

# Java масиви

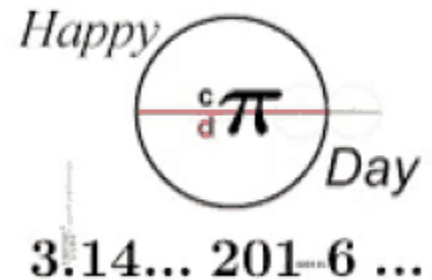
40	55	63	17	22	68	89	97	89
0	1	2	3	4	5	6	7	8

<- Array Indices

**Array Length = 9**

**First Index = 0**

**Last Index = 8**

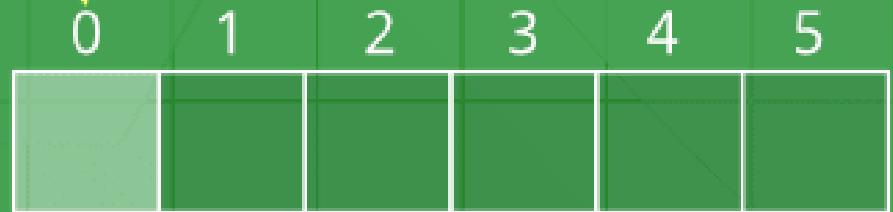


array variable

`arr [ 0 ];`

index of the element  
to be accessed

arr



# Масив с 12 елемента (дължината на масива е 12).

Name of array (c) →	c[ 0 ]	-45
	c[ 1 ]	6
	c[ 2 ]	0
	c[ 3 ]	72
	c[ 4 ]	1543
	c[ 5 ]	-89
	c[ 6 ]	0
	c[ 7 ]	62
	c[ 8 ]	-3
	c[ 9 ]	1
	c[ 10 ]	6453
Index (or subscript) of the element in array c →	c[ 11 ]	78

# Декларирайте масив

- Масивите се използват за съхранение на няколко стойности в една променлива, вместо деклариране на отделни променливи за всяка стойност.
- За да декларирайте масив, дефинирайте типа променлива с **квадратни скоби** :

```
String[] cars;
```

# Инициализация на масив

- Вече сме декларирали променлива, която съдържа масив от низове.
- За да добавим стойности към него, поставяме стойностите в списък, разделен със запетая, вътре в къдрави скоби:

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
```

Unsorted Array



# Инициализация на масив

- За да създадете масив от цели числа, можете да напишете:

```
int[] myNum = {10, 20, 30, 40};
```



# Достъп до елементите на масив

- Достъпът до елементите на масив става с поредния номер на елементите на масива /index/
- Индексацията на елементите на масива започва от 0.
- [0] е първият елемент. [1] е вторият елемент и т.н.

---

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};  
System.out.println(cars[0]);  
// Outputs Volvo
```

# Промяна на елемент на масив

- За да промените стойността на конкретен елемент, трябва да се използва индекса на елемента:
- *пример1*

`cars[0] = "Opel";`

- *пример2*

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};  
cars[0] = "Opel";  
System.out.println(cars[0]);  
// Now outputs Opel instead of Volvo
```



# Дължина на масива

- Броят на елементите определя **дължината на масива**
- Веднъж създаден, дължината на масива остава неизменна



# Дължина на масива

- За да разберете колко елементи има масив, използвайте **length** свойството на масива:

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};  
System.out.println(cars.length);  
// Outputs 4
```

40	55	63	17	22	68	89	97	89
0	1	2	3	4	5	6	7	8

<- Array Indices

**Array Length = 9**

**First Index = 0**

**Last Index = 8**

# Преминаване през масив

- Можете да прегледате елементите от масива с for цикъла и да използвате length свойството, за да укажете колко пъти трябва да работи цикълът.
- Следващият пример извежда всички елементи от масива на **автомобилите** :

