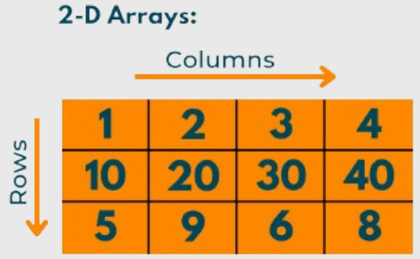
VIENDIMENSIJU MASĪVI PROGRAMMĒŠANAS VALODĀ JAVA

Gļebs Čečotkins 2PT

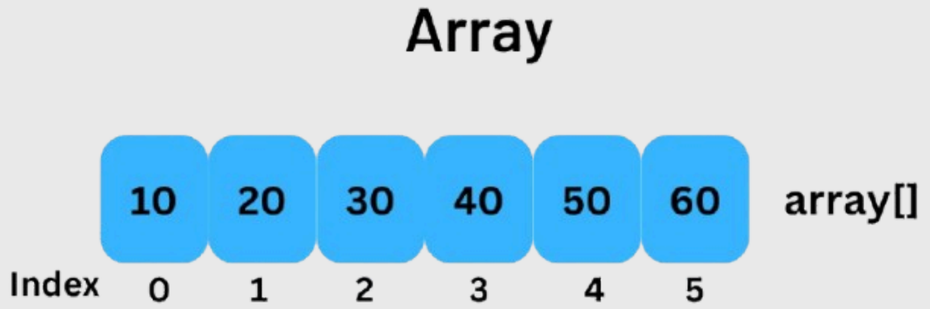
KAS TAS IR?

Masīvs ir datu struktūra, kurā tiek glabāti viena tipa elementi. To var uzskatīt par numurētu šūnu kopu, no kurām katra var saturēt noteiktus datus (vienu datu elementu katrā šūnā). Piekļuve konkrētai šūnai tiek veikta, izmantojot tās numuru.



VIENDIMENSIJU MASĪVS

Viendimensiju masīvs ir lineāra viena tipa elementu secība, kurai piekļūst, izmantojot indeksus sakot no 0:



VIENDIMENSIJU MASĪVU PIELIETOJUMI

- Sarakstus un kolekcijas
- Datu glabāšanu un izgūšanu
- Steks un rindas
- Matricas un vektorus
- Dinamisko programmēšanu
- Kārtošanas un meklēšanas algoritmus
- Grafu algoritmus
- Histogrammas un frekvenču skaitīšanu
- Attēlu apstrādi
- Kriptogrāfija

PRIEKŠROCĪBAS

- Vienkārša piekļuve: Masīva elementi ir viegli pieejami, izmantojot indeksu, kas ļauj ātri un efektīvi izgūt un modificēt datus.
- Vienkāršota atmiņas pārvaldība: Atšķirībā no dinamiskām datu struktūrām, piemēram, sarakstiem un kopām, masīvi aizņem nepārtrauktu atmiņas bloku, kas palielina elementu piekļuves ātrumu un vienkāršo atmiņas pārvaldību.
- Atbalsts standarta bibliotēkām: Java nodrošina daudzas iebūvētas metodes darbam ar masīviem (kārtošana, meklēšana, kopēšana utt.).
- Koda optimizēšana: kods kļūst optimizētāks, īsāks un vieglāk ir atlasīt un kārtot datus.

TRŪKUMI

- Fiksēts izmērs – pēc izveides garumu nevar mainīt
- Nav iebūvaētu darba metožu (atšķirībā no kolekcijām)
- Neērtības ievietot/noņemt elementus

KĀ DEKLARĒT UN INICIALIZĒT MASĪVU

- Deklarācija ar garumu:

Datu tips nosaukums izmers

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

- Deklarācija ar inicializētām vērtībām:

Datu tips nosaukums Piešķirtās vērtības

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

MASĪVU DATU TIPI

- Masīvu datu tipu piemēri:

```
int[] numbers;
String[] strings;
double[] values;
char[] characters;
```

- Masīvu datu tipu piemēri:

```
int[] numbers = {1, 2, 3, 4, 5};
String[] strings = {"Gleb", "Maksim", "Leonid"};
double[] values = {3.14, 2.71, 1.41};
char[] characters = {'A', 'B', 'C'};
```

KĀ ATTĒLOT MASĪVU JAVA VALODĀ?

- Masīva elementus var attēlot ekrānā (tas ir, konsolē), izmantojot,for ciklu.

```
int[] numbers = {1, 2, 3, 4, 5};
for (int i = 0; i < numbers.length; i++) {
    System.out.println(numbers[i]);
}
```

- Rezultātā programma izvadīs šādu rezultātu:

```
1
2
3
4
5
```

ĪSĀKS VEIDS, KĀ IZDRUKĀT MASĪVU

- Masīva elementus var attēlot ekrānā , izmantojot, for-each ciklu.

```
int[] numbers = {1, 2, 3, 4, 5};
for (int num : numbers) {
    System.out.println(num);
}
```

- Rezultātā programma izvadīs šādu rezultātu:

```
1
2
3
4
5
```

KĀ ATTĒLOT MASĪVU APGRIEZTĀ SECĪBĀ JAVA VALODĀ?

