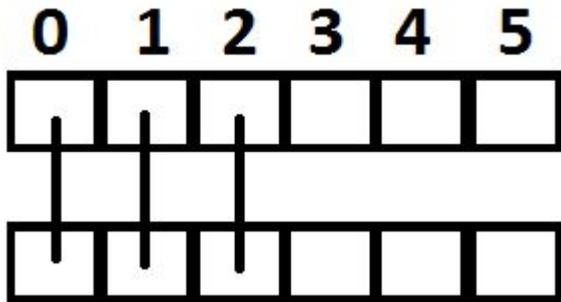
Копирование массивов:

```
char[] a = new char[5] {'A', 'B', 'C', 'D', 'E'};  
char[] b = new char[5];
```

```
System.arraycopy(a, 0, b, 0, a.length); // теперь a и b – два  
разных массива с одинаковым содержимым
```

```
char[] c1 = Arrays.copyOf(a, a.length); // c1 = c  
char[] c2 = Arrays.copyOf(a, 2); // {'A', 'B'}  
char[] c3 = Arrays.copyOfRange(a, 2, 5); // c = {'C', 'D', 'E'}
```

# System.arraycopy



# Строки и массивы

```
import java.util.Arrays;
```

```
...
```

```
int[] a = new int[100];
```

```
Arrays.fill(a, 50); // [50, 50, ... , 50]
```

```
Arrays.sort(a);
```

```
Arrays.sort(a, 50, 60);
```

```
int[] x = {1,2,3,4}, y = {1,2,3,4}, z = {5,6,7};
```

```
System.out.println(Arrays.equals(x, y)); // true
```

```
System.out.println(Arrays.equals(x, z)); // false
```

```
int n = Arrays.binarySearch(x, 3); // 2
```

```
String s = Arrays.toString(a);
```

```
System.out.println(s);
```

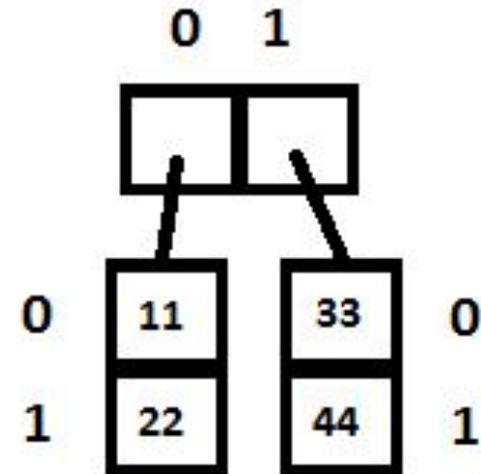
# Строки и массивы

```
int[][] a = { {11, 22}, {33, 44} };  
System.out.println(a[1][1]); // 44
```

```
11 33  
22 44
```

```
byte[][] b = new byte[2][3];  
b[1][2] = 1;
```

```
0 0      0 0  
0 0  ->->->->  0 0  
0 0      0 1
```

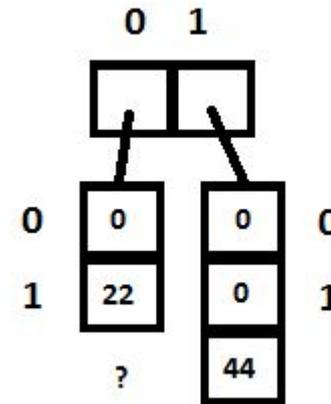


# Строки и массивы

```
int[][] n = new int[2][];  
n[0] = new int[2];  
n[1] = new int[3];
```

```
n[0][1] = 22;  
n[1][2] = 44;
```

```
0 0  
22 0  
-- 44
```



# Строки и массивы

Кодировки:

- \* US-ASCII
- \* ISO-8859-1
- \* UTF-8
- \* UTF-16BE
- \* UTF-16LE
- \* UTF-16

# Строки и массивы

Массив из строки:

```
String s = "abcde";
```

```
byte[] b1 = s.getBytes();
```

```
byte[] b2;
```

```
try {
```

```
    b2 = s.getBytes("UTF-8");
```

```
} catch (UnsupportedEncodingException ex) {
```

```
    ...
```

```
}
```

# Строки и массивы

Строка из массива:

```
byte[] b = ...; //
```

```
String s1 = new String(b);
```

```
String s2 = new String(b, "utf-16");
```

```
char[] c = {'a', 'b', 'c', 'd'};
```

```
String s3 = new String(c); // "abcd"
```

### Задача 1:

1. Задать массив целых чисел длиной 10 эл.
2. Отсортировать числа в массиве
3. Вывести на экран результат.
4. Сделать возможность наполнения массива с клавиатуры\*.

### Задача 2:

1. Задать массив целых чисел длиной N эл.
2. Поменять 1-й и последний элемент местами.
3. Вывести массив на экран.

### Задача 3:

1. Ввести с клавиатуры массив из 10 чисел.
2. Разбить его на 2 массива равной длины.
3. Отсортировать каждую из половинок и вывести их содержимое на экран.

### Задача 4:

1. Среднее арифметическое элементов массива.

### Задачи на 5-ку 😊

- Написать код для зеркального переворота массива (1,2,3,4) -> (4,3,2,1).
- Написать программу, которая позволит задать длину массива с клавиатуры, наполнить его элементами, а затем выводить нужный элемент по его индексу на консоль.