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};  
for (int i = 0; i < cars.length; i++) {  
    System.out.println(cars[i]);  
}
```

# Деклариране и създаване на масиви

## /Instantiating an Array in Java/

- Масивите са **обекти от референтен тип**
- **Съхраняват се в Heap паметта**, където заемат непрекъсната област
- **Създават се динамично** с ключовата дума **new**
- Пример за масив с елементи от примитивен тип

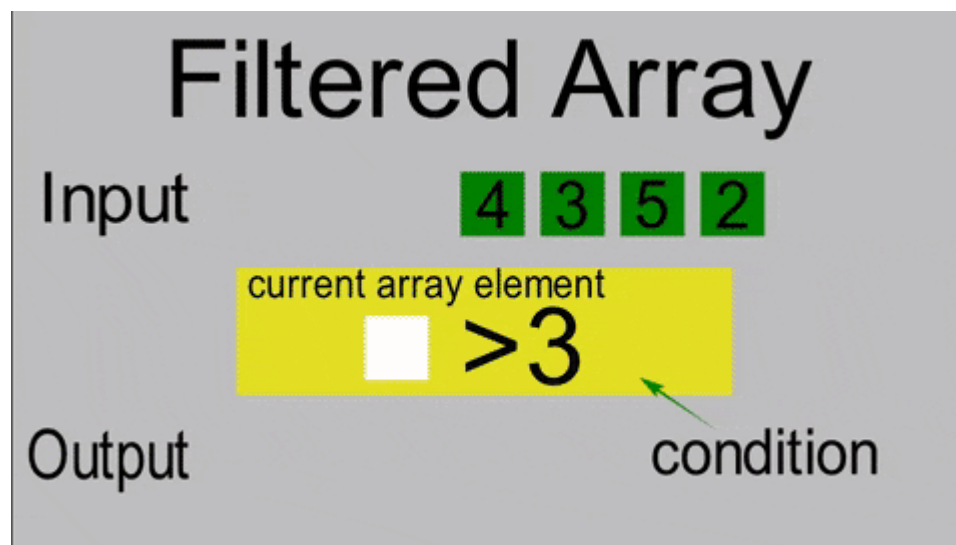
```
int c[] = new int [ 12];
```

Еквивалентно на:

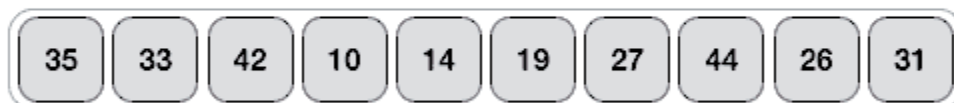
```
int c[]; // декларира масив променлива  
c = new int[ 12]; // създава масив
```

- Пример за масив с елементи от референтен тип
- `String b[] = new String[ 100];`

# Примери за използване на масиви



Unsorted Array



# Примери за използване на масиви

- **Основни стъпки в използването на масиви**

1. Деклариране на масиви
2. Създаване на масиви
3. Инициализиране елементите на масиви
4. Обработка на елементи на масив

# Създаване на масив и инициализиране на елементите с 0 by Default

```
3
4 public class InitArray
5 {
6     public static void main( String args[] )
7     {
8         int array[]; // declare array named array
9
10        array = new int[ 10 ]; // create the space for array
11
12        System.out.printf( "%s%8s\n", "Index", "Value" ); // column headings
13
14        // output each array element's value
15        for ( int counter = 0; counter < array.length; counter++ )
16            System.out.printf( "%5d%8d\n", counter, array[ counter ] );
17    } // end main
18 } // end class InitArray
```

Index	Value
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0

# Описание

- Ред 8 Декларира array като масив от `int` елементи
- Ред 10 Създава 10 `int` елементи за array; всеки `int` елемент е инициализиран с **0** по подразбиране
- Ред 15 `array.length` връща дължината на array
- Ред 16 `array[counter]` връща `int` стойността на index място в масива array

Index	Value
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0

Всеки **int** елемент на масива е инициализиран на **0** по подразбиране



# Използване на инициализиращ списък

- Списък от стойности, разделени със запетая и ограден с фигурни скоби({})

```
int n[] = { 10, 20, 30, 40, 50};
```

- Създава и инициализира масив от пет елемента
- Индексът на масива може да взима стойности 0, 1, 2, 3, 4
- **Няма нужда** от ключовата дума **new**

# Използване на инициализиращ списък

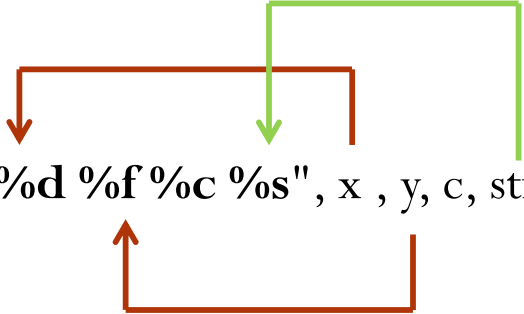
Компилаторът използва  
инициализиращ списък за  
да създаде array

Index	Value
0	32
1	27
2	64
3	18
4	95
5	14
6	90
7	70
8	60
9	37

