

Java Basics

Data Structures

Arrays

Lesson 04

OLGA SMOLYAKOVA

План конспекта

1. Что такое структуры данных	
2. Одномерные массивы примитивных типов данных	
3. Сортировки	
4. Массивы массивов	
5. Практическая работа	

- 1.1 Решим задачу. Напишите приложение, которое вводит 5 чисел с клавиатуры и выводит их сумму в виде, например:

$$(2) + (-3) + (12) + (4) + (-1) = [14]$$

```
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int a;
        int b;
        int c;
        int d;
        int e;
        int z;

        // опустим код, который контролирует корректность ввода
        System.out.print("> ");
        a = sc.nextInt();

        System.out.print("> ");
        b = sc.nextInt();

        System.out.print("> ");
        c = sc.nextInt();

        System.out.print("> ");
        d = sc.nextInt();

        System.out.print("> ");
        e = sc.nextInt();

        z = a + b + c + d + e;

        System.out.println("(" + a + ") + (" + b + ") + (" + c
            + ") + (" + d + ") + (" + e + ") = " + "[" + z + "]");
    }
}
```

Результат:

```
<terminated> Main [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe (Mar 31, 2021, 11:
> 1
> -2
> 3
> -4
> 5
(1) + (-2) + (3) + (-4) + (5) = [3]
```

- 1.2 Добавим новую возможность предыдущей задаче. Напишите приложение, которое вводит 5 чисел с клавиатуры и выводит их сумму в виде

$$(2) + (-3) + (12) + (4) + (-1) = [14]$$

или как

$$\begin{array}{ccccccc} & & & & & & 14 \\ & & & & & -1 = & \\ & & & & 4 + & & \\ & & 12 + & & & & \\ & -3 + & & & & & \\ & & 2 + & & & & \end{array}$$

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int a;
    int b;
    int c;
    int d;
    int e;
    int z;
    // опустим код, который контролирует корректность ввода
    System.out.print("> ");
    a = sc.nextInt();

    System.out.print("> ");
    b = sc.nextInt();

    System.out.print("> ");
    c = sc.nextInt();

    System.out.print("> ");
    d = sc.nextInt();

    System.out.print("> ");
    e = sc.nextInt();

    z = a + b + c + d + e;

    String message = "Введите 1, если хотите вывести результат
строкой.\n" + "Введите любое другое число, если хотите вывести
результат лесенкой. \n> ";
    System.out.print(message);
    int menu;
    while(!sc.hasNextInt()) {
        sc.nextLine();
        System.out.print(message);
    }
}
```

```

menu = sc.nextInt();

if(menu == 1) {
    System.out.println("(" + a + ") + (" + b + ") + ("
        + c + ") + (" + d + ") + (" + e + ") = "
        + "[" + z + "]");
} else {
    System.out.println("(" + a + ") +\n\t (" + b
        + ") +\n\t\t (" + c + ") +\n\t\t\t ("
        + d + ") +\n\t\t\t\t ("
        + e + ") =\n\t\t\t\t\t " + "[" + z + "]");
}
}

```

Результат:

```

> 1
> -2
> 3
> -4
> 5
Введите 1, если хотите вывести результат строкой.
Введите любое другое число, если хотите вывести результат лесенкой.
> 2
(1) +
      (-2) +
            (3) +
                  (-4) +
                        (5) =
                              [3]

```

1.3 Модифицируем задачу дальше. В приложении по-прежнему необходимо иметь возможность выводить результат в двух разных видах, но приложение теперь должно спрашивать, сколько чисел хочет ввести пользователь.

Решить эту задачу объявив n переменных не получится – появилась необходимость централизованно хранить данные. Для таких случаев подходят структуры данных, которые называются массивами.

Итак, решим задачу.

```

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n;
        int[] mas;
        // опустим код, который контролирует корректность ввода
    }
}

```

```

System.out.print("Сколько элементов хотет ввести > ");
n = sc.nextInt();

mas = new int[n];
int sum = 0;

for(int i=0; i<mas.length; i++) {
    System.out.print("mas[" + i + "]=");
    mas[i] = sc.nextInt();
    sum = sum + mas[i];
}

String message = "Введите 1, если хотите вывести результат
строкой.\n" + "Введите любое другое число, если хотите вывести
результат лесенкой. \n> ";
System.out.print(message);
int menu;
while(!sc.hasNextInt()) {
    sc.nextLine();
    System.out.print(message);
}
menu = sc.nextInt();

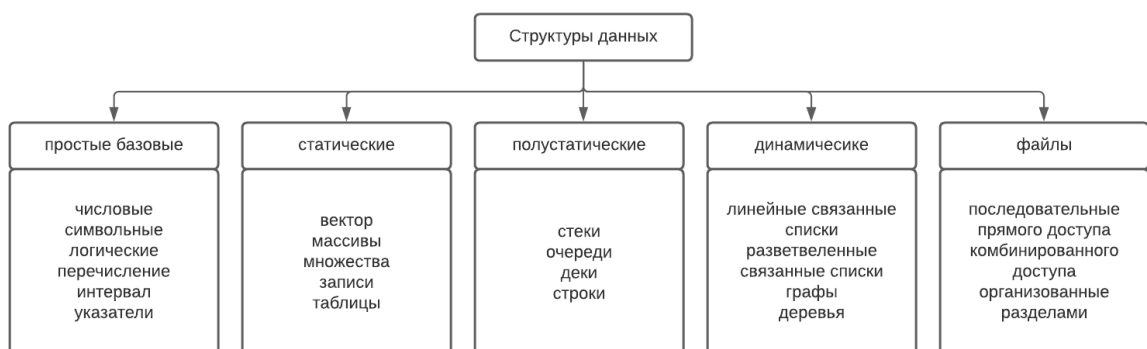
if(menu == 1) {
    for(int i = 0; i < mas.length; i++) {
        if(i == mas.length - 1) {
            System.out.print("(" + mas[i] + ") ");
        }else {
            System.out.print("(" + mas[i] + ") + ");
        }
    }
    System.out.println(" = " + "[" + sum + "]");
}else {
    String delimiter = "\n\t";
    for(int i = 0; i < mas.length; i++) {
        System.out.print("(" + mas[i] + ") + "
            + delimiter);
        delimiter = delimiter + '\t';
    }
    System.out.println(" = " + "[" + sum + "]");
}
}
}

```

1.4 Что же такое структуры данных?

Структурой данных называют множество элементов данных и связей между ними. Такое определение кажется непонятным при первом изучении. Попробуем сказать проще. Структура данных – это такой способ организации данных, который позволяет их обрабатывать единообразно и циклически в зависимости от способов, предусмотренных конкретной структурой. Группа студентов, например, тоже может быть примером структуры данных (если студентов рассматривать как данные, конечно 😊). Группе (всем студентам) можно назначить занятие, можно циклически перебирать студентов в группе (о которых будет известно только то, что они студенты, а на самом деле могут быть докторами или космонавтами) и задавать каждому вопрос и т.д.

Приведем общую классификацию структур данных.



Структуры данных и алгоритмы их обработки в отдельном курсе. В текущем курсе по Java понадобятся знания только некоторых из них.

2 Одномерные массивы примитивных типов данных

2.1 Давайте разберемся, что такое массив и как с ним работать.

В простейшем представлении можно представить одномерный массив в виде набора нумерованных ячеек.



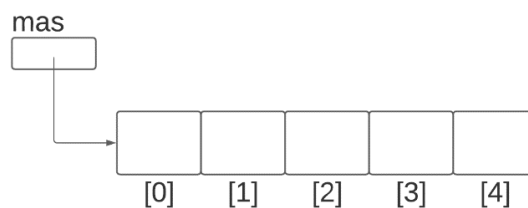
В каждой ячейке может храниться информация.



Данные в ячейках массива можно даже поменять местами.



В языке Java одномерный массив – это ссылочный тип данных. Массивы элементов базовых типов (массивы примитивных типов) состоят из значений, проиндексированных, начиная с нуля.

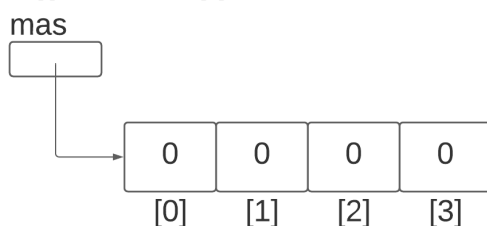


Чтобы создать массив нужно создать объект. Выделение памяти для массивов происходит также, как и для объектов, с помощью оператора **new**. Чтобы создать массив используется синтаксис:

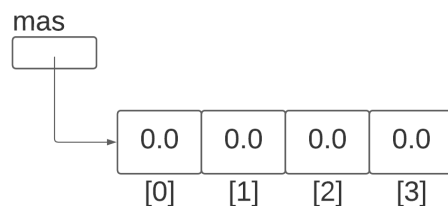
```
тип_массива[] имя_ссылки_на_массив = new тип_массива[размер];  
int[] mas = new int[10];
```

Значения элементов неинициализированных массивов, для которых выделена память, устанавливается в нуль.

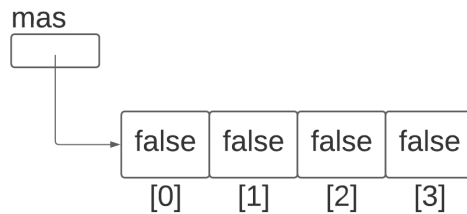
```
int[] mas = new int[4];
```



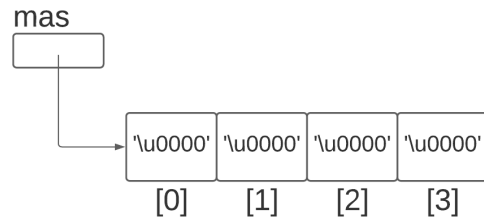
```
double[] mas = new double[4];
```




```
boolean[] mas = new boolean[4];
```



```
char[] mas = new char[4];
```



2.2 Решим задачу. Создайте массив из пяти элементов типа **int** и инициализируйте его.