- Masīva elementus ekrānā var attēlot apgrieztā secībā, izmantojot ciklu for, sākot no pēdējā indeksa un virzoties uz nulli.

```
int[] numbers = {10, 20, 30, 40, 50};
System.out.println("apgrieztā secībā");
for (int i = numbers.length - 1; i >= 0; i--) {
    System.out.print(numbers[i] + " ");
}
```

- Rezultātā programma izvadīs šādu rezultātu:

```
apgrieztā secībā
50 40 30 20 10
```

ELEMENTU IEVIETOŠANA MASĪVĀ AR SCANNER

- Ar Scanner klases palīdzību ir iespējams masīva elementus ievadīt lietotājam no konsoles. Šādā situācijā, deklarējot masīvu nepieciešams norādīt atslēgvārdu new un figūriekavās masīva elementu skaitu.
- Rezultātā programma izvadīs šādu rezultātu:

```
Ievadi cparu ar kārtas indeksu 0
1
Ievadi cparu ar kārtas indeksu 1
2
Ievadi cparu ar kārtas indeksu 2
3
Ievadi cparu ar kārtas indeksu 3
4
Ievadi cparu ar kārtas indeksu 4
5
1
2
3
4
5
```

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

    int[] numbers = new int[5];
    Scanner scan = new Scanner(System.in);
    for(int i=0; i<numbers.length;i++) {
        System.out.println("Ievadi cparu ar kārtas indeksu "+i);
        numbers[i] = scan.nextInt();
    }
    scan.close();
    for(int num: numbers) {
        System.out.println(num);
    }

}
```

ELEMENTU IEVIETOŠANA MASĪVĀ AR RANDOM

- Izmantojot Random klases palīdzību, masīvs tiek aizpildīts ar nejaušiem skaitļiem, manā gadījumā no 1 līdz 5.

```
int[] numbers = new int[5];
Random rand = new Random();
for(int i=0; i<numbers.length;i++) {
    numbers[i] = rand.nextInt(5)+1;
}

for(int num: numbers) {
    System.out.println(num);
}
```

- Rezultātā programma izvadīs šādu rezultātu:

```
4
3
3
2
4
```

MASĪVA KĀRTOŠANA

- Metode Arrays.sort ļauj kārtot masīva elementus augošā secībā:

```java
import java.util.Arrays;

public class uzd1 {
    public static void main(String[ ] args) {
        int[] numbers = {1, 3, 1, 5, 4};
        Arrays.sort(numbers);
        System.out.println(Arrays.toString(numbers));
    }
}
```

- Rezultātā programma izvadīs šādu rezultātu:

```
[1, 1, 3, 4, 5]
```

MEKLĒT ELEMENTU MASĪVĀ

- Arrays.binarySearch metode tiek izmantota, lai meklētu elementu sakārtotā masīvā. Ja elements tiek atrasts, tiek atgriezts tā indekss, pretējā gadījumā negatīva vērtība:

```java
import java.util.Arrays;

public class uzd1 {
    public static void main(String[ ] args) {
        int[] numbers = {9, 3, 5, 6, 7};
        int index = Arrays.binarySearch(numbers, 5);
        System.out.println("Index 5: " + index);
    }
}
```

- Rezultātā programma izvadīs šādu rezultātu:

```
Index 5: 2
```

MASĪVA KOPĒŠANA

- Metode Arrays.copyOf ļauj izveidot masīva kopiju ar noteiktu garumu:

```java
import java.util.Arrays;

public class uzd1 {
    public static void main(String[ ] args) {
        int[] original = {1, 2, 3, 4, 5};
        int[] copy = Arrays.copyOf(original, 3);
        System.out.println(Arrays.toString(copy));
    }
}
```

- Rezultātā programma izvadīs šādu rezultātu:

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

MASĪVA KONVERTĒŠANA VIRKNĒ

- Metode Arrays.toString konvertē masīvu virknē, kas ir ērti izvadei un atkļūdošanai:

```java
import java.util.Arrays;

public class uzd1 {
    public static void main(String[ ] args) {
        int[] numbers = {1, 2, 3, 4, 5};
        String arrayString = Arrays.toString(numbers);
        System.out.println(arrayString);
    }
}
```

- Rezultātā programma izvadīs šādu rezultātu:

```
[1, 2, 3, 4, 5]
```

NOSACĪJUMI UN FILTRĒŠANA MASĪVOS

- Filtrēšana ir process, kurā no masīva tiek atlasīti tikai tie elementi, kas atbilst noteiktiem nosacījumiem.
- piemērs 1: Izdrukāt no masīva tikai pozitīvus skaitļus

```
int[] numbers = {-5, 0, 12, -3, 7, 8};
for (int num : numbers) {
    if (num > 0) {
        System.out.println(num);
    }
}
```

- Rezultātā programma izvadīs šādu rezultātu:

```
12
7
8
```

- Ko var pārbaudīt, izmantojot nosacījumus:
  - num > 0 – pozitīvi skaitļ
  - num % 2 == 0 – pāra skaitlis
  - num < 100 – mazāks par noteiktu vērtību
  - num != 0 – izslēgt nulles
  - num == 5 – atrast noteiktu vērtību

KĻŪDAS, STRĀDĀJOT AR MASĪVIEM

- Java valodā masīviem ir fiksēts garums, un tas ir pieejams, izmantojot garuma īpašību.
- Piemēram:

```
int[] numbers = new int[5];
System.out.println(numbers[6]);
```

- Rezultātā programma izvadīs šādu rezultātu:

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 6 out of bounds for length 5
        at uzd1.main(uzd1.java:6)
```

- Atkārtojot masīva apstrādi, vienmēr izmantojiet īpašību length, lai izvairītos no ArrayIndexOutOfBoundsException:

1.UZDEVUMS

Izveidojiet masīvu no 5 veseliem skaitļiem, lietotājs ievada vērtības un izdrukā visus masīva elementus, izmantojot for ciklu.

1.UZDEVUMS

```java
import java.util.Scanner;

public class uzd1 {
    public static void main(String[ ] args) {
        int[] numbers = new int[5];
        Scanner scan = new Scanner(System.