```
3
4 public class InitArray
5 {
6     public static void main( String args[] )
7     {
8         // initializer list specifies the value for each element
9         int array[] = { 32, 27, 64, 18, 95, 14, 90, 70, 60, 37 };
10
11         System.out.printf( "%s%8s\n", "Index", "Value" ); // column headings
12
13         // output each array element's value
14         for ( int counter = 0; counter < array.length; counter++ )
15             System.out.printf( "%5d%8d\n", counter, array[ counter ] );
16     } // end main
17 } // end class InitArray
```

# Форматиране на изхода с **printf**

```
public class string_b {  
    public static void main(String args[]) {  
        int x = 20;  
        float y = 5.5f;  
        char c = 'J';  
        String str = "Hello Java";  
  
        //Displaying formatted string  
        System.out.printf("The formatted string: %d %f %c %s", x , y, c, str);  
    }  
}
```



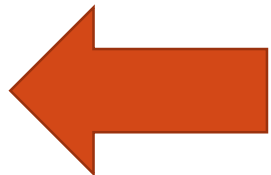
The diagram illustrates the mapping of variables to format specifiers in the printf statement. It features two colored lines: a red line and a green line. The red line starts from the variable 'x' and points to the '%d' specifier. The green line starts from the variable 'y' and points to the '%f' specifier. Another red line starts from the variable 'c' and points to the '%c' specifier. A final green line starts from the variable 'str' and points to the '%s' specifier. These lines are arranged in a stepped fashion, connecting each variable to its corresponding format specifier in the string.

**The formatted string: 20 5.500000 J Hello Java**

# Esc sequences

- **%d** – for signed **decimal integer**
- **%f** – for the **floating point**
- **%o** – octal number
- **%c** – for a **character**
- **%s** – a **string**
- **%i** – use for integer base 10
- **%u** – for unsigned decimal number
- **%x** – hexadecimal number
- **%%** – for writing % (percentage)
- **%n** – **for new line = \n**
- <https://www.jquery-az.com/10-examples-learn-java-string-formatting-printf-method/>

# Пресмятане на израз и съхраняване на резултата в елементите на масив

- Пример:
- Инициализиране на елементите на масив с четни числа по формулата  
$$2 + 2 * \text{index}$$

- Табулиране на стойностите на елементите на масива при печат

# Правила за добро програмиране

- Използване на константи:
- **Константите се наричат още `read-only` променливи.**
- Такива променливи **правят програмите по-читаеми** отколкото използването на конкретни стойности.
- Една `read-only` променливатакато **`ARRAY_LENGTH`** ясно определя целите за използването ѝ.


# Обичайна грешка при програмиране

- **Повторно присвояване** на стойност на **read-only** променлива **води до грешка при компилация.**



# Пресмятане на израз и съхраняване на резултата в елементите на масив

```
1 // Fig. 7.4: InitArray.java
2 // Calculating values to be placed into elements of an array.
3
4 public class InitArray
5 {
6     public static void main( String args[] )
7     {
8         final int ARRAY_LENGTH = 10; // declare constant
9         int array[] = new int[ ARRAY_LENGTH ]; // create array
10
11         // calculate value for each array element
12         for ( int counter = 0; counter < array.length; counter++ )
13             array[ counter ] = 2 + 2 * counter;
14
15         System.out.printf( "%s%8s\n", "Index", "Value" ); // column headings
16
17         // output each array element's value
18         for ( int counter = 0; counter < array.length; counter++ )
19             System.out.printf( "%5d%8d\n", counter, array[ counter ] );
20     } // end main
21 } // end class InitArray
```



Index	Value
0	2
1	4
2	6
3	8
4	10
5	12
6	14
7	16
8	18
9	20



# Сумиране на елементите на масив

```
1  // Fig. 7.5: SumArray.java
2  // Computing the sum of the elements of an array.
3
4  public class SumArray
5  {
6      public static void main( String args[] )
7      {
8          int array[] = { 87, 68, 94, 100, 83, 78, 85, 91, 76, 87 };
9          int total = 0;
10
11             // add each element's value to total
12             for ( int counter = 0; counter < array.length; counter++ )
13                 total += array[ counter ];
14
15             System.out.printf( "Total of array elements: %d\n", total );
16         } // end main
17     } // end class SumArray
```