# Занятие 4

# Ветвление и циклы

- \*if/else

- \*switch

- \*while

- \*do/while

- \*for

# Ветвление и циклы (if)

```
if (условие) {
```

```
...
```

```
}
```

```
if (!условие) {
```

```
...
```

```
}
```

```
if (условие) {
```

```
...
```

```
} else {
```

```
...
```

```
}
```

# Условия перехода

Операторы:

- \* && - и
- \* || - или
- \* ! – отрицание
- \* != - не равно
- \* == - равно
- \* <, >, <=, >= - больше/меньше

```
if ( (! a) && ((b > 0) || (c == 3) || (d != 0))) {  
...  
}
```

# Условия перехода

Оператор ?:

```
short x = (a > 0) ? 1 : 22;
```

```
int y = ((a != 0) && (b > 5) && (c <= 77)) ? 0 : 1;
```

# Ветвление и циклы (if)

```
int a = 1;  
boolean b = a > 0;  
  
if (b) // if (b == true)  
    System.out.println("YES");  
else  
    System.out.println("NO");
```

# Ветвление и циклы (if)

```
int a = 5;
```

```
if (a == 1)
```

```
    System.out.println("1");
```

```
else if (a == 2)
```

```
    System.out.println("2");
```

```
else
```

```
    System.out.println("> 2");
```

# Ветвление и циклы (if)

```
int x = 2;
```

```
switch (x) {
```

```
case 1:
```

```
    System.out.println("first");
```

```
    break;
```

```
case 2:
```

```
    System.out.println("second");
```

```
    break;
```

```
case 3:
```

```
    System.out.println("third");
```

```
    break;
```

```
default:
```

```
    System.out.println("otherwise");
```

```
    break;
```

```
}
```

# Ветвление и циклы (if)

```
int x = 2;
```

```
if (x == 1)
```

```
    System.out.println("first");
```

```
else if (x == 2)
```

```
    System.out.println("second");
```

```
else if (x == 3)
```

```
    System.out.println("third");
```

```
else
```

```
    System.out.println("otherwise");
```

# Ветвление и циклы (if)

```
String s = "hello"; // Java 7
```

```
switch (s) {  
  case "1234":  
    System.out.println("first");  
    break;  
  case "hello":  
    System.out.println("second");  
    break;  
  case "gdfggf":  
    System.out.println("third");  
    break;  
  default:  
    System.out.println("otherwise");  
    break;  
}
```

# Ветвление и циклы (while)

```
while (условие) {
```

```
...
```

```
}
```

```
int n = 5;
```

```
while (n-- > 0)
```

```
    System.out.println("N = " + n);
```

# Ветвление и циклы (do/while)

```
do {  
...  
} while (условие);
```

```
int n = 5;  
do {  
    System.out.println("N = " + n--);  
} while (n != 0);
```

# Ветвление и циклы (for)

```
for (инициализация; условие; итерация) {  
    ...  
}
```

```
int[] a = new int[100];  
for (int i = 0; i < a.length; i++)  
    a[i] = i;
```

# Ветвление и циклы (for)

```
int[] a = new int[100];  
int[] b = new int[100];
```

```
for (int i = 0; i < a.length; i++) {  
    a[i] = i;  
    b[i] = i * 2;  
}
```

# Ветвление и циклы (for)

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

```
for (int i = 0, j = a.length - 1; i < a.length && j >= 0; i++, j--) {  
    a[i] = i;  
    b[j] = j;  
}
```

# Ветвление и циклы (for)

```
import java.util.Random;
...
int[] a = new int[100];
Random r = new Random();

for (int i = 0; i < a.length; i++)
    a[i] = r.nextInt();

int sum = 0; // long
for (int i = 0; i < a.length; i++) {
    if (a[i] > 5)
        sum += a[i];
}

System.out.println("Сумма = " + sum);
```

# Ветвление и циклы

```
int a = 0;
while (true) {
    a++;
    if (a == 5)
        break; //!!!
}
```

```
for (int i = 0; i < 100; i++) {
    if (i == 10)
        continue; // !!!

    System.out.println(i);
}
```

# Ветвление и циклы

for-each:

```
long[] a = {1,2,3,4,6};  
for (long x : a)  
    System.out.println(x);
```

```
String[] b = {"111", "222", "333"};  
for (String s : b)  
    System.out.println(s);
```

# Контрольный вопрос



```
int a = 5, b = 7, c = 13;
if ((a <= b) && ((b > 0) && ((c < 10) || (c != 12))))
    System.out.println("YES");
else
    System.out.println("NO");
if ((a > c) || (b > c) || (c - a >= 1 + b))
    System.out.println("YES");
else
    System.out.println("NO");
```

# Задачи

```
int[] x = new int[28];  
  
for (int i = 0; i < x.length; i++) {  
    x[i] = i * 635;  
}
```

Вывести на экран 0-й, 14-й и 27-й элементы массива.

# Задачи

1. Вывести на экран числа от 10 до 20 с помощью всех известных циклов.
2. Вывести на экран все числа от 1 до 100, которые делятся на 3 без остатка.
3. Заполнить массив числами от 100 до 0.
4. Дано массив из 10 целых чисел. Вывести на экран сумму всех его элементов кроме первого и последнего.
5. \* Написать игру «Крестики нолики» используя двумерные массивы.

# Задания

6. Найти в массиве чисел элементы с наибольшим и наименьшим значениями.
7. Найти в массиве число, которое повторяется наибольшее количество раз. Не использовать коллекции.
- 8\*. С помощью консоли пользователь вводит математическое выражение типа “ $1+33-4*7$ ”. Написать программу для подсчета его значения. Приоритет операций не учитывается.

# Задачи

9. Написать метод для зеркального переворота элементов в массиве ( $[1, 2, 3, 4] \rightarrow [4, 3, 2, 1]$ ).
10. Написать метод, который заполнит массив произвольного размера числами по возрастанию, начиная с центра массива, например,  $[5, 4, 3, 2, 1, 0, 1, 2, 3, 4, 5]$ .