```
public static void main(String[] args) {  
    int[] mas;  
    mas = new int[5];  
  
    mas[0] = 234;  
    mas[1] = 345;  
    mas[2] = 456;  
    mas[3] = 567;  
    mas[4] = 678;  
  
    System.out.println(mas[0]);  
    System.out.println(mas[1]);  
    System.out.println(mas[2]);  
    System.out.println(mas[3]);  
    System.out.println(mas[4]);  
}
```

Результат:

A screenshot of a Java IDE's console window. The title bar shows 'Problems', 'Javadoc', 'Declaration', and 'Console'. The console output shows the numbers 234, 345, 456, 567, and 678, each on a new line.

2.3 Для обращения к ячейке массива в Java используется синтаксис:

```
ссылка_на_массив[индекс_элемента]  
mas[2] = 234;
```

Индекс – это порядковый номер элемента в массиве. Индексация элементов в массиве начинается в нуля, т.е. первым элементов в массиве будет элемент к которому нужно обратиться как

```
ссылка_на_массив[0];  
mas[0] = 234;
```

Количество элементов массива (длина массива) хранится в самом массиве (т.к. массив – объект, он может хранить несколько данных одновременно, в том числе и длину массива). Узнать количество элементов массива можно следующим образом:

```
public static void main(String[] args) {
    int[] mas;
    int n = 100;

    mas = new int[n];

    int size = mas.length;
    System.out.println("Массив содержит " + size
                       + " элементов.");
}
```

Как видно из примера, объявляя массив в качестве его размера можно указать переменную целочисленного типа. Это значит, что во время работы приложения можно определить длину массива (например, спросить у пользователя), и создать массив требуемой длины.

```
public static void main(String[] args) {
    int[] mas;
    int n;

    Scanner sc = new Scanner(System.in);
    n = sc.nextInt();
    mas = new int[n];

    System.out.println("Массив содержит " + mas.length
                       + " элементов.");
}
```

После создания массива изменить его длину уже нельзя.

```
int[] mas;
int n = 20;
mas = new int[n];
```

```
mas.length = 30;
```

Для обработки элементов, которые хранятся в массиве, в Java чаще всего применяются циклы. Решим задачу по созданию и инициализации массива из 5 элементов через цикл.

2.4 Создайте массив из пяти элементов типа **double** и инициализируйте его элементы, читая значения с клавиатуры.

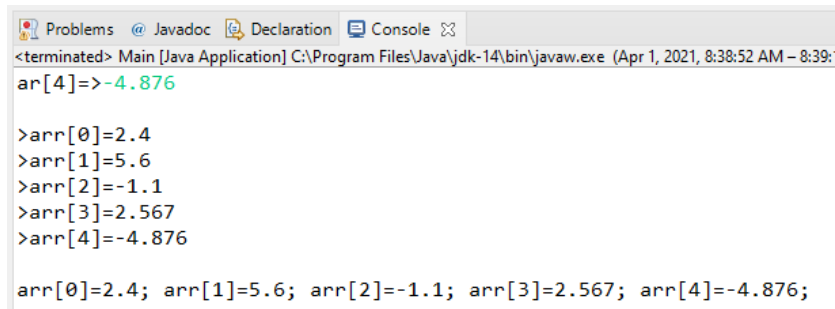
```
public static void main(String[] args) {
    double[] ar = new double[5];
    Scanner sc = new Scanner(System.in);

    for(int i=0; i<ar.length; i++){
        System.out.print("ar[" + i + "]=>");
        ar[i] = sc.nextDouble();
    }
    System.out.println();

    for (int i=0; i<ar.length; i++){
        System.out.println(">arr[" + i + "]= " + ar[i]);
    }
    System.out.println();

    for (int i=0; i<ar.length; i++){
        System.out.print("arr[" + i + "]= " + ar[i] + "; ");
    }
    System.out.println();
}
```

Результат:



```
<terminated> Main [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe (Apr 1, 2021, 8:38:52 AM - 8:39:
ar[4]=>-4.876

>arr[0]=2.4
>arr[1]=5.6
>arr[2]=-1.1
>arr[3]=2.567
>arr[4]=-4.876

arr[0]=2.4; arr[1]=5.6; arr[2]=-1.1; arr[3]=2.567; arr[4]=-4.876;
```

Постоянно вводить значения элементов массива с клавиатуры для решения задач бывает сложно (тратится много времени на ручной ввод). Перепишем решение с использованием генератора псевдослучайных чисел.

2.5 Создайте массив из пяти элементов типа **int** и инициализируйте его элементы случайными числами.

```
import java.util.Random;
public class Main {
    public static void main(String[] args) {
        int[] ar = new int[5];
        Random rand = new Random();
    }
}
```

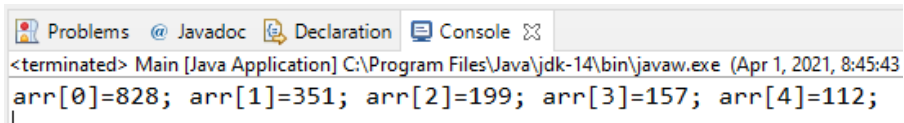
```

        for(int i=0; i<ar.length; i++){
            ar[i] = rand.nextInt(1000);
        }

        for (int i=0; i<ar.length; i++){
            System.out.print("arr[" + i + "]= " + ar[i] + "; ");
        }
        System.out.println();
    }
}

```

Результат:



```

Problems @ Javadoc Declaration Console
<terminated> Main [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe (Apr 1, 2021, 8:45:43)
arr[0]=828; arr[1]=351; arr[2]=199; arr[3]=157; arr[4]=112;

```

Для закрепления знаний по работе с массивами решим следующий ряд задач.

2.5 Решите задачу. Ввести с клавиатуры 10 чисел. Вывести их в обратном порядке.

```

public static void main(String[] args) {
    int n = 10;
    double[] mas = new double[n];

    Scanner sc = new Scanner(System.in);

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

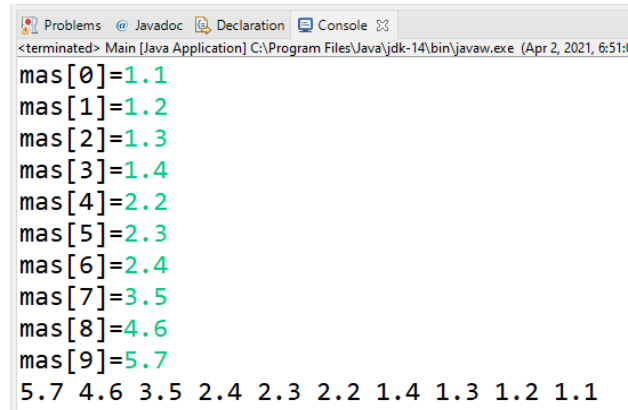
        System.out.print("mas[" + i + "]=");
        while (!sc.hasNextDouble()) {
            sc.nextLine();
            System.out.print("mas[" + i + "]=");
        }
        mas[i] = sc.nextDouble();

    }

    for(int i = mas.length - 1; i >=0; i--) {
        System.out.print(mas[i] + " ");
    }
    System.out.println();
}

```

Результат:



```
<terminated> Main [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe (Apr 2, 2021, 6:51:11 AM)
mas[0]=1.1
mas[1]=1.2
mas[2]=1.3
mas[3]=1.4
mas[4]=2.2
mas[5]=2.3
mas[6]=2.4
mas[7]=3.5
mas[8]=4.6
mas[9]=5.7
5.7 4.6 3.5 2.4 2.3 2.2 1.4 1.3 1.2 1.1
```

2.6 Решите задачу. Посчитать сумму элементов массива.

```
public class Main {

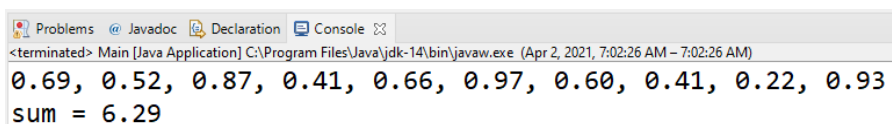
    public static void main(String[] args) {
        int n = 10;
        double[] mas = new double[n];
        double sum = 0.0;

        Random rand = new Random();
        for (int i = 0; i < mas.length; i++) {
            mas[i] = rand.nextDouble();
        }

        for(int i=0; i< mas.length; i++) {
            sum = sum + mas[i];
        }

        for(int i = 0; i < mas.length; i++) {
            if(i == mas.length - 1) {
                System.out.printf("%4.2f \n", mas[i]);
            }else {
                System.out.printf("%4.2f, ", mas[i]);
            }
        }
        System.out.printf("sum = %4.2f \n", sum);
    }
}
```

Результат:



```
<terminated> Main [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe (Apr 2, 2021, 7:02:26 AM - 7:02:26 AM)
0.69, 0.52, 0.87, 0.41, 0.66, 0.97, 0.60, 0.41, 0.22, 0.93
sum = 6.29
```

2.7 Решите задачу. Посчитать сумму элементов массива с четным индексом.

```
public static void main(String[] args) {
    int n = 10;
    double[] mas = new double[n];
    double sum = 0.0;

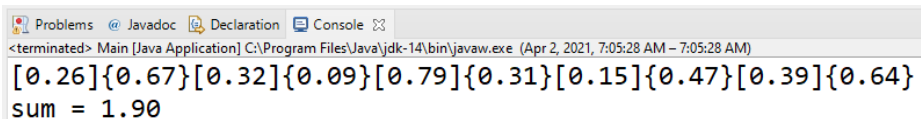
    Random rand = new Random();

    for (int i = 0; i < mas.length; i++) {
        mas[i] = rand.nextDouble();
    }

    for (int i = 0; i < mas.length; i = i + 2) {
        sum = sum + mas[i];
    }

    for (int i = 0; i < mas.length; i++) {
        if (i % 2 == 0) {
            System.out.printf("[%4.2f]", mas[i]);
        } else {
            System.out.printf("{%4.2f}", mas[i]);
        }
    }
    System.out.println();
    System.out.printf("sum = %4.2f \n", sum);
}
```

Результат:



```
<terminated> Main [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe (Apr 2, 2021, 7:05:28 AM - 7:05:28 AM)
[0.26]{0.67}[0.32]{0.09}[0.79]{0.31}[0.15]{0.47}[0.39]{0.64}
sum = 1.90
```

2.8 Решите задачу. Посчитать сумму и произведение нечетных элементов массива.