```
Total of array elements: 849
```

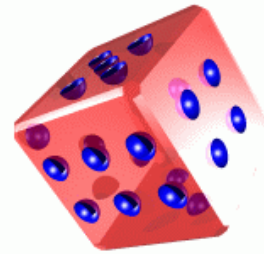
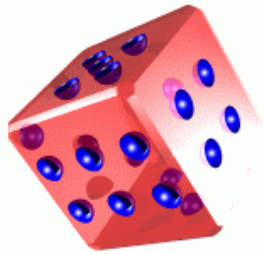
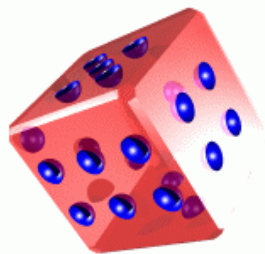
## Използване на елементите на масив като броячи (акумулиращи стойност на променливи)

- Нека в масив да се запишат показанията на зарче от 6000 хвърляния; т.е честотата, с която се падат числата 1-6 при хвърляне на зарче.
- Така се елиминира нуждата от използване на switch команда!



# Генериране на случайни числа

`1 + randomNumbers.nextInt( 6 )`



```

3  import java.util.Random;
4
5  public class RollDie
6  {
7      public static void main( String args[] )
8      {
9          Random randomNumbers = new Random(); // random number generator
10         int frequency[] = new int[ 7 ]; // array of frequency counters
11
12         // roll die 6000 times; use die value as frequency index
13         for ( int roll = 1; roll <= 6000; roll++ )
14             ++frequency[ 1 + randomNumbers.nextInt( 6 ) ];
15
16         System.out.printf( "%s%10s\n", "Face", "Frequency" );
17
18         // output each array element's value
19         for ( int face = 1; face < frequency.length; face++ )
20             System.out.printf( "%4d%10d\n", face, frequency[ face ] );
21     } // end main
22 } // end class RollDie

```

Face	Frequency
1	988
2	963
3	1018
4	1041
5	978
6	1012

**int frequency[]**

frequency[0]	frequency[1]	frequency[2]	frequency[3]	frequency[4]	frequency[5]	frequency[6]
	988	963	1018	1041	978	1012

# Задача:

- Създайте масив с максимален брой елементи равен на 20.
- Осигурете възможност да се дефинират броя на елементите на масива и да се инициализират от клавиатурата.
- Нека програмата да е с валидиран вход – да не е възможно броят на елементите да е отрицателен и по-голям от 20.
- Създайте меню, което да позволява извеждане на
  - Сумата на елементите на масива;
  - Произведението на елементите на масива;
  - Броя на положителните елементи на масива;
  - Изход

```
package com.company;

import java.util.Scanner;

public class Array1 {

    public static void main ( String[] args ) {

        int[] array = new int[ 20 ];
        int sum = 0;
        int n, k;
        int p = 1;
        Scanner as = new Scanner ( System.in );
        do {
            System.out.println ( "Въведи брой на елементи" );
            n = as.nextInt ( );
        } while ( ( n <= 0 ) || ( n > 20 ) );
        for ( int i = 0 ; i < n ; i++ ) {
            System.out.println ( "елемент array[" + i + "]= " );
            array[ i ] = as.nextInt ( );
        }
    }
}
```

```
do {  
    System.out.println ( "Избери опцията менюто:" );  
    System.out.println ( "1. Сума на елементите" );  
    System.out.println ( "2. Произведение на елементите" );  
    System.out.println ( "3. Брой на положителните елементи" );  
    System.out.println ( "4. Изход" );  
    System.out.print ( "Моля избери точка от менюто:" );  
    k = as.nextInt ( );  
    switch ( k ) {  
        case 1:  
            for ( int i = 0 ; i < n ; i++ ) {  
                sum += array[ i ];  
            }  
            System.out.println ( sum );  
            break;  
        case 2:  
            for ( int i = 0 ; i < n ; i++ ) {  
                p *= array[ i ];  
            }  
            System.out.println ( p );  
            break;  
    }  
}
```