# Задачи

```
for (int i = 3; i < 100; i += 3)  
    System.out.println(i);
```

# Задачи

```
private static void task1(int[] a) {  
    final int len = a.length;  
    int t;  
  
    for (int i = 0; i < len / 2; i++) {  
        t = a[i];  
        a[i] = a[len - i - 1];  
        a[len - i - 1] = t;  
    }  
}
```

# Задачи

```
private static void task2(int[] a) {  
    final int len = a.length;  
    int n = (len % 2 == 0) ? 1 : 0;  
  
    for (int i = len / 2; i < len; i++)  
        a[i] = a[a.length - i - 1] = n++;  
    // a[i] = n; a[a.length - i - 1] = n++;  
}
```

# Задания

```
private static void task6(int[] a) {  
    int min, max;  
  
    min = max = a[0];  
  
    for (int i = 1; i < a.length; i++) {  
        if (a[i] > max)  
            max = a[i];  
        if (a[i] < min)  
            min = a[i];  
    }  
  
    System.out.println(min + " / " + max);  
}
```

[6, 4, 7, 1, 2]

# Задания

```
private static int task7(int[] a) {
    int res = 0, cc, cp = 0;

    for (int i = 0; i < a.length; i++) {
        cc = 0;

        for (int j = 0; j < a.length; j++) {
            if (a[j] == a[i])
                cc++;
        }

        if (cc > cp) {
            cp = cc;
            res = a[i];
        }
    } // for

    System.out.println(res);
    return res;
}
```

[1, 2, 2, 2, 3, 3]

# ОСНОВЫ ООП

Задача про калькулятор для  
самостоятельного разбора:

<http://bit.ly/1fWMEiB>

(Calcpp.zip)



# Занятие 4

# Отладка программ

**Отладка** — этап разработки программы, на котором обнаруживают, локализуют и устраняют ошибки.

Чтобы понять, где возникла ошибка, придется :

- \* узнавать текущие значения переменных;
- \* выяснять, по какому пути выполнялась программа.

# Технологии отладки

**Использование отладчиков** — программ, которые включают в себя пользовательский интерфейс для пошагового выполнения программы: оператор за оператором, функция за функцией, с остановками на некоторых строках исходного кода или при достижении определённого условия.

**Вывод текущего состояния программы** с помощью расположенных в критических точках программы операторов вывода (trace код) — на экран или в файл. Вывод отладочных сведений в файл называется журналированием.

# Отладка программ

Пройтись отладчиком по следующим примерам кода.

# Отладка программ

```
int[] x = new int[14];  
for (int i = 0; i < x.length; i++) {  
    if ((i * 444) % 17 > 0)  
        x[i] = i;  
    else  
        x[i] = -1;  
}
```

Проследить за выполнением программы в debugger.

# Assert

Добавить опцию `-ea` в командную строку при запуске программы.

```
int x = 10;  
assert x == 10;  
assert x == 11; // java.lang.AssertionError
```

```
assert x == 12 : x; // java.lang.AssertionError: 10
```

# Методы

*Метод — это именованный обособленный блок кода.*

# Методы

Методы используются в программировании, чтобы уменьшить его сложность:

- \* Вместо того, чтобы писать непрерывную последовательность команд, в которой вскоре перестаешь ориентироваться, программу разбивают на подпрограммы, каждая из которых решает небольшую законченную задачу, а потом большая программа составляется из этих подпрограмм (декомпозицией).
- \* Уменьшается общее количество кода, потому что, как правило, один метод используется в программе несколько раз.
- \* Написанный однажды и всесторонне проверенный метод, может быть включен в библиотеку и использоваться в других программах.

# Методы

## Процедуры:

```
public class MyClass {  
  
    static void doJob(String s) {  
        for (int i = 0; i < 10; i++)  
            System.out.println(s + ":" + i);  
    }  
  
    public static void main(String[] args) {  
        doJob("hello1");  
        doJob("hello2");  
    }  
}
```

# Методы

- \* Примитивные типы передаются по значению (int, long etc.)
- \* Ссылки на объекты передаются по значению (объекты передаются по ссылке).

```
import java.util.*;
```

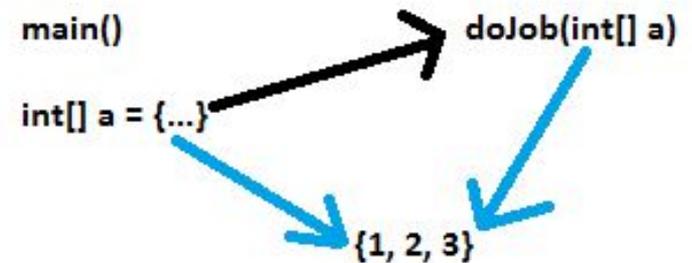
```
public class MyClass {
```

```
    static void doJob(int x) { // копирование аргумента  
        x = x + 60;  
    }
```

```
    static void doJob(int[] a) {  
        a[0] = 777;  
    }
```

```
    public static void main(String[] args) {  
        int x = 1;  
        System.out.println("x1 =" + x); // 1  
        doJob(x);  
        System.out.println("x2 =" + x); // 1
```

```
        int[] a = new int[] {1, 2, 3};  
        doJob(a);  
        System.out.println(Arrays.toString(a)); // [777, 2, 3]  
    }  
}
```



# Методы

Функции:

```
public class MyClass {  
  
    public static void main(String[] args) {  
        double[] x = {1.2, 1.3, 1.5, 6, 8};  
        System.out.println("Result: " + calc(x));  
    }  
  
    static double calc(double[] n) { // среднее арифметическое  
        double sum = 0;  
        for (int i = 0; i < n.length; i++)  
            sum += n[i];  
  
        return sum / n.length;  
    }  
}
```

## Функции:

```
public class MyClass {  
    static int add(int x, int y) {  
        return x + y;  
    }  
}
```

```
static int add(int ... n) { // тоже самое, что ... int add(int[] n)  
    int sum = 0;  
    for (int x : n)  
        sum += x;  
  
    return sum;  
}
```

```
public static void main(String[] args) {  
    int r = add(1, 5);  
    System.out.println("Result: " + r);  
    r = add(1, 2, 3, 4);  
    System.out.println("Result: " + r);  
  
    System.out.println("Result: " + add(1, 5) + ", " + add(1, 2, 3, 4));  
}
```

```
public class MyClass {  
  
    static void method1() {  
        System.out.println("method1");  
        method2(100);  
    }  
  
    static void method2(int x) {  
        System.out.println("method2 " + x);  
        method3(x);  
    }  
  
    static void method3(int x) {  
        System.out.println("method3 " + x);  
        // method1(); // бесконечный цикл!  
    }  
  
    public static void main(String[] args) {  
        method1();  
    }  
}
```