```
public static void main(String[] args) {
    int n = 10;
    double[] mas = new double[n];

    double sum = 0.0;
    double mul = 1.0;

    Random rand = new Random();
```

```

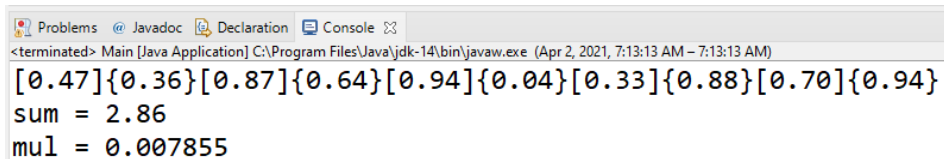
    for (int i = 0; i < mas.length; i++) {
        mas[i] = rand.nextDouble();
    }

    for (int i = 1; i < mas.length; i = i + 2) {
        sum = sum + mas[i];
        mul = mul * mas[i];
    }

    for (int i = 0; i < mas.length; i++) {
        if (i % 2 == 0) {
            System.out.printf("[%.2f]", mas[i]);
        } else {
            System.out.printf("{%.2f}", mas[i]);
        }
    }
    System.out.println();
    System.out.printf("sum = %.2f \n", sum);
    System.out.printf("mul = %.6f \n", mul);
}

```

Результат:



```

[0.47]{0.36}[0.87]{0.64}[0.94]{0.04}[0.33]{0.88}[0.70]{0.94}
sum = 2.86
mul = 0.007855

```

2.9 Решите задачу. Найти минимальное значение среди элементов массива.

```

public static void main(String[] args) {
    int n = 10;
    double[] mas = new double[n];

    double min;

    Random rand = new Random();
    for (int i = 0; i < mas.length; i++) {
        mas[i] = rand.nextDouble();
    }

    min = mas[0];
    for (int i = 1; i < mas.length; i++) {
        if (mas[i] < min) {
            min = mas[i];
        }
    }
}

```

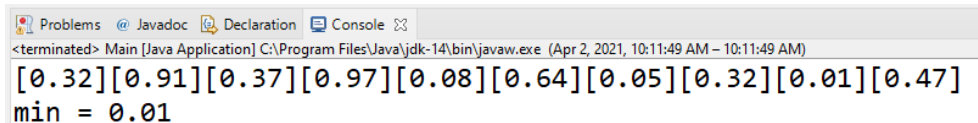
```

        for (int i = 0; i < mas.length; i++) {
            System.out.printf("[%4.2f]", mas[i]);
        }

        System.out.println();
        System.out.printf("min = %4.2f \n", min);
    }
}

```

Результат:



```

[0.32][0.91][0.37][0.97][0.08][0.64][0.05][0.32][0.01][0.47]
min = 0.01

```

2.10 Решите задачу. В целочисленной последовательности есть нулевые элементы. Создать массив из номеров этих элементов.

```

public static void main(String[] args) {
    int[] mas = new int[] { 899, 586, 212, 0, 432, 0, 495, 161,
        984, 0, 0, 754, 309, 361, 40, 14, 697, 422, 884, 0 };

    int countZero = 0;

    for (int i = 0; i < mas.length; i++) {
        if (mas[i] == 0) {
            countZero++;
        }
    }

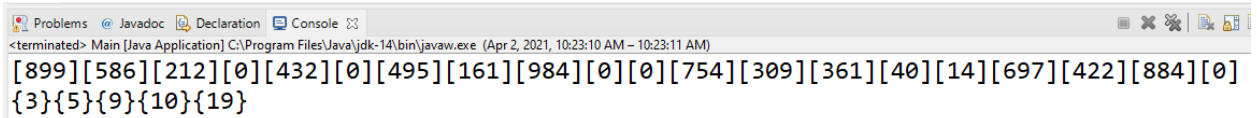
    int[] zeroPositionsArray = new int[countZero];
    for (int i = 0, j = 0; i < mas.length; i++) {
        if (mas[i] == 0) {
            zeroPositionsArray[j++] = i;
        }
    }

    for (int i = 0; i < mas.length; i++) {
        System.out.printf("[%d]", mas[i]);
    }

    System.out.println();
    for (int i = 0; i < zeroPositionsArray.length; i++) {
        System.out.printf("{%d}", zeroPositionsArray[i]);
    }
}

```


Результат:



```
<terminated> Main [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe (Apr 2, 2021, 10:23:10 AM – 10:23:11 AM)
[899][586][212][0][432][0][495][161][984][0][0][754][309][361][40][14][697][422][884][0]
{3}{5}{9}{10}{19}
```

Иногда значения элементов массива известны заранее. В этом случае можно инициализировать массив одновременно с его созданием. Есть два способа инициализации массива при объявлении:

Первый:

```
int[] ar1 = new int[] {24, 83, 72, 33, 0, 75, 67, 97, 0};
```

С помощью такого синтаксиса можно инициализировать любой создаваемый массив в любой точке программы.

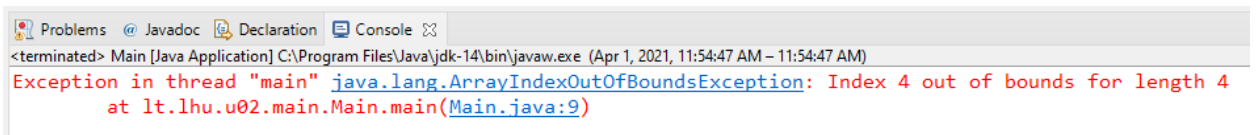
Второй:

```
int[] ar2 = {24, 83, 72, 33, 0, 75, 67, 97, 0};
```

Инициализацию таким способом можно провести только вместе с одновременным объявлением ссылки на массив.

2.11 Если в коде ошибиться и обратиться к ячейке массива, которой не существует, то JVM сгенерирует исключительную ситуацию `ArrayIndexOutOfBoundsException`.

```
public static void main(String[] args) {
    int[] ar = new int[] { 24, 83, 72, 33 };
    System.out.println(ar[4]);
}
```

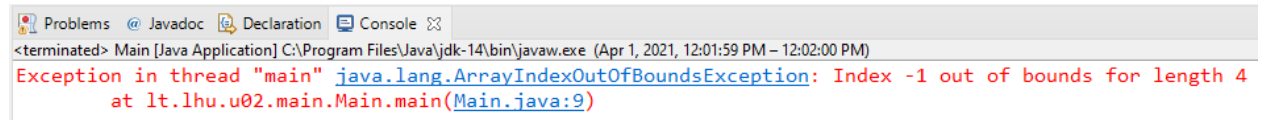


```
<terminated> Main [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe (Apr 1, 2021, 11:54:47 AM – 11:54:47 AM)
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 4 out of bounds for length 4
at lt.lhu.u02.main.Main.main(Main.java:9)
```

Данную исключительную ситуацию нельзя перехватывать. Это так называемое runtime-исключение библиотеки Java SE, при возникновении которого следует исправить (переписать) код так, чтобы выход за границы массива не осуществлялся. Иногда это исключение называют одним из исключений “кривых рук программиста” 😊.

Также исключение `ArrayIndexOutOfBoundsException` сгенерируется, если попытаться в качестве индекса написать отрицательное значение (а вот компилятор такой код скомпилирует без ошибок, т.к. для компилятора главное, чтобы индекс массива был целым числом).

```
int[] ar = new int[] { 24, 83, 72, 33 };
System.out.println(ar[-1]);
```



The screenshot shows an IDE window with a console tab. The console output indicates a Java application terminated with an exception: `Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index -1 out of bounds for length 4 at lt.lhu.u02.main.Main.main(Main.java:9)`. The IDE interface includes tabs for Problems, Javadoc, Declaration, and Console.

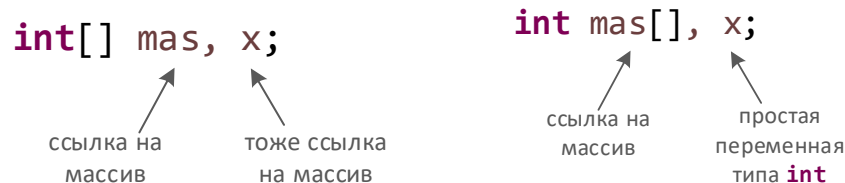
2.12 Java поддерживает синтаксис объявления ссылок на массив как таким

```
int[] mas; // java - синтаксис
```

так и вот таким.

```
int mas[]; // c++ - синтаксис
```

Разница в синтаксисе выявляется только при наличии объявления другой переменной в одном операторе.



3 Сортировки

3.1 Сортировка - это процесс упорядочивания наборов данных одного типа по возрастанию или убыванию значения какого-либо признака.

Например,

25 10 5 76 - неотсортированный массив целых чисел;

5 10 15 20 25 - отсортированный по возрастанию массив целых чисел;

25 20 15 10 5 - отсортированный по убыванию массив целых чисел.

Методов сортировки большое количество. При реализации программ и выборе алгоритма сортировки обращают внимание на быстродействие алгоритма и на минимальное использование дополнительной памяти. В качестве показателя быстродействия алгоритма используют оценки количества операций присваивания и количества операций сравнения.

Изучим два простейших алгоритма сортировки.

3.2 Сортировка выбором (Selection Sort)

Для сортировки прямым выбором (по возрастанию) в массиве необходимо найти наименьший элемент и поменять его местами с элементом, стоящим на первой позиции. Далее снова найти наименьший элемент массива (но не учитывая тот, который уже стоит на первом месте) и поменять его со вторым элементом местами. Этот алгоритм повторяется до тех пор, пока для поиска наименьшего не останется сравнить два последних элемента массива.

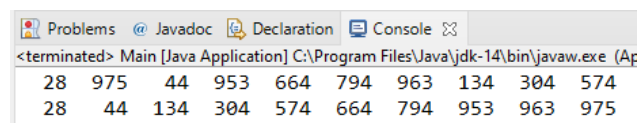
```
public static void main(String[] args) {
    int[] ar = new int[10];
    Random rand = new Random();

    for (int i = 0; i < ar.length; i++) {
        ar[i] = rand.nextInt(1000);
    }

    for (int i = 0; i < ar.length; i++) {
        System.out.printf("%4d ", ar[i]);
    }
    System.out.println();
    //////////////////////////////////////
    int min, temp;
    for (int i = 0; i < ar.length - 1; i++) {
        min = i;

        for (int j = i + 1; j < ar.length; j++) {
            if (ar[j] < ar[min]) {
                min = j;
            }
        }
        temp = ar[i];
        ar[i] = ar[min];
        ar[min] = temp;
    }
    //////////////////////////////////////
    for (int i = 0; i < ar.length; i++) {
        System.out.printf("%4d ", ar[i]);
    }
    System.out.println();
}
```

Результат:



```
<terminated> Main [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe (Ap
28  975  44  953  664  794  963  134  304  574
28  44  134  304  574  664  794  953  963  975
```

Если выводить массив после изменений на каждой итерации поиска, то можно увидеть пошаговый процесс упорядочивания элементов массива.