```
        case 3:
            for ( int i = 0 ; i < n ; i++ ) {
                if ( array[ i ] > 0 )
                    k++;
            }
            System.out.println ( k );
            break;
        case 4:
            break;
        default:
            System.out.println ( "Няма такава опция" );
    }
} while ( k != 4 );
}
```



# Примери за използване на масиви

- Използване на стълбови диаграми за визуализиране на масив от данни
- Дължината на стълбовете определя големината на стойностите в зададен мащаб (пропорционално)
  - Пример: Разпределение на брой оценки
  - Всеки елемент на масива array задава брой оценки в интервал от 10:  
[0, 9], [10, 19], ..., [80, 89], [90-99], [100]

```

~
4 public class BarChart
5 {
6     public static void main( String args[] )
7     {
8         int array[] = { 0, 0, 0, 0, 0, 0, 1, 2, 4, 2, 1 };
9
10        System.out.println( "Grade distribution:" );
11
12        // for each array element, output a bar of the chart
13        for ( int counter = 0; counter < array.length; counter++ )
14        {
15            // output bar label ( "00-09: ", ..., "90-99: ", "100: " )
16            if ( counter == 10 )
17                System.out.printf( "%5d: ", 100 );
18            else
19                System.out.printf( "%02d-%02d: ",
20                                   counter * 10, counter * 10 + 9 );
21
22            // print bar of asterisks
23            for ( int stars = 0; stars < array[ counter ]; stars++ )
24                System.out.print( "*" );
25
26            System.out.println(); // start a new line of output
27        } // end outer for
28    } // end main
29 } // end class BarChart

```

```

Grade distribution:
00-09:
10-19:
20-29:
30-39:
40-49:
50-59:
60-69: *
70-79: **
80-89: ****
90-99: **
100: *

```

```
Grade distribution:
00-09:
10-19:
20-29:
30-39:
40-49:
50-59:
60-69: *
70-79: **
80-89: ****
90-99: **
    100: *
```

- Ред 8  
Деклариране на array с инициализиращ списък
- Ред 19  
Използване на флаг 0 за извеждане на водещи нули пред едноцифрени стойности
- Редове 23-24  
За всеки елемент на array, print на съответен брой '\*'

# Използване на масиви за изследване на статистически извадки

- **40 студента оценяват качеството на храната**
- **Въведена е скала 1-10:  
1 означава много лоша,  
10 означава отлична**
- **Получени са 40 анкети и резултатите са  
въведени в масив от цели елементи**
- **Обобщаване на резултатите в масив с 10  
елемента.**

Declare responses as array to store 40 responses

```
~
4 public class StudentPoll
5 {
6     public static void main( String args[] )
7     {
8         // array of survey responses
9         int responses[] = { 1, 2, 6, 4, 8, 5, 9, 7, 8, 10, 1, 6, 3, 8, 6,
10             10, 3, 8, 2, 7, 6, 5, 7, 6, 8, 6, 7, 5, 6, 6, 5, 6, 7, 5, 6,
11             4, 8, 6, 8, 10 };
12         int frequency[] = new int[ 11 ]; // array of frequency counters
13
14         // for each answer, select responses element and use that value
15         // as frequency index to determine element to increment
16         for ( int answer = 0; answer < responses.length; answer++ )
17             ++frequency[ responses[ answer ] ];
18
19         System.out.printf( "%s%10s", "Rating", "Frequency" );
20
21         // output each array element's value
22         for ( int rating = 1; rating < frequency.length; rating++ )
23             System.out.printf( "%d%10d", rating, frequency[ rating ] );
24     } // end main
25 } // end class StudentPoll
```

For each response, increment frequency values at index associated with that response

Declare frequency as array of 11 int and ignore the first element

Rating	Frequency
--------	-----------

1	2
---	---

2	2
---	---

3	2
---	---

4	2
---	---

5	5
---	---

6	11
---	----

7	5
---	---

8	7
---	---

9	1
---	---

10	3
----	---

## Задача:

- Да се напише програма, която да пресмята по колко ученици от три класа, получили съответно общи първа, втора и трета награда за участие в състезание по програмиране, могат да отидат на екскурзия.
- Размерът на всяка обща награда и цената на екскурзията за един ученик (в лева и стотинки) са известни и се въвеждат от клавиатурата.