# Методы

Передача массива в метод:

```
static void method(int[] param) {  
    ...  
}
```

```
...  
int[] a = {1,2,3,4};  
method(a);  
// или  
method(new int[] {1, 2, 3, 4} );
```

# Задачи

- \* Написать функцию, которая принимает в качестве аргументов одну строку  $X$ , целое число  $Y$  и число с плавающей точкой  $Z$  и возвращает как результат строку в виде
$$S = x + y + z.$$
- \* Написать функцию, которая принимает массив чисел в качестве аргумента, увеличивает его первые 3 элемента на 1 и возвращает их сумму как результат. После изменения массив и сумму надо вывести на экран.
- \* Написать ф-ю, которая принимает на вход массив чисел и возвращает его длину в байтах как результат.
- \* Написать ф-ю для объединения 2-х массивов в один. Вывести результат на консоль.

# Задачи

```
public class MyClass {  
    private static int sizeof(byte[] a) {  
        return a.length;  
    }  
  
    private static int sizeof(int[] a) {  
        return a.length * 4;  
    }  
  
    public static void main(String[] args) {  
        byte[] a1 = {1, 2, 3};  
        int[] a2 = {1, 2, 3, 4};  
  
        System.out.println(sizeof(a1) + ", " + sizeof(a2));  
    }  
}
```



# **Занятие 6**

# Дата и время

```
import java.util.Date;

public class MyClass {
    public static void main(String[] args) {
        @SuppressWarnings("deprecation")
        Date dt1 = new Date(2014-1900, 7, 25);
        System.out.println(dt1);

        long ctm = System.currentTimeMillis(); // текущая дата и время
        Date dt2 = new Date(ctm);
        System.out.println(dt2);

        System.out.println(dt1.equals(dt2));
        System.out.println(dt1.compareTo(dt2));
        System.out.println(dt2.compareTo(dt1));
    }
}
```

```
Mon Aug 25 00:00:00 EEST 2014
Wed Aug 27 13:10:10 EEST 2014
false
-1
1
```

# Скорость работы кода

```
public class MyClass {  
    public static void main(String[] args) {  
        long start = System.currentTimeMillis();  
  
        for (int i = 0; i < Short.MAX_VALUE; i++) {  
            System.out.println(i);  
        }  
  
        long res = System.currentTimeMillis() - start;  
        System.out.println("Result = " + res);  
    }  
}
```

# Дата и время

```
import java.text.SimpleDateFormat;
import java.util.Date;

public class MyClass {
    public static void main(String[] args) {
        Date dt = new Date(System.currentTimeMillis());
        System.out.println(dt);

        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss:SS");
        String res = sdf.format(dt);
        System.out.println(res);

        sdf = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
        res = sdf.format(dt);
        System.out.println(res);

        sdf = new SimpleDateFormat("dd.MM.yyyy DDD"); // day in year
        res = sdf.format(dt);
        System.out.println(res);
    }
}
```

---

```
Wed Aug 27 13:15:05 EEST 2014
2014-08-27 13:15:05:143
27/08/2014 13:15:05
27.08.2014 239
```

# Дата и время

```
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;

public class MyClass {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");

        System.out.println("Enter the date: ");
        String dtStr = sc.nextLine();

        try {
            Date dt = sdf.parse(dtStr);
            System.out.println(dt);
        } catch (ParseException e) {
            System.out.println("Wrong date!");
        }

        sc.close();
    }
}
```

```
Enter the date:
02/01/2014 13:01:02
Thu Jan 02 13:01:02 EET 2014
```

# Дата и время

```
import java.util.Calendar;
import java.util.Date;

public class MyClass {
    public static void main(String[] args) {
        Calendar c = Calendar.getInstance();
        System.out.println("Year: " + c.get(Calendar.YEAR));
        System.out.println("Month: " + (c.get(Calendar.MONTH) + 1)); // 0..11
        System.out.println("Day: " + c.get(Calendar.DAY_OF_MONTH));

        c.add(Calendar.YEAR, 1);
        c.add(Calendar.MONTH, 2);
        c.add(Calendar.DAY_OF_MONTH, -2);

        Date dt = c.getTime();
        System.out.println(dt);
    }
}
```

```
Year: 2014
Month: 11
Day: 4
Sat Jan 02 09:18:32 EET 2016
```

# StringBuilder

```
String str1 = "1", str2 = "2", str3 = "3", str4 = "4";  
String res = str1 + str2 + str3 + str4; // не эффективно!
```

// тоже самое, что

```
String t1 = str1 + str2; // GC
```

```
String t2 = t1 + str3; // GC
```

```
String res = t2 + str4;
```

# StringBuilder

```
public class MyClass {  
    public static void main(String[] args) {  
        // #1  
        String s = "";  
        for (int i = 0; i < 10; i++)  
            s += "x";  
        System.out.println(s);  
  
        // #2  
        StringBuilder sb = new StringBuilder();  
        for (int i = 0; i < 10; i++)  
            sb.append('x');  
        System.out.println(sb.toString());  
  
        // #3  
        sb = new StringBuilder("start-");  
        for (int i = 0; i < 10; i++)  
            sb.append('x');  
        System.out.println(sb.toString());  
  
        // #4  
        sb = new StringBuilder(10);  
        for (int i = 0; i < 10; i++)  
            sb.append('x');  
        System.out.println(sb.toString());  
  
        // #5  
        sb = new StringBuilder("0123456789");  
        sb.deleteCharAt(0);  
        System.out.println(sb.toString());  
        sb.insert(0, 'A');  
        System.out.println(sb.toString());  
        sb.reverse();  
        System.out.println(sb.toString());  
        sb.append(777);  
        System.out.println(sb.toString());  
    }  
}
```

```
XXXXXXXXXXXXX  
XXXXXXXXXXXXX  
start-XXXXXXXXXXXXX  
XXXXXXXXXXXXX  
123456789  
A123456789  
987654321A  
987654321A777
```