```
public static void main(String[] args) {
    int[] ar = new int[10];
    Random rand = new Random();

    for (int i = 0; i < ar.length; i++) {
        ar[i] = rand.nextInt(1000);
    }

    System.out.print("Исходный массив:\t\t ");
    for (int i = 0; i < ar.length; i++) {
        System.out.printf("%4d ", ar[i]);
    }
    System.out.println();

    //////////////////////////////////////
    int min, temp;
    for (int i = 0; i < ar.length - 1; i++) {
        min = i;

        for (int j = i + 1; j < ar.length; j++) {
            if (ar[j] < ar[min]) {
                min = j;
            }
        }
        temp = ar[i];
        ar[i] = ar[min];
        ar[min] = temp;

        System.out.print("Массив после " + i + "-й итерации:\t ");
        for (int k = 0; k < ar.length; k++) {
            System.out.printf("%4d ", ar[k]);
        }
        System.out.println();
    }
    //////////////////////////////////////

    System.out.print("Отсортированный массив:\t\t ");
    for (int i = 0; i < ar.length; i++) {
        System.out.printf("%4d ", ar[i]);
    }
    System.out.println();
}
```

Результат:

<terminated> Main [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe (Apr 1, 2021, 1:12:47 PM – 1:12:47 PM)										
Исходный массив:	705	826	752	769	245	257	292	268	744	819
Массив после 0-й итерации:	245	826	752	769	705	257	292	268	744	819
Массив после 1-й итерации:	245	257	752	769	705	826	292	268	744	819
Массив после 2-й итерации:	245	257	268	769	705	826	292	752	744	819
Массив после 3-й итерации:	245	257	268	292	705	826	769	752	744	819
Массив после 4-й итерации:	245	257	268	292	705	826	769	752	744	819
Массив после 5-й итерации:	245	257	268	292	705	744	769	752	826	819
Массив после 6-й итерации:	245	257	268	292	705	744	752	769	826	819
Массив после 7-й итерации:	245	257	268	292	705	744	752	769	826	819
Массив после 8-й итерации:	245	257	268	292	705	744	752	769	819	826
Отсортированный массив:	245	257	268	292	705	744	752	769	819	826

3.3 Сортировка пузырьком (Bubble Sort)

Для сортировки пузырьком (по возрастанию) начиная с начала массива просматриваются попарно по 2 элемента (первый со вторым, второй с третьим, третий с четвертым и т.д.). Если второй элемент в паре меньше первого элемента – элементы меняются местами (так делается для всех пар сравнений). После того, как были сравнены все пары элементов в массиве процесс повторяется. Таким образом элементы с самыми маленькими значениями потихоньку перемещаются справа налево, а более тяжелые элементы перемещаются слева направо, то есть они как будто всплывают. Сортировка прекращается, когда при попарном сравнении всех элементов не было ни одной перестановки.

```
public static void main(String[] args) {
    int[] ar = new int[10];
    Random rand = new Random();

    for (int i = 0; i < ar.length; i++) {
        ar[i] = rand.nextInt(1000);
    }

    System.out.print("Исходный массив:\t\t ");
    for (int i = 0; i < ar.length; i++) {
        System.out.printf("%4d ", ar[i]);
    }
    System.out.println();
    //////////////////////////////////////
    boolean sorted = false;
    int temp;
    while(!sorted) {
        sorted = true;
```

```

        for (int i = 0; i < ar.length - 1; i++) {
            if (ar[i] > ar[i+1]) {
                temp = ar[i];
                ar[i] = ar[i+1];
                ar[i+1] = temp;
                sorted = false;
            }
        }
    }

    //////////////////////////////////////
    System.out.print("Отсортированный массив:\t\t ");
    for (int i = 0; i < ar.length; i++) {
        System.out.printf("%4d ", ar[i]);
    }
    System.out.println();
}

```

Результат:

```

<terminated> Main [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe (Apr 1, 2021, 2:08:05 PM - 2:08:05 PM)
Исходный массив:          955  364  121  518  480  550  27  295  309  252
Отсортированный массив:    27  121  252  295  309  364  480  518  550  955

```

Если выводить массив после изменений на каждой итерации поиска, то можно увидеть пошаговый процесс упорядочивания элементов массива.

```

public static void main(String[] args) {
    int[] ar = new int[10];
    Random rand = new Random();

    for (int i = 0; i < ar.length; i++) {
        ar[i] = rand.nextInt(1000);
    }

    System.out.print("Исходный массив:\t ");
    for (int i = 0; i < ar.length; i++) {
        System.out.printf("%4d ", ar[i]);
    }
    System.out.println();
    //////////////////////////////////////
    boolean sorted = false;
    int temp;
    while(!sorted) {
        sorted = true;
        for (int i = 0; i < ar.length - 1; i++) {

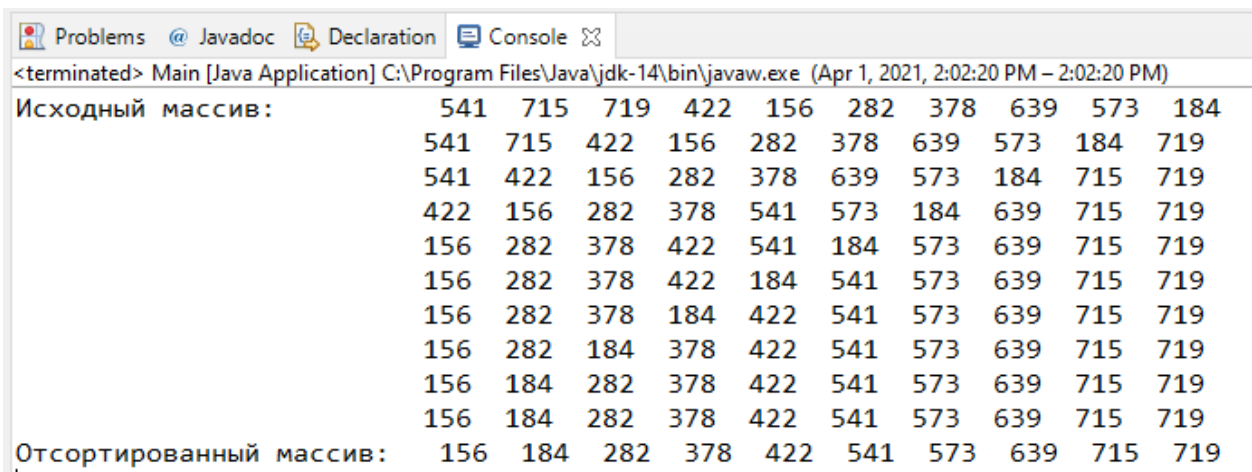
```

```

        if (ar[i] > ar[i+1]) {
            temp = ar[i];
            ar[i] = ar[i+1];
            ar[i+1] = temp;
            sorted = false;
        }
    }
    System.out.print("\t\t\t");
    for (int k = 0; k < ar.length; k++) {
        System.out.printf("%4d ", ar[k]);
    }
    System.out.println();
}
////////////////////////////////////
System.out.print("Отсортированный массив:\t ");
for (int i = 0; i < ar.length; i++) {
    System.out.printf("%4d ", ar[i]);
}
System.out.println();
}

```

Результат:



```

<terminated> Main [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe (Apr 1, 2021, 2:02:20 PM – 2:02:20 PM)
Исходный массив:      541  715  719  422  156  282  378  639  573  184
                      541  715  422  156  282  378  639  573  184  719
                      541  422  156  282  378  639  573  184  715  719
                      422  156  282  378  541  573  184  639  715  719
                      156  282  378  422  541  184  573  639  715  719
                      156  282  378  422  184  541  573  639  715  719
                      156  282  378  184  422  541  573  639  715  719
                      156  282  184  378  422  541  573  639  715  719
                      156  184  282  378  422  541  573  639  715  719
                      156  184  282  378  422  541  573  639  715  719
Отсортированный массив: 156  184  282  378  422  541  573  639  715  719

```

3.4 Использование возможностей Java для сортировки

Метод **sort()** из класса **Arrays** использует усовершенствованный алгоритм быстрой сортировки (quicksort), который эффективен в большинстве случаев. Для того чтобы отсортировать массив, необходимо написать всего одну строку.