```
1 package com.company;
2 import java.util.Scanner;
3 /**
4  *
5  * Да се напише програма, която да пресмята по колко ученици от три класа,
6  * получили съответно общи първа, втора и трета награда за участие в
7  * състезание по програмиране, могат да отидат на екскурзия. Размерът на
8  * всяка обща награда и цената на екскурзията за един ученик (в лева и стотинки)
9  * са известни и се въвеждат от клавиатурата.
10 */
11 public class Main {
12
13     public static void main(String[] args) {
14         Scanner kbd;
15         int[] studentsOnExcursion; //array of 3 elements
16         double[] prizeValues; // array of 3 elements
17         double priceOfExcursion;
18         // initializaiton
19         kbd= new Scanner(System.in);
20         studentsOnExcursion = new int[3];
21         prizeValues = new double[3];
22
23         System.out.print("Enter price of excursion: ");
24         priceOfExcursion = kbd.nextDouble();
25     }
```



```
25
26 for (int i = 0; i < prizeValues.length; i++) {
27     System.out.printf( s: "Enter amount of Prize No. %d: ", ...objects: i+1);
28     prizeValues[i] = kbd.nextDouble();
29 }
30
31 // processing
32 for (int i = 0; i < prizeValues.length; i++) {
33     studentsOnExcursion[i] = (int) (prizeValues[i] /priceOfExcursion);
34 }
35
36 // output
37 System.out.println("Students on excursion");
38 System.out.printf( s: "%20s%30s%n", ...objects: "Prize", "Total students on excursion");
39 for (int i = 0; i < studentsOnExcursion.length; i++) {
40     System.out.printf( s: "%20d%30d%n", ...objects: i+ 1,studentsOnExcursion[i]);
41 }
42 }
43
44 }
45
```

# Преговор

```

public class TestArray {

    public static void main(String[] args) {
        double[] myList = {1.9, 2.9, 3.4, 3.5};

        // Print all the array elements
        for (int i = 0; i < myList.length; i++) {
            System.out.println(myList[i] + " ");
        }

        // Summing all elements
        double total = 0;
        for (int i = 0; i < myList.length; i++) {
            total += myList[i];
        }
        System.out.println("Total is " + total);

        // Finding the largest element
        double max = myList[0];
        for (int i = 1; i < myList.length; i++) {
            if (myList[i] > max) max = myList[i];
        }
        System.out.println("Max is " + max);
    }
}

```

1.9

2.9

3.4

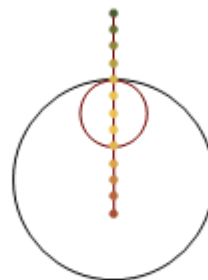
3.5

Total is 11.7

Max is 3.5

# Задача - Обръщане на елементите на масив

- Да се напише програма, която позволява да се въведе масив, обръща го (така че първият елемент да стане последен, втория – предпоследен и т.н.) и отпечатва така получения масив.



# Задача - Обръщане на елементите на масив

- Алгоритъм:

1. Деклариране и инициализация на масив от KB

```
int n, arr[30], swop, p;
```

```
.  
for(int i=0 ; i<n ; i++){  
    ....,Въведете стойностите в масива: ,  
.  
{
```

2. Смяна на два елемента

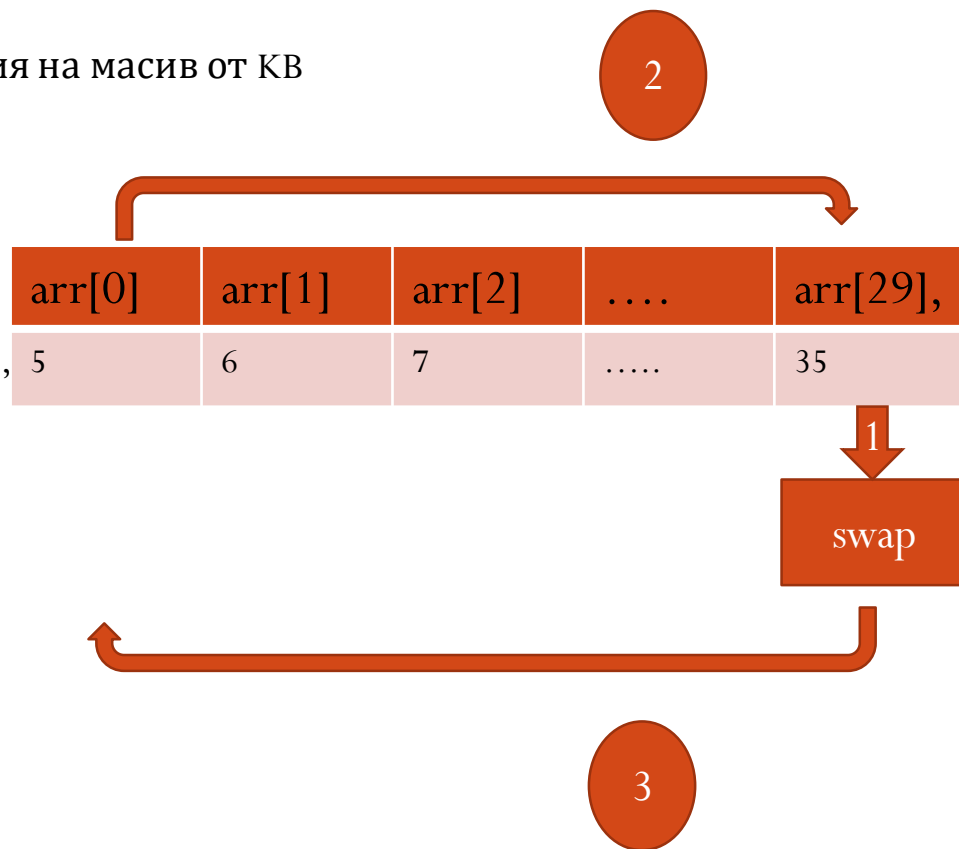
```
for (int j=n-1 , i=0; j>i ; j-- , i++){
```