# Math

```
public class MyClass {  
    public static void main(String[] args) {  
        System.out.println("Pi = " + Math.PI);  
  
        System.out.println("Cos = " + Math.cos(1));  
        System.out.println("Sin = " + Math.sin(1));  
  
        System.out.println("Lg = " + Math.log10(0.5));  
  
        System.out.println("Max = " + Math.max(1, 2));  
        System.out.println("Min = " + Math.min(1, 2));  
    }  
}
```

```
Pi = 3.141592653589793  
Cos = 0.5403023058681398  
Sin = 0.8414709848078965  
Lg = -0.3010299956639812  
Max = 2  
Min = 1
```

# BigInteger

```
import java.math.BigInteger;
import java.util.Arrays;

public class MyClass {
    public static void main(String[] args) {
        BigInteger i1 = new BigInteger("1234567890123456789012322334534543543");
        BigInteger i2 = new BigInteger("8324784758934274328974385873425979823495");

        System.out.println(i1.add(i2));
        System.out.println(i1.subtract(i2));
        System.out.println(i2.multiply(i1));

        byte[] b = i1.toByteArray();
        System.out.println(Arrays.toString(b));
    }
}
```

```
8326019326824397785763398195760514367038
-8323550191044150872185373551091445279952
10277511955569396902559818785229283877137302637858631886247639240427531942785
[0, -19, -60, -27, 126, 102, -98, -75, 45, 58, 121, 51, 51, -16, 40, -73]
```

# Format

```
public class MyClass {  
    public static void main(String[] args) {  
        String s1 = "string";  
        int n = 20;  
  
        String res = String.format("String: %s, number: %d", s1, n);  
        System.out.println(res);  
    }  
}
```

String: string, number: 20

# Format

Conversion	Argument Category	Description
'b', 'B'	general	If the argument <i>arg</i> is <code>null</code> , then the result is <code>"false"</code> . If <i>arg</i> is a <code>boolean</code> or <code>Boolean</code> , then the result is the string returned by <code>String.valueOf(arg)</code> . Otherwise, the result is <code>"true"</code> .
'h', 'H'	general	If the argument <i>arg</i> is <code>null</code> , then the result is <code>"null"</code> . Otherwise, the result is obtained by invoking <code>Integer.toHexString(arg.hashCode())</code> .
's', 'S'	general	If the argument <i>arg</i> is <code>null</code> , then the result is <code>"null"</code> . If <i>arg</i> implements <code>Formattable</code> , then <code>arg.formatTo</code> is invoked. Otherwise, the result is obtained by invoking <code>arg.toString()</code> .
'c', 'C'	character	The result is a Unicode character
'd'	integral	The result is formatted as a decimal integer
'o'	integral	The result is formatted as an octal integer
'x', 'X'	integral	The result is formatted as a hexadecimal integer
'e', 'E'	floating point	The result is formatted as a decimal number in computerized scientific notation
'f'	floating point	The result is formatted as a decimal number
'g', 'G'	floating point	The result is formatted using computerized scientific notation or decimal format, depending on the precision and the value after rounding.
'a', 'A'	floating point	The result is formatted as a hexadecimal floating-point number with a significand and an exponent
't', 'T'	date/time	Prefix for date and time conversion characters. See <a href="#">Date/Time Conversions</a> .
'%'	percent	The result is a literal <code>'%'</code> ( <code>'\u0025'</code> )
'n'	line separator	The result is the platform-specific line separator





# Сортировка (пузырьковая)

```
import java.util.Arrays;

public class MyClass {
    static void swap(int[] a, int x, int y) {
        int t = a[x];
        a[x] = a[y];
        a[y] = t;
    }

    static void bubble_sort(int[] a) {
        for (int i = 0; i < a.length - 1; i++) {
            for (int j = 0; j < a.length - i - 1; j++) {
                if (a[j] > a[j + 1]) {
                    swap(a, j, j + 1);
                    System.out.println(Arrays.toString(a)); // debug
                }
            }
        }
    }

    public static void main(String[] args) {
        int[] a = {5, 7, 3, 5, 9, 1};
        bubble_sort(a);
        System.out.println(Arrays.toString(a));
    }
}
```

```
[5, 3, 7, 5, 9, 1]
[5, 3, 5, 7, 9, 1]
[5, 3, 5, 7, 1, 9]
[3, 5, 5, 7, 1, 9]
[3, 5, 5, 1, 7, 9]
[3, 5, 1, 5, 7, 9]
[3, 1, 5, 5, 7, 9]
[1, 3, 5, 5, 7, 9]
[1, 3, 5, 5, 7, 9]
```

# Задачи

1. Ввести с консоли дату. Сравнить ее с текущей датой в системе. Вывести отличающиеся части (год, месяц) на экран.
2. Написать свой вариант ф-и `Arrays.equals` для `short[]`.
3. Написать свой вариант ф-и `Arrays.toString()` для `int[]`.
4. Написать код, который позволит менять и читать значения произвольных битов в массиве `int`-ов.
5. Ввести с консоли число в бинарном формате. Перевести его в `int` и вывести на экран (“10” -> 2).
6. Ввести с консоли целое число. Посчитать количество единиц в его бинарном представлении (“1011” -> 3).