```
Arrays.sort(arr); // где arr это имя массива
```

```

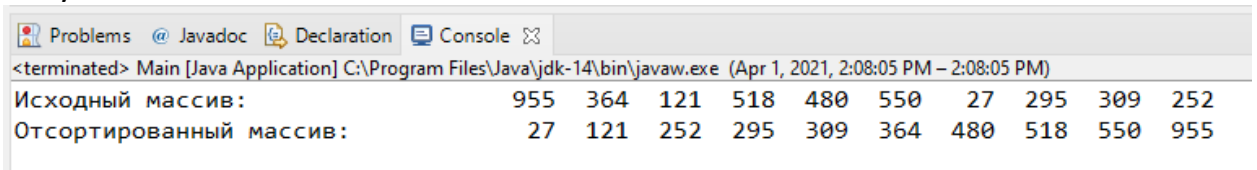
public static void main(String[] args) {
    int[] ar = new int[10];
    Random rand = new Random();

    for (int i = 0; i < ar.length; i++) {
        ar[i] = rand.nextInt(1000);
    }

    System.out.print("Исходный массив:\t\t ");
    for (int i = 0; i < ar.length; i++) {
        System.out.printf("%4d ", ar[i]);
    }
    System.out.println();
    //////////////////////////////////////
    Arrays.sort(ar);
    //////////////////////////////////////
    System.out.print("Отсортированный массив:\t\t ");
    for (int i = 0; i < ar.length; i++) {
        System.out.printf("%4d ", ar[i]);
    }
    System.out.println();
}

```

Результат:



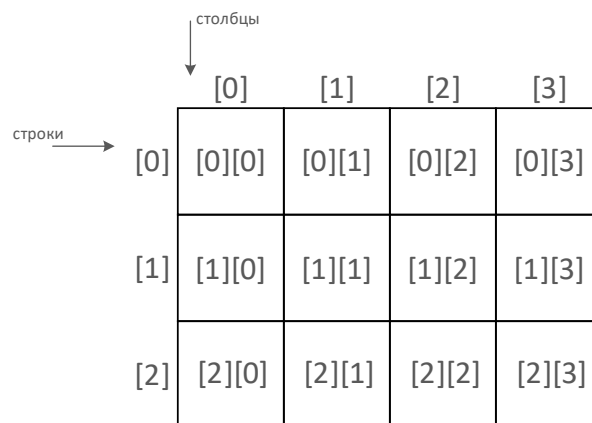
```

<terminated> Main [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe (Apr 1, 2021, 2:08:05 PM - 2:08:05 PM)
Исходный массив:          955  364  121  518  480  550   27  295  309  252
Отсортированный массив:    27  121  252  295  309  364  480  518  550  955

```


4.1 В Java существуют массивы и большей размерности, чем 1 – массивы массивов. Технически, двумерных массивов (матриц) с Java нет, хотя использовать матрицу как модель массива массивов вполне допустимо. Начнем изучать работу с массивами массивов используя модель – матрицу.

Матрица или двумерный массив - это набор однотипных элементов, доступ к элементам которого осуществляется по двум индексам: индексу строки и индексу столбца.



Синтаксис объявления в Java двумерного массива (массива массивов) следующий:

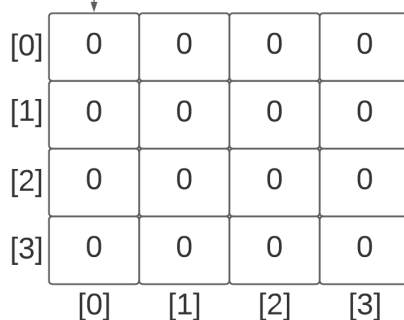
```
тип_массива[][] ссылка_на_массив_массивов =
    new тип_массива[число_строк][число_столбцов];
```

```
int[][] mas = new int[3][4];
double[][] ar = new double[5][6];
```

По аналогии с одномерными массивами все ячейки инициализируются значением по умолчанию.

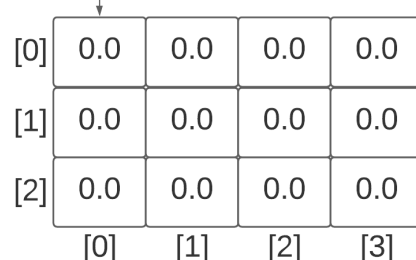
```
int[][] mas = new int[4][4];
```

mas



```
double[][] mas = new double[3][4];
```

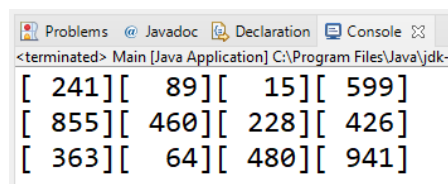
mas



- 4.2 Решим задачу. Создать двумерный массив из 3-х строк и 4-х столбцов. Инициализировать его случайными числами и вывести на консоль.

```
public static void main(String[] args) {  
  
    int[][] mas = new int[3][4];  
  
    Random rand = new Random();  
  
    for(int i=0; i<mas.length; i++) {  
        for(int j=0; j<mas[i].length; j++) {  
            mas[i][j] = rand.nextInt(1000);  
        }  
    }  
  
    for(int i=0; i<mas.length; i++) {  
        for(int j=0; j<mas[i].length; j++) {  
            System.out.printf("[%4d]", mas[i][j]);  
        }  
        System.out.println();  
    }  
}
```

Результат:



```
<terminated> Main [Java Application] C:\Program Files\Java\jdk-  
[ 241][ 89][ 15][ 599]  
[ 855][ 460][ 228][ 426]  
[ 363][ 64][ 480][ 941]
```

- 4.3 Для определения количества строк в массиве массивов следует написать

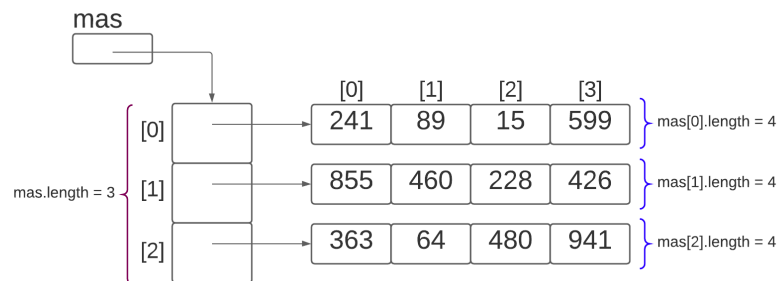
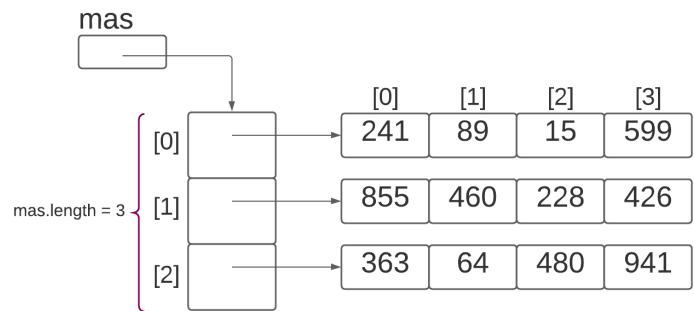
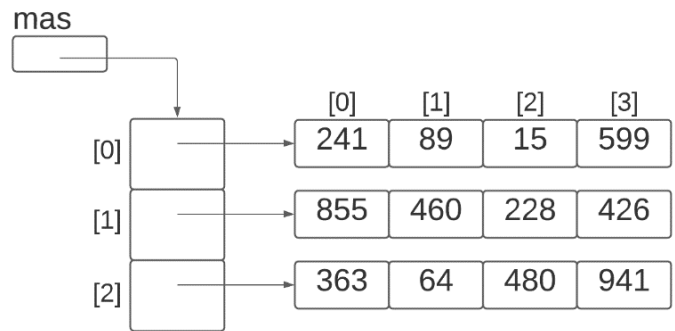
`mas.length`

Для определения количества элементов в строке массива (количества столбцов) используется следующий синтаксис:

`mas[i].length`
где `i` – индекс строки

Такой синтаксис определяется представлением двумерного массива в памяти как массива массивов, т.е. массива, элементами которого являются другие массивы. Таким образом при объявлении двумерного массива `int[][] mas = new int[3][4];` ссылка `mas` ссылается на одномерный массив; ячейки этого

массива являются, в свою очередь, ссылками на другие одномерные массивы.



Изучим код примера:

```
public static void main(String[] args) {  
  
    int[][] mas = new int[3][4];  
  
    Random rand = new Random();  
  
    for(int i=0; i<mas.length; i++) {  
        for(int j=0; j<mas[i].length; j++) {  
            mas[i][j] = rand.nextInt(1000);  
        }  
    }  
}
```

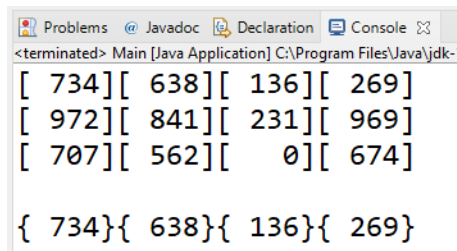
```

    for(int i=0; i<mas.length; i++) {
        for(int j=0; j<mas[i].length; j++) {
            System.out.printf("[%4d]", mas[i][j]);
        }
        System.out.println();
    }

    int[] ar = mas[0];
    System.out.println();
    for(int i = 0; i< ar.length; i++) {
        System.out.printf("{%4d}", ar[i]);
    }
}

```

Результат:



```

<terminated> Main [Java Application] C:\Program Files\Java\jdk-1
[ 734][ 638][ 136][ 269]
[ 972][ 841][ 231][ 969]
[ 707][ 562][ 0][ 674]

{ 734}{ 638}{ 136}{ 269}

```

4.4 Напишем код, который в двумерном массиве переставляет местами строки.