```
    swop=arr[j];
```

```
    arr[j]=arr[i];
```

```
    arr[i]=swop;
```

```
{
```

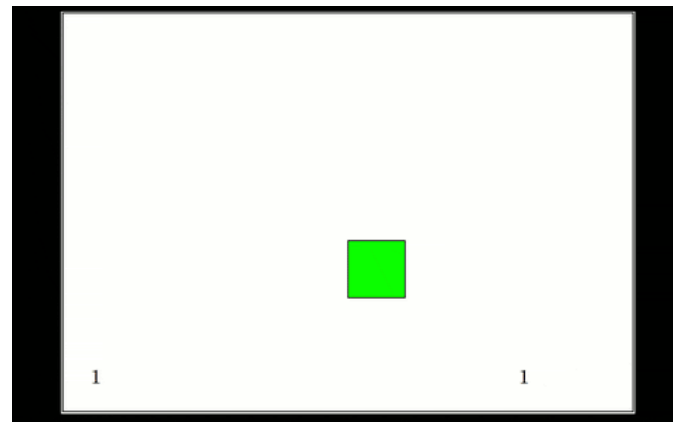


# Задача

- Да се напише програма, която по зададено естествено число  $n$  отпечатва цифрите в десетичния му запис в обратен ред.

## ПРИМЕР:

ВХОД        1024  
РЕЗУЛТАТ   4201



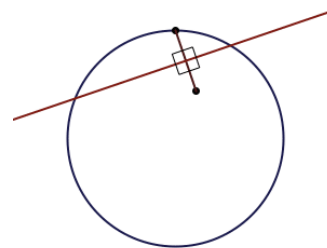
# Примерно решение

Old number is: 235467;

Expected number is: 764532;

Process completed.

```
public class Zad8{  
    public static void main(String [] args){  
        long number=235467;  
        long k=number;  
        long result=0;  
        while(k!=0){  
            int p=(int)k%10;  
            result=result*10+p;  
            k=k/10;  
        }  
        System.out.println("Old number is:      "+number+";\nExpected number is: "+result+";");  
    }  
}
```



```

public class Zad8{
    public static void main(String [] args){
        long number=235467;
        long k=number;
        long result=0;
        while(k!=0){
            int p=(int)k%10;
            result=result*10+p;
            k=k/10;
        }
        System.out.println("Old number is:      "+number+"\nExpected number is: "+result+"");
    }
}

```

#### General Output

```

-----Configuration: <Default>-----
Old number is:      235467;
Expected number is: 764532;

Process completed.

```



# Домашно

- Задача „Щастлива година“
- Създайте програма HAPPY, която чете от клавиатурата последователност от години, до въвеждане на нула.
- Съхранява годините в масив.
- Извежда на екрана броя на тези от тях, които нямат еднакви цифри в записа си, както и самите години.
- Забележка: Всички зададени във входа години да са четирицифрени

# Домашно

- Всяка редица от равни числа в едномерен сортиран масив, се нарича площадка.
- Да се напише програма, която намира началото и дължината на всяка площадка в даден сортиран във възходящ ред едномерен масив `arr_sort[10]`, а ако няма площадка да извежда съответното съобщение.