# Решения

```
static void task1() {  
    Calendar c1 = Calendar.getInstance();  
  
    Calendar c2 = Calendar.getInstance();  
    c2.setTime(new Date());  
  
    if ( ! c1.equals(c2) ) {  
        int m1 = c1.get(Calendar.MILLISECOND);  
        int m2 = c2.get(Calendar.MILLISECOND);  
  
        if (m1 != m2)  
            System.out.println(m1 + " - " + m2);  
  
        //...  
    }  
}
```

# Решения

```
static boolean equals(short[] a1, short[] a2) {  
    if (a1.length != a2.length)  
        return false;  
  
    for (int i = 0; i < a1.length; i++) {  
        if (a1[i] != a2[i])  
            return false;  
    }  
  
    return true;  
}
```

```
public static void main(String[] args) {  
    short[] a1 = {1, 2, 3, 4};  
    short[] a2 = {1, 2, 3, 4};  
    short[] a3 = {1, 7, 3, 1};  
    short[] a4 = {1, 7, 1};  
  
    System.out.println(equals(a1, a2));  
    System.out.println(equals(a2, a3));  
    System.out.println(equals(a1, a4));  
}
```

# Решения

```
static String toString(int[] a) {
    StringBuilder sb = new StringBuilder("[");

    for (int i = 0; i < a.length; i++) {
        if (i < a.length - 1)
            sb.append(a[i]).append(", ");
        else
            sb.append(a[i]);
    }

    return sb.append("]").toString();
}

public static void main(String[] args) {
    int[] a1 = {1, 2, 3, 4};

    System.out.println(toString(a1));
}
```

```
static int setBit(int x, int pos) {
    return x | (1 << pos);
}

static boolean isBitSet(int x, int pos) {
    return (x & (1 << pos)) != 0;
}

static void setBit(int[] a, int pos) {
    int intN = a.length - (pos / 32) - 1;
    int bitN = pos % 32;
    a[intN] = setBit(a[intN], bitN);
}

static boolean isBitSet(int[] a, int pos) {
    int intN = a.length - (pos / 32) - 1;
    int bitN = pos % 32;
    return isBitSet(a[intN], bitN);
}

public static void main(String[] args) {
    int[] a = {0, 0};

    setBit(a, 0);
    setBit(a, 1);
    setBit(a, 32);
    setBit(a, 63);

    System.out.println(isBitSet(a, 0));
    System.out.println(isBitSet(a, 1));
    System.out.println(isBitSet(a, 32));
    System.out.println(isBitSet(a, 63));
    System.out.println(isBitSet(a, 55)); // false!

    System.out.println(toString(a));
}
```

```
public static void main(String[] args) {  
    int n = 117, count = 0; // 1110101  
  
    while (n != 0) {  
        if (n % 2 > 0)  
            count++;  
  
        n = n / 2;  
    }  
  
    System.out.println(count);  
}
```



# Занятие 7

# Тестирование

1.

```
public static int getRes(int x, int y) {  
    if ((x == 7) || (y > x))  
        return x * y;  
    else if (y - x > 3*3)  
        return y - x;  
    else  
        return x / y;  
}
```

```
public static void doJob() {  
    final int r = getRes(7, 1);  
    System.out.println("Result = " + r);  
    // Что будет выведено на экран?  
}
```

# Тестирование

2. Что будет выведено на экран?

```
public final void doJob() {  
    double x = 3;  
    System.out.println("Result=" + "x");  
    System.out.println("Result=" + x);  
}
```

3. Что будет выведено на экран?

```
public final void doJob() {  
    long x = 1;  
    double y = 2;  
    System.out.println("Result=" + (x + y));  
}
```

# Тестирование

4.

```
public void A() {  
    int[] a = new int[100];  
    a[0] = 2;  
    for (int j = 1; j < a.length; j++)  
        a[j] = a[j - 1] + 1;  
}
```

Первые 6 чисел массива - ?

# Тестирование

5.

```
public static void main(String[] args) {  
    int[] a = new int[50];  
    Arrays.fill(a, 3);  
    for (int j = 1; j < a.length; j++)  
        a[j] = a[j - 1] + 0x10;  
}
```

Первые 3 числа массива - ?

# Тестирование

6.

```
public static void main(String[] a) {  
    int n = 2;  
    while (n++ < 10)  
        n += 2;  
    System.out.println(n); // какое будет значение n?  
}
```

# Тестирование

7.

```
public static int calc(int x) {  
    int xx = x;  
    do {  
        xx *= 55;  
    } while (xx < 6);  
  
    return x;  
}
```

```
public static void main(String[] args) {  
    int r = calc(3);  
    System.out.println(r); // что будет выведено на экран?  
}
```

# Тестирование

8.

```
public static void test(int[] x) {  
    x[0] = x[1] = x[2] = 7;  
    x[3]++;  
}
```

```
public static void main(String[] args) {  
    final int[] x = {1,2,3,4,5,6};  
    test(x); test(x); test(x);  
    System.out.println(Arrays.toString(x));  
    // что будет выведено на экран?  
}
```

# Тестирование

9. Найти ошибку:

```
public void aaabbb() {  
    int x;  
    for (x = 7; x < 66; ++x);  
        System.out.println(x);  
}
```

10. Найти ошибки:

```
public int xxxx(int x; int y) {  
    int x = x + y;  
    return x;  
}
```

# Тестирование

11.

```
public static void main(String[] args) {  
    int y = 8; int z = 7;  
    if ((y == z) || ((z - y == 1) && (z / y > 0)))  
        System.out.println("YES");  
    else  
        System.out.println("NO"); // на экране - ?  
}
```

12.

```
public static void main(String[] args) {  
    String s = "abcdefghijk";  
    System.out.println(s.substring(4) + s.substring(0, 2)); // на экране -  
    ?  
}
```

# Тестирование

13.

```
byte[] a = {1,2,3,4,5,6,7,8,9,10,11};
```

```
byte[] b = new byte[a.length/2];
```

```
System.arraycopy(a, 5, b, 1, 2);
```

```
System.out.println(Arrays.toString(b)); // На экране?
```

# Тестирование

14.

```
int[] a = new int[] {1,1,1};
```

```
a[a[a[0]++]] = 7;
```

```
a[a[0]--] = 8;
```

```
a[++a[0]] = 9;
```

```
System.out.println(Arrays.toString(a)); // ???
```