```

public static void main(String[] args) {
    int[][] mas = new int[3][4];

    Random rand = new Random();

    for(int i=0; i<mas.length; i++) {
        for(int j=0; j<mas[i].length; j++) {
            mas[i][j] = rand.nextInt(1000);
        }
    }

    for(int i=0; i<mas.length; i++) {
        for(int j=0; j<mas[i].length; j++) {
            System.out.printf("[%4d]", mas[i][j]);
        }
        System.out.println();
    }

    int[] ar = mas[0];
    mas[0] = mas[1];
    mas[1] = ar;
}

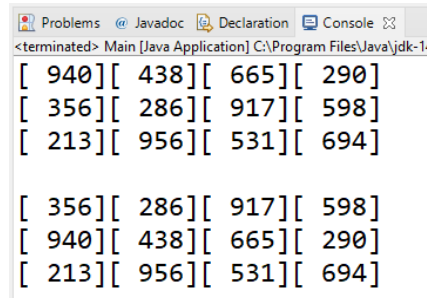
```

```

        System.out.println();
        for(int i=0; i<mas.length; i++) {
            for(int j=0; j<mas[i].length; j++) {
                System.out.printf("%4d", mas[i][j]);
            }
            System.out.println();
        }
    }
}

```

Результат:



```

<terminated> Main [Java Application] C:\Program Files\Java\jdk-1...
[ 940][ 438][ 665][ 290]
[ 356][ 286][ 917][ 598]
[ 213][ 956][ 531][ 694]

[ 356][ 286][ 917][ 598]
[ 940][ 438][ 665][ 290]
[ 213][ 956][ 531][ 694]

```

4.5 Также в Java возможно создавать двумерные массивы, где в каждой строке содержится разное количество элементов. Например:

```

public static void main(String[] args) {

    int[][] mas = new int[4][];

    int[] line1 = { 1, 2, 3, 4 };
    int[] line2 = { 5, 6, 7 };
    int[] line4 = { 8, 9 };

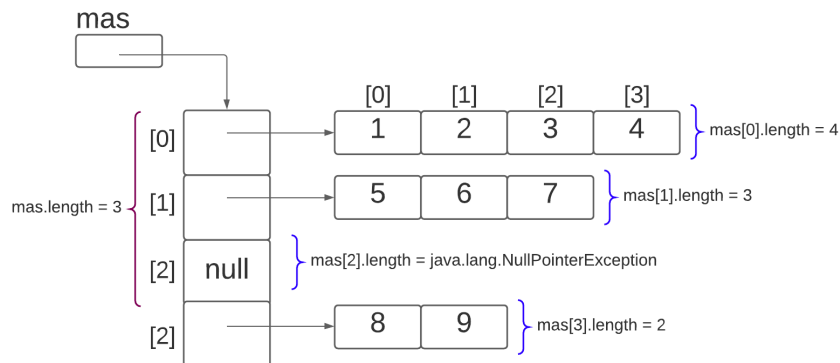
    mas[0] = line1;
    mas[1] = line2;
    mas[3] = line4;

    for (int i = 0; i < mas.length; i++) {
        if (mas[i] != null) {
            for (int j = 0; j < mas[i].length; j++) {
                System.out.printf("%4d", mas[i][j]);
            }
        }
        else {
            System.out.print("{null}");
        }
        System.out.println();
    }
}

```

Результат:

```
Problems Javadoc Declaration Console
<terminated> Main [Java Application] C:\Program Files\Java\jdk-14
[ 1][ 2][ 3][ 4]
[ 5][ 6][ 7]
{null}
[ 8][ 9]
```



4.6 Инициализация массивов массивов одновременно с его созданием выглядит следующим образом:

```
int[][] mas1 = new int[][] { {1,2,3}, {11,22,33,44}, {55,66}};
int[][] mas2 = { {1,2,3}, {11,22,33,44}, {55,66}};
```

4.7 Решите задачу. Посчитать сумму элементов в каждой строке двумерного массива.

```
public static void main(String[] args) {
    int[][] mas = new int[3][4];
    Random rand = new Random();

    for (int i = 0; i < mas.length; i++) {
        for (int j = 0; j < mas[i].length; j++) {
            mas[i][j] = rand.nextInt(1000);
        }
    }

    int sum;
    for (int i = 0; i < mas.length; i++) {
        sum = 0;
        for (int j = 0; j < mas[i].length; j++) {
            System.out.printf("[%4d]", mas[i][j]);
            sum = sum + mas[i][j];
        }
        System.out.println("    - " + sum);
    }
}
```

Результат:

```
Problems Javadoc Declaration Console
<terminated> Task02 (1) [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe
[ 853][ 372][ 754][ 873] - 2852
[ 593][ 998][ 469][ 740] - 2800
[ 26][ 836][ 334][ 323] - 1519
```

4.8 Решите задачу. Дана матрица. Вывести на экран первую и последнюю строки.

```
public static void main(String[] args) {
    int[][] mas = new int[7][5];
    Random rand = new Random();

    for (int i = 0; i < mas.length; i++) {
        for (int j = 0; j < mas[i].length; j++) {
            mas[i][j] = rand.nextInt(1000);
        }
    }

    for (int i = 0; i < mas.length; i++) {
        for (int j = 0; j < mas[i].length; j++) {
            System.out.printf("%4d", mas[i][j]);
        }
        System.out.println();
    }

    System.out.println("-----");
    for(int element : mas[0]) {
        System.out.printf("%4d", element);
    }
    System.out.println();

    System.out.println("-----");
    for(int element : mas[mas.length - 1]) {
        System.out.printf("%4d", element);
    }
    System.out.println();
}
```

Результат:

```
Problems Javadoc Declaration Console
<terminated> Task02 (1) [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe
[ 106][ 442][ 686][ 697][ 407]
[ 44][ 13][ 871][ 432][ 233]
[ 599][ 60][ 878][ 513][ 215]
[ 63][ 278][ 81][ 465][ 624]
[ 562][ 572][ 203][ 430][ 791]
[ 0][ 993][ 452][ 224][ 816]
[ 391][ 905][ 47][ 633][ 607]
-----
[ 106][ 442][ 686][ 697][ 407]
-----
[ 391][ 905][ 47][ 633][ 607]
```

4.9 Решите задачу. Дана матрица. Вывести на экран все строки с четными индексами.

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

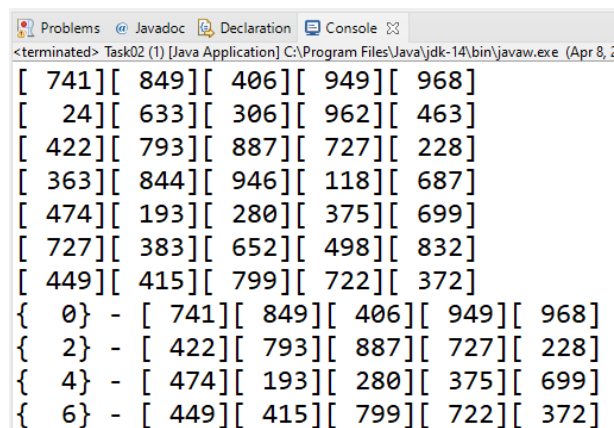
    int[][] mas = new int[7][5];
    Random rand = new Random();

    for (int i = 0; i < mas.length; i++) {
        for (int j = 0; j < mas[i].length; j++) {
            mas[i][j] = rand.nextInt(1000);
        }
    }

    for (int i = 0; i < mas.length; i++) {
        for (int j = 0; j < mas[i].length; j++) {
            System.out.printf("[%4d]", mas[i][j]);
        }
        System.out.println();
    }

    for (int i = 0; i < mas.length; i = i + 2) {
        System.out.printf("{%3d} - ", i);
        for (int j = 0; j < mas[i].length; j++) {
            System.out.printf("[%4d]", mas[i][j]);
        }
        System.out.println();
    }
}
```

Результат:



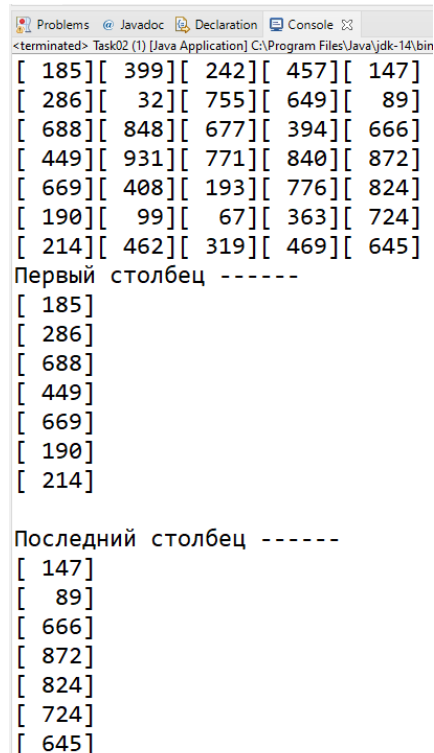
```
<terminated> Task02 (1) [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe (Apr 8, 2021)
[ 741][ 849][ 406][ 949][ 968]
[  24][ 633][ 306][ 962][ 463]
[ 422][ 793][ 887][ 727][ 228]
[ 363][ 844][ 946][ 118][ 687]
[ 474][ 193][ 280][ 375][ 699]
[ 727][ 383][ 652][ 498][ 832]
[ 449][ 415][ 799][ 722][ 372]
{  0} - [ 741][ 849][ 406][ 949][ 968]
{  2} - [ 422][ 793][ 887][ 727][ 228]
{  4} - [ 474][ 193][ 280][ 375][ 699]
{  6} - [ 449][ 415][ 799][ 722][ 372]
```


4.10 Решите задачу. Дана матрица. Вывести на экран первый и последний столбцы.

```
public static void main(String[] args) {
    int[][] mas = new int[7][5];
    Random rand = new Random();
    for (int i = 0; i < mas.length; i++) {
        for (int j = 0; j < mas[i].length; j++) {
            mas[i][j] = rand.nextInt(1000);
        }
    }

    for (int i = 0; i < mas.length; i++) {
        for (int j = 0; j < mas[i].length; j++) {
            System.out.printf("%4d", mas[i][j]);
        }
        System.out.println();
    }
    System.out.println("Первый столбец -----");
    for(int i = 0; i < mas.length; i++) {
        System.out.printf("%4d\n", mas[i][0]);
    }
    System.out.println();
    System.out.println("Последний столбец -----");
    for(int i = 0; i < mas.length; i++) {
        System.out.printf("%4d\n", mas[i][mas[i].length - 1]);
    }
    System.out.println();
}
```

Результат:



```
<terminated> Task02 (1) [Java Application] C:\Program Files\Java\jdk-14\bin
[ 185][ 399][ 242][ 457][ 147]
[ 286][  32][ 755][ 649][  89]
[ 688][ 848][ 677][ 394][ 666]
[ 449][ 931][ 771][ 840][ 872]
[ 669][ 408][ 193][ 776][ 824]
[ 190][  99][  67][ 363][ 724]
[ 214][ 462][ 319][ 469][ 645]
Первый столбец -----
[ 185]
[ 286]
[ 688]
[ 449]
[ 669]
[ 190]
[ 214]
Последний столбец -----
[ 147]
[  89]
[ 666]
[ 872]
[ 824]
[ 724]
[ 645]
```

4.11 Решите задачу. Дана матрица. Вывести на экран все нечетные столбцы, у которых первый элемент больше последнего.

