



Увод в програмирането с Java



Масиви (I част)

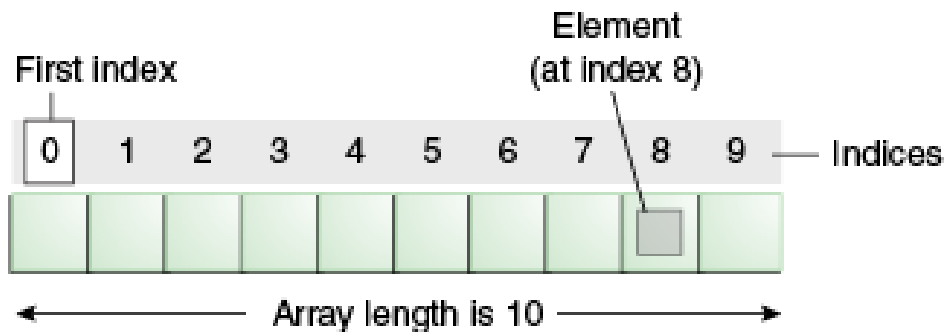
Съдържание

- Деклариране
- Достъп до елементите
- Обхождане
- Копиране
- Домашно

ДЕКЛАРИРАНЕ

Масив

Обект, който съдържа в себе си определен брой други обекти от един и същ тип.



Деклариране на масив

От всеки един тип може да се направи масив. Той се използва, когато имаме много променливи от един и същ тип и ни се налага да ги обхождаме машинно.

```
int[] numbers = {1, 2, 3, 4}; //масив от int
```

```
boolean[] isRaining = {true, true, false, false};
```

Други начини за създаване

Когато няма да задаваме стойността на елементите в началото, но знаем техния брой:

```
int[] bones = new int[206];
```

//сега създадохме масив от int с 206 елемента, който е празен - елементите му нямат стойност.

Достъп до елементите

Достъп до елементите

Достъпът до елементите става с квадратни скоби:

```
String[] bones = new String[206];
```

```
bones[0] = "ulna";
```

```
//първия елемент на масива има стойност, всички останали са празни
```

Накратко

Масивът:

- Е поредица от елементи от един и същи тип
- Има фиксиран размер (брой елементи)
- Поддържа елементите по фиксиран начин

Деклариране

Има два начина:

- Когато знаем кои са елементите:

– `int[] myArray = {1, 2, 3, 4, 5};`

- Когато знаем само броя им:

– `int[] myArray = new int[5];`

– Операторът “new” заделя памет за масива.

Достъпване

- Могат да се достъпват чрез []
- Индексът на първия елемент е 0
- Индексът на последния елемент е дължината на масива - 1

Задача

Имате масива

```
int[] numbers = {1, 2, 3, 4, 5};
```

Обърнете елементите му, така че новия масив да е
5, 4, 3, 2, 1

Обхождане

Обхождане

Често ни се налага да минем през всички елементи на масив и да направим нещо с тях. Този процес се казва обхождане.

Възможен е чрез цикли.

```
int[] array = {3 ,5 ,6, 6,7,8};
```

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

```
    array[i]=5+i;
```

```
}
```

Въпрос

- Имате масив с 10 елемента, искате да добавите 11ти, възможно ли е?

Задача

Даден ви е масивът `int[] array = {43, 56, 4, 3, 6, 8, 43, 5, 7, 87, 4, 3, 5, 6}`.

Намерете най - големият елемент в него и го изведете.

Програмата ви трябва да работи за всички масиви от тип `int`.

Задачи

Задача

1. Да се напише програма, която създава масив с 20 елемента от целочислен тип и инициализира всеки от елементите със стойност равна на индекса на елемента умножен по 5. Елементите на масива да се изведат на конзолата.

Задача

2. Да се напише програма, която чете два масива от конзолата и проверява дали са еднакви.

Задача

3. Даден ви е масива `int[] array`.

Изведете на конзолата `true`, ако в масива има число 6 и то е на първа или последна позиция. В противен случай изведете `false`.

Домашно

Задача

Прочетете 7 глава от книгата: „**Масиви**”

<http://www.introprogramming.info/intro-java-book/read-online/glava7-masivi/>

Задача

1. Дадени са ви два масива - `int[] arr1` и `int[] arr2`.
Изведете на конзолата `true` ако двата масива имат един и същ първи или последен елемент

Задача

2. Дадени са ви два масива - `int[] arr1` и `int[] arr2`.

Масивите са с еднаква дължина.

Създайте масив, който на четни позиции има елементите на четни позиции от `arr1`, а на нечетни има елементите на нечетни позиции от `arr2`.

Пример:

```
int[] arr1 = {3,5,8,9,4,5};
```

```
int[] arr2 = {11,22,33,44,55,66};
```

```
→ arr3: {11,5,33,9,55,5}
```

Задача

3. Даден ви е масива `int[] array`. Създайте нов масив, с елементите от първия, но без числото 6.

Пример:

`int[] array = {2,4,5,6,8,6,3,4,6}`

→ `array2: {2,4,5,8,3,4};`

Задача

4. Потребителя въвежда текст на английски. Изведете на конзолата същия текст, но без буквите 'A' и 'a'.

Пример:

вход: Alligators are very dangerous animals.

изход: lligtors re very dngorous nimls.

Подсказка: променливите от тип `String` могат да бъдат превръщани в масиви от тип `char[]`.

```
String text = "hello";
```

```
char[] letters = text.toCharArray();
```

Задача

Гледайте това видео:

https://youtu.be/TRovth_4MsM

Гледайте видеото до частта с List тя не ни интересува за сега, ще я учите след празниците

- Урока е за C#, но на Java е същото с изключение на декларирането на типа, вместо var в Java се използва конкретен тип.