# Тестирование

15.

```
StringBuilder sb = new StringBuilder(10);  
sb.append('A').append('b').append("777");  
sb.deleteCharAt(1);  
System.out.println(sb.toString() + 1); // ???
```

# Тестирование

16. Что будет выведено на экран? (4 'ktv/)

```
for (int i = 0, j = 1; i < 10; i++, j++)  
    System.out.print(i + j);
```

17.

```
String s = "aaaa";  
System.out.println(s.replace("b", "a"));
```

# Тестирование

18. Вывод:

```
int x = 01; int y = 02;  
System.out.println((x << 1) + (y >> 1));
```

19. r = ?

```
int i = 1;  
String r = "";  
String s = "abcd!ert";  
while (s.charAt(i) != '!') r+=s.charAt(i++);  
System.out.println(r);
```

# Тестирование

20. На консоли?

```
String s = "one two three";  
String[] a = s.split(" ");  
System.out.println(a[1]); // ???
```

21. На консоли?

```
char[] a = "abcdefghijk".toCharArray();  
for (char c : a)  
    System.out.print((c > 'e') ? c : 'a');
```

# Тестирование

22. На консоли?

```
int u = 13;
```

```
u += 2; u %= 4;
```

```
u >>= 1; u = ~u;
```

```
System.out.println(u);
```

# Тестирование

23.

```
long n = 3;  
do {  
    n = (n * n) % 135;  
} while (n < 27);  
System.out.println(n);
```

# Тестирование

24.

```
for (int i = 0; i < 4; i++) {  
    if ((i & 1) > 0)  
        System.out.print("+");  
    else  
        System.out.print("-");  
}
```

# Тестирование

25.

```
String s = " 1234%d89%do ";  
s = String.format(s, 0, 1).replace('1', 'Z').trim().toLowerCase();  
System.out.println(s);
```

# Тестирование

26.

```
byte[] b = {3, 3, 3, 3, 3, 3};  
for (int k = 1; k < b.length - 1; k++)  
    b[k] = (byte)(b[k - 1] + b[k + 1]);  
  
System.out.println(Arrays.toString(b));
```

# Тестирование

27.

```
public class Main {
    static int do1(int a) {
        int b = a;
        a = b;
        b = a;
        return b;
    }

    static int do2(int a) {
        int b = a;
        b *= b;
        return do1(b);
    }

    public static void main(String[] args) {
        System.out.println(do2(5)); // ??????????????????
    }
}
```

# Тестирование

28.

```
int[] a = {11, 77};
```

```
int x = 7;
```

```
int y = x + 5 > 1 ? (x + 7 != 14 ? x : 7) : 7;
```

```
System.out.println(a[y % 2]); // ???
```

# Тестирование

29.

```
double d = 0.5;  
d = Math.pow(d, 2);  
String res = d + d > 0.45 ? "yes" : "no";  
System.out.println(res.length()); // ???
```

# Тестирование

30.

```
int c = 1, d = -6, e = 0;
```

```
switch (c) {  
    case 5:  
        c = 8;  
        break;  
    case 7:  
        c = 9;  
        break;  
    default:  
        c = 11;  
}
```

```
if ((c + d == e) || (e + c == d) || ((e = 1) > 0))  
    System.out.println("yo!");  
else  
    System.out.println("no!");
```

# Задачи

1. Дано текст из 3-х слов “word1 word2 word3”. Поменять 1-е и 3-е слова местами и вывести на экран результат.
2. Дано текст и определенное слово. Посчитать сколько раз заданное слово встречается в тексте. 0 1 и много
3. Дано 3 массива чисел. С помощью 1-2-х циклов найти сумму элементов во всех массивах.
4. Дано текст. Убрать лишние пробелы, расставить знаки препинания. Пример: «Я длинное предложение Я второе предложение.» -> «Я длинное предложение. Я второе предложение.»

# Решения

```
static void task1() {  
    String input = "word1 word2 word3";  
    String[] parts = input.split(" ");  
    String res = parts[2] + parts[1] + parts[0];  
  
    System.out.println(res);  
}
```

# Решения

```
static void task2() {  
    String word = "hello";  
    String text = "hello just hello another hello and hello d";  
    int pos = 0, count = 0;  
  
    do {  
        pos = text.indexOf(word, pos);  
  
        if (pos >= 0) {  
            count++;  
            pos += word.length();  
        }  
    } while (pos >= 0);  
  
    System.out.println(count);  
}
```

# Решения

```
static void task3() {  
    int[] a1 = {1, 2, 3}, a2 = {4, 5}, a3 = {6, 7, 8};  
  
    int[][] all = {a1, a2, a3};  
    int res = 0;  
  
    for (int i = 0; i < all.length; i++) {  
        for (int j = 0; j < all[i].length; j++)  
            res += all[i][j];  
    }  
  
    System.out.println(res);  
}
```

# Решения

```
public class MyClass {
    public static void main(String[] args) {
        final String text = "I am very bad text Please fix me. Yes!";

        StringBuilder sb = new StringBuilder();
        String[] words = text.split(" ");

        for (int i = 0; i < words.length; i++) {
            if (words[i].length() > 0) {
                sb.append(words[i]);

                if ((i + 1 < words.length) &&
                    (words[i + 1].length() > 0) &&
                    Character.isUpperCase(words[i + 1].charAt(0)) &&
                    (! words[i].endsWith(".")))
                {
                    sb.append(". ");
                }
                else
                    sb.append(' ');
            }
        }

        System.out.println(sb.toString());
    }
}
```



# **Занятие 8**

## **Практика**

# Задачи

1. Вывести на экран все буквы английского алфавита.
2. Проверить баланс круглых скобок в выражении. Пример: “ $((x - 8) + a(y - 1))$ ”.
3. Дан словарь в виде «англ. слово=русское слово, англ. слово=русское слово,...». Написать программу переводчик на основе словаря.
4. Вывести на экран сумму цифр заданного числа.
5. Вычислить  $x^n$ .
6. Вывести на экран все простые числа, которые меньше заданного  $N$ .
7. Написать программу для решения квадратных уравнений. \*