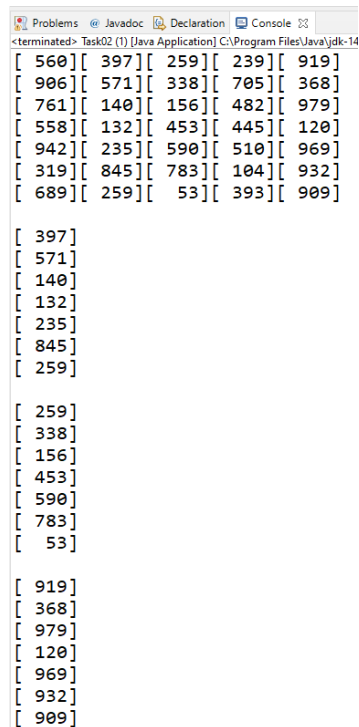
```
public static void main(String[] args) {
    int[][] mas = new int[7][5];
    Random rand = new Random();

    for (int i = 0; i < mas.length; i++) {
        for (int j = 0; j < mas[i].length; j++) {
            mas[i][j] = rand.nextInt(1000);
        }
    }

    for (int i = 0; i < mas.length; i++) {
        for (int j = 0; j < mas[i].length; j++) {
            System.out.printf("%4d", mas[i][j]);
        }
        System.out.println();
    }
    System.out.println();

    for (int j = 0; j < mas[0].length; j++) {
        if (mas[0][j] > mas[mas.length - 1][j]) {
            for (int k = 0; k < mas.length; k++) {
                System.out.printf("%4d\\n", mas[k][j]);
            }
            System.out.println();
        }
    }
}
```

Результат:



```
<terminated> Task02 (1) [Java Application] C:\Program Files\Java\jdk-14
[ 560][ 397][ 259][ 239][ 919]
[ 906][ 571][ 338][ 705][ 368]
[ 761][ 140][ 156][ 482][ 979]
[ 558][ 132][ 453][ 445][ 120]
[ 942][ 235][ 590][ 510][ 969]
[ 319][ 845][ 783][ 104][ 932]
[ 689][ 259][ 53][ 393][ 909]

[ 397]
[ 571]
[ 140]
[ 132]
[ 235]
[ 845]
[ 259]

[ 259]
[ 338]
[ 156]
[ 453]
[ 590]
[ 783]
[ 53]

[ 919]
[ 368]
[ 979]
[ 120]
[ 969]
[ 932]
[ 909]
```

или можно решить вот так

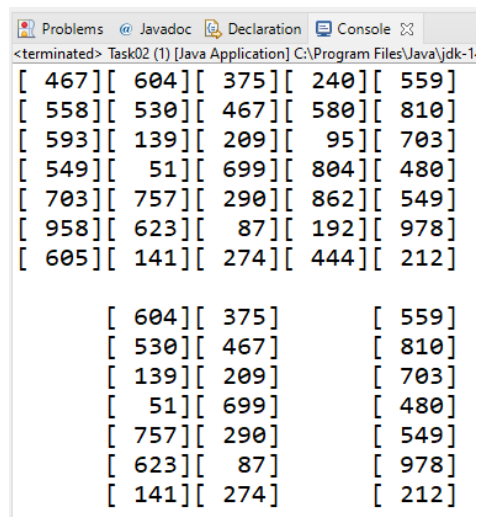
```
public static void main(String[] args) {
    int[][] mas = new int[7][5];
    Random rand = new Random();

    for (int i = 0; i < mas.length; i++) {
        for (int j = 0; j < mas[i].length; j++) {
            mas[i][j] = rand.nextInt(1000);
        }
    }

    for (int i = 0; i < mas.length; i++) {
        for (int j = 0; j < mas[i].length; j++) {
            System.out.printf("[%4d]", mas[i][j]);
        }
        System.out.println();
    }
    System.out.println();

    for (int i = 0; i < mas.length; i++) {
        for (int j = 0; j < mas[i].length; j++) {
            if (mas[0][j] > mas[mas.length - 1][j]) {
                System.out.printf("[%4d]", mas[i][j]);
            } else {
                System.out.printf("      ");
            }
        }
        System.out.println();
    }
    System.out.println();
}
```

Результат:



```
<terminated> Task02 (1) [Java Application] C:\Program Files\Java\jdk-1.
[ 467][ 604][ 375][ 240][ 559]
[ 558][ 530][ 467][ 580][ 810]
[ 593][ 139][ 209][ 95][ 703]
[ 549][ 51][ 699][ 804][ 480]
[ 703][ 757][ 290][ 862][ 549]
[ 958][ 623][ 87][ 192][ 978]
[ 605][ 141][ 274][ 444][ 212]

      [ 604][ 375]      [ 559]
      [ 530][ 467]      [ 810]
      [ 139][ 209]      [ 703]
      [ 51][ 699]      [ 480]
      [ 757][ 290]      [ 549]
      [ 623][ 87]      [ 978]
      [ 141][ 274]      [ 212]
```

4.12 Решите задачу. Дан двумерный массив 5×5. Найти сумму отрицательных нечетных элементов.

```
public static void main(String[] args) {
    int[][] mas = new int[5][5];
    Random rand = new Random();

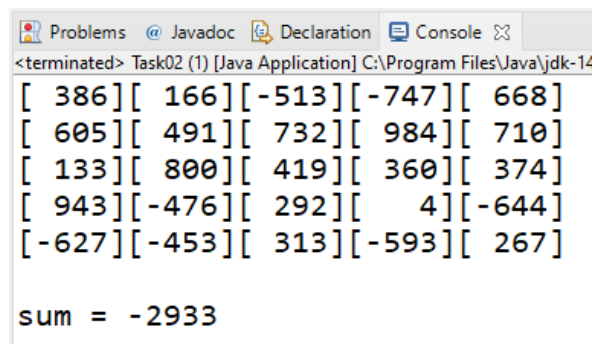
    for (int i = 0; i < mas.length; i++) {
        for (int j = 0; j < mas[i].length; j++) {
            mas[i][j] = rand.nextInt(1000);
            if(rand.nextInt(1000) % 3 == 0) {
                mas[i][j] = -mas[i][j];
            }
        }
    }

    for (int i = 0; i < mas.length; i++) {
        for (int j = 0; j < mas[i].length; j++) {
            System.out.printf("[%4d]", mas[i][j]);
        }
        System.out.println();
    }
    System.out.println();

    int sum = 0;
    for (int i = 0; i < mas.length; i++) {
        for (int j = 0; j < mas[i].length; j++) {
            if(Math.abs(mas[i][j]) % 2 == 1 && mas[i][j] < 0) {
                sum += mas[i][j];
            }
        }
    }

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

Результат:



```
<terminated> Task02 (1) [Java Application] C:\Program Files\Java\jdk-14
[ 386][ 166][-513][-747][ 668]
[ 605][ 491][ 732][ 984][ 710]
[ 133][ 800][ 419][ 360][ 374]
[ 943][-476][ 292][  4][-644]
[-627][-453][ 313][-593][ 267]

sum = -2933
```

4.13 Решите задачу. Дана матрица размера $m \times n$. Вывести ее элементы в следующем порядке: первая строка справа налево, вторая строка слева направо, третья строка справа налево и так далее.

```
public static void main(String[] args) {
    Random rand = new Random();

    int m = rand.nextInt(10)+1;
    int n = rand.nextInt(10)+1;
    int[][] mas = new int[m][n];

    for (int i = 0; i < mas.length; i++) {
        for (int j = 0; j < mas[i].length; j++) {
            mas[i][j] = rand.nextInt(1000);
            if(rand.nextInt(1000) % 3 == 0) {
                mas[i][j] = -mas[i][j];
            }
        }
    }

    for (int i = 0; i < mas.length; i++) {
        for (int j = 0; j < mas[i].length; j++) {
            System.out.printf("[%4d]", mas[i][j]);
        }
        System.out.println();
    }
    System.out.println();

    for (int i = 0; i < mas.length; i++) {
        for (int j = 0; j < mas[i].length; j++) {
            System.out.printf("[%4d]", mas[i][j]);
        }
        System.out.println();
        i++;
        if (i >= mas.length) {
            break;
        }

        for (int j = mas[i].length - 1; j >=0; j--) {
            System.out.printf("[%4d]", mas[i][j]);
        }
        System.out.println();
    }
    System.out.println();
}
```

Результат:

```
Problems Javadoc Declaration Console
<terminated> Task02 (1) [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe (Apr 9, 2021, 10:42:55 PM)
[ 777][ 713][-920][-885][ 198][ 254][ 859][ 868][ 527][ -96]
[ 382][-496][ 918][ 712][ 833][-426][ 750][ 110][ 865][ 76]
[ 939][ 86][ 652][ 683][-852][-950][-226][ 277][-356][ -14]
[-613][ 747][-964][ 318][ 641][-280][-714][-314][-716][-575]
[ 582][ 273][ 534][-984][-959][ 145][ 426][ 683][-321][ 45]
[-299][ 218][ 25][-447][ 327][-827][ -47][ 445][ 399][ 500]
[-562][ 55][-724][-533][-783][-790][ 179][ -57][-770][ 351]

[ 777][ 713][-920][-885][ 198][ 254][ 859][ 868][ 527][ -96]
[ 76][ 865][ 110][ 750][-426][ 833][ 712][ 918][-496][ 382]
[ 939][ 86][ 652][ 683][-852][-950][-226][ 277][-356][ -14]
[-575][-716][-314][-714][-280][ 641][ 318][-964][ 747][-613]
[ 582][ 273][ 534][-984][-959][ 145][ 426][ 683][-321][ 45]
[ 500][ 399][ 445][ -47][-827][ 327][-447][ 25][ 218][-299]
[-562][ 55][-724][-533][-783][-790][ 179][ -57][-770][ 351]
```

4.14 Решите задачу. Получить квадратную матрицу порядка n:

$$\begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & n-1 \end{pmatrix}$$

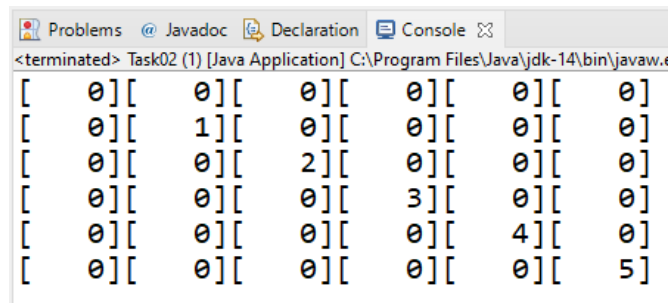
```
public static void main(String[] args) {
    int n = 6;

    int[][] mas = new int[n][n];

    for (int i = 0; i < mas.length; i++) {
        mas[i][i] = i;
    }

    for (int i = 0; i < mas.length; i++) {
        for (int j = 0; j < mas[i].length; j++) {
            System.out.printf("%4d", mas[i][j]);
        }
        System.out.println();
    }
    System.out.println();
}
```

Результат:



```
<terminated> Task02 (1) [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe
[ 0][ 0][ 0][ 0][ 0][ 0]
[ 0][ 1][ 0][ 0][ 0][ 0]
[ 0][ 0][ 2][ 0][ 0][ 0]
[ 0][ 0][ 0][ 3][ 0][ 0]
[ 0][ 0][ 0][ 0][ 4][ 0]
[ 0][ 0][ 0][ 0][ 0][ 5]
```

4.15 Решите задачу. Сформировать квадратную матрицу порядка n по заданному образцу(n - четное):

$$\begin{pmatrix} 1 & 2 & 3 & \dots & n \\ n & n-1 & n-2 & \dots & 1 \\ 1 & 2 & 3 & \dots & n \\ n & n-1 & n-2 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ n & n-1 & n-2 & \dots & 1 \end{pmatrix}$$

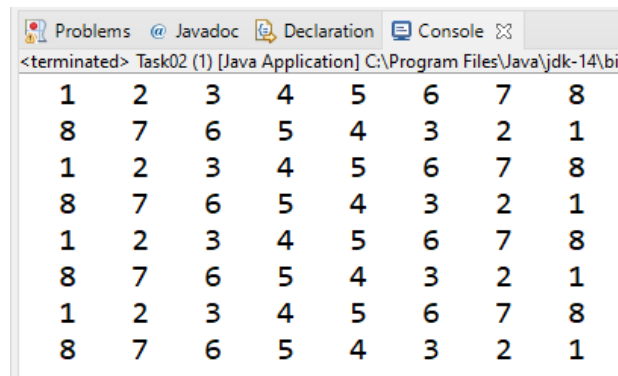
```
public static void main(String[] args) {
    int n = 8;

    int[][] mas = new int[n][n];

    for (int i = 0; i < mas.length; i=i+2) {
        for (int j = 0; j < mas[i].length; j++) {
            mas[i][j] = j+1;
            mas[i + 1][j] = mas[i].length - j;
        }
    }

    for (int i = 0; i < mas.length; i++) {
        for (int j = 0; j < mas[i].length; j++) {
            System.out.printf("%3d ", mas[i][j]);
        }
        System.out.println();
    }
    System.out.println();
}
```

Результат:



```
<terminated> Task02 (1) [Java Application] C:\Program Files\Java\jdk-14\bin\java.exe
1 2 3 4 5 6 7 8
8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8
8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8
8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8
8 7 6 5 4 3 2 1
```

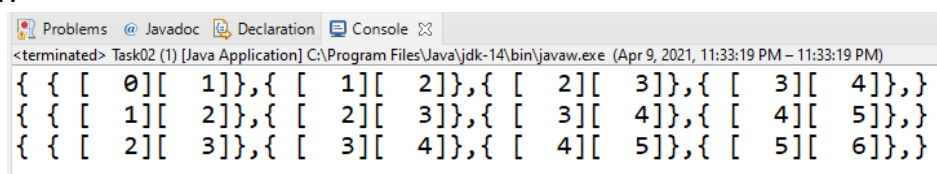
4.16 В Java можно создавать массивы любой размерности. Рассмотрим пример работы с трехмерным массивом (массивом массивов массивов):

```
public static void main(String[] args) {
    int[][][] mas = new int[3][4][2];

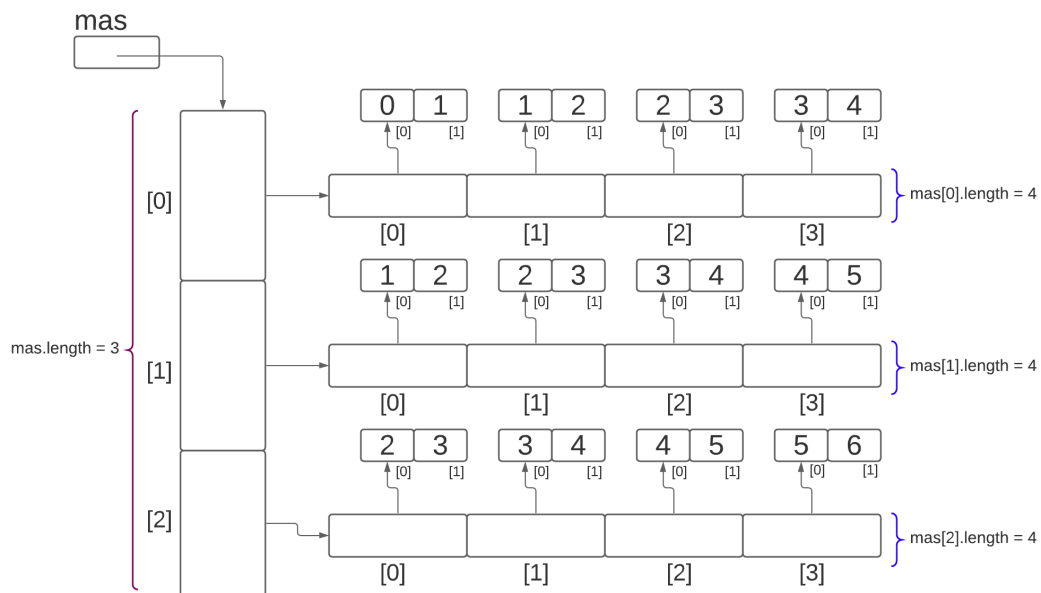
    for (int i = 0; i < mas.length; i++) {
        for (int j = 0; j < mas[i].length; j++) {
            for (int k = 0; k < mas[i][j].length; k++) {
                mas[i][j][k] = i + j + k;
            }
        }
    }

    for (int i = 0; i < mas.length; i++) {
        System.out.print("{ ");
        for (int j = 0; j < mas[i].length; j++) {
            System.out.print("{ ");
            for (int k = 0; k < mas[i][j].length; k++) {
                System.out.printf("[%3d]", mas[i][j][k]);
            }
            System.out.print("},");
        }
        System.out.print("}\n");
    }
    System.out.println();
}
```

Результат:



```
<terminated> Task02 (1) [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe (Apr 9, 2021, 11:33:19 PM - 11:33:19 PM)
{ { [ 0][ 1]}, { [ 1][ 2]}, { [ 2][ 3]}, { [ 3][ 4]}, },
{ { [ 1][ 2]}, { [ 2][ 3]}, { [ 3][ 4]}, { [ 4][ 5]}, },
{ { [ 2][ 3]}, { [ 3][ 4]}, { [ 4][ 5]}, { [ 5][ 6]}, },
}
```

4.17 Кроме массивов примитивных типов в Java можно объявлять массивы ссылочных типов данных. Тогда типом конечной ячейки является ссылка на объект. (Более подробно массивы ссылочных типов данных рассматриваются в теме “Классы и объекты”).

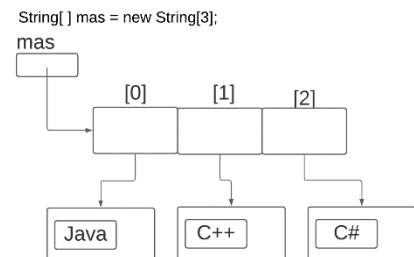
```
public static void main(String[] args) {

    String[] mas = new String[3];

    mas[0] = new String("Java");
    mas[1] = new String("C++");
    mas[2] = new String("C#");

    for(String s : mas) {
        System.out.print(s + " ");
    }
}
```

Результат:



```
Problems Javadoc Declaration Console
<terminated> Task02 (1) [Java Application] C:\Program Files\Java\
Java C++ C#
```

Подготовка рабочего пространства для решения задач

1. Создайте проект с именем Unit04[Surname] (например, Unit04Ivanov).
2. Далее создайте класс Task01 и поместите его в пакет `by.epam.unit04.main`.
3. Аналогично создайте еще 9 классов Task02 ... Task10.
4. В каждом классе создайте метод `main` и пишите в нем код решения задачи из списка задач с таким же номером, как и у класса.

Список задач

Задача 1

Дана последовательность натуральных чисел a_1, a_2, \dots, a_n . Создать массив из четных чисел этой последовательности. Если таких чисел нет, то вывести сообщение об этом факте.

Задача 2

Дана последовательность чисел a_1, a_2, \dots, a_n . Указать наименьшую длину числовой оси, содержащую все эти числа.

Задача 3

Дан массив действительных чисел, размерность которого N . Подсчитать, сколько в нем отрицательных, положительных и нулевых элементов.

Задача 4

Реализовать сортировки выбором или пузырьком по убыванию.

Задача 5

Реализовать алгоритм сортировки вставками.

Задача 6

Дан двумерный массив $n \times m$ элементов. Определить, сколько раз встречается число 7 среди элементов массива.

Задача 7

Дана квадратная матрица. Вывести на экран элементы, стоящие на диагонали.

Задача 8

Дана матрица. Вывести k-ю строку и p-й столбец матрицы.

Задача 9

Сформировать квадратную матрицу порядка n по заданному образцу(n - четное):

$$\begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 & 1 \\ 0 & 0 & 0 & \dots & 0 & 2 & 0 \\ 0 & 0 & 0 & \dots & 3 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & n-1 & 0 & \dots & 0 & 0 & 0 \\ n & 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix}$$

Задача 10

Сформировать квадратную матрицу порядка n по заданному образцу(n - четное):

$$\begin{pmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & \dots & 0 & 0 & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 & 1 \end{pmatrix}$$