# 1

```
static void task1() {  
    for (char c = 'A'; c <= 'Z'; c++)  
        System.out.print(c);  
    System.out.println("");  
}
```

# 2

```
static boolean task2(String s) {  
    int b = 0;  
  
    for (int i = 0; i < s.length(); i++) {  
        if (s.charAt(i) == '(')  
            b++;  
        else if (s.charAt(i) == ')')  
            b--;  
    }  
  
    return (b == 0);  
}
```

# 3

```
static String task3(String dict, String s) {
    String[] pairs = dict.split(";");
    String[] eng = new String[pairs.length];
    String[] rus = new String[pairs.length];

    for (int i = 0; i < pairs.length; i++) {
        String[] p = pairs[i].trim().split("=");
        eng[i] = p[0];
        rus[i] = p[1];
    }

    String[] words = s.split(" ");
    StringBuilder sb = new StringBuilder();

    for (String w : words) {
        for (int i = 0; i < eng.length; i++) {
            if (eng[i].equals(w))
                sb.append(rus[i]).append(' ');
        }
    }

    return sb.toString();
}
```

```
System.out.println(task3("yes=да; no=нет", "no yes"));
```

# 4

```
static int task4(int x) {  
    int count = 0;  
  
    while (x != 0) {  
        count = count + (x % 10);  
        x /= 10;  
    }  
  
    return count;  
}
```

# 5

```
static long task5(int x, int n) {  
    int res = 1;  
  
    while (n-- > 0)  
        res *= x;  
  
    return res;  
}
```

---

# 6

```
static void task6(int n) {  
    boolean flag;  
    for (int i = 1; i <= n; i++) {  
        flag = true;  
        for (int j = 2; j < i; j++) {  
            flag = flag & (i % j > 0);  
        }  
        if (flag)  
            System.out.print(i + " ");  
    }  
}
```



если остается время...

Практика

# Практика

1. Найти все уникальные (не повторяющиеся) элементы массива.
2. Построить таблицу значений ф-и на отрезке  $[-1; 0.75]$ .

$$f(x) = x \left( 1 + \frac{2}{x+7} \right)^{x/10}$$

3. Найти в предложении самое короткое и самое длинное слово.
4. Написать аналог `String.replace()`.
5. Написать аналог `String.indexOf()`.

# Практика

6. Реализовать аналог `String.substring()`.
7. Реализовать аналог `String.trim` и `trimLeft/trimRight`.
8. Реализовать аналог `Integer.parseInt`.
9. Вывести на экран таблицу умножения.
10. Дано два числа `short1` и `short2`. Сделать из них один `int` методом конкатенации с помощью побитовых операций.
11. Дано два массива одинаковой длины, упорядоченных по возрастанию. Объединить массивы так, чтобы получился один упорядоченный массив. Пример:  $\{1, 3, 10\} + \{2, 5, 8\} = \{1, 2, 3, 5, 8, 10\}$ .

# Задача 1

```
static void task_0(int[] a) {  
    int n;  
  
    for (int i = 0; i < a.length; i++) {  
        n = 0;  
  
        for (int j = 0; j < a.length; j++) {  
            if (a[i] == a[j])  
                n++;  
        }  
  
        if (n == 1)  
            System.out.println(a[i]);  
    }  
}
```



# Задача 2

```
static void task_1k() {  
    double x = -1;  
    double res;  
  
    while (x <= 0.75) {  
        res = x * Math.pow(1 + (2 / (x + 7)), x / 10);  
        System.out.println("f(" + x + ") = " + res);  
        x += 0.05;  
    }  
}
```

# Задача 3

```
static void maxString() {  
    String s = "one two three a";  
    String[] a = s.split(" ");  
    int min = 0, max = 0;  
  
    for (int i = 1; i < a.length; i++) {  
        if (a[i].length() > a[max].length())  
            max = i;  
        if (a[i].length() < a[min].length())  
            min = i;  
    }  
  
    System.out.println("min = " + a[min]);  
    System.out.println("max = " + a[max]);  
}
```

# Задача 5

```
static int indexOf(String s, String sub) {
    boolean b;

    for (int i = 0; i <= s.length() - sub.length(); i++) {
        b = true;

        for (int j = 0; j < sub.length(); j++) {
            b = b & (s.charAt(i + j) == sub.charAt(j));
            if (! b)
                break;
        }

        if (b)
            return i;
    }

    return -1;
}
```

# Задача 6

```
static String substring(String s, int start, int end) {  
    if (start < 0 || end < 0 || (start > end) || (start >= s.length()) || (end >= s.length()))  
        return null;  
  
    StringBuilder sb = new StringBuilder(end - start);  
  
    for (int i = start; i < end; i++)  
        sb.append(s.charAt(i));  
  
    return sb.toString();  
}
```

# Задача 7

```
static String trimLeft(String s) {
    StringBuilder sb = new StringBuilder(s);

    while (sb.length() > 0 && sb.charAt(0) == ' ')
        sb.deleteCharAt(0);

    return sb.toString();
}

static String trimRight(String s) {
    StringBuilder sb = new StringBuilder(s);

    while (sb.length() > 0 && sb.charAt(sb.length() - 1) == ' ')
        sb.deleteCharAt(sb.length() - 1);

    return sb.toString();
}

static String trim(String s) {
    return trimRight(trimLeft(s));
}
```

# Задача 8

```
static int parseInt(String s) throws NumberFormatException {
    int m = 1;
    int p = s.length() - 1;
    char c;
    int res = 0;

    while (p >= 0) {
        c = s.charAt(p--);

        if (c < '0' || c > '9')
            throw new NumberFormatException("wrong number format");

        res = res + ((c - 48) * m);
        m *= 10;
    }

    return res;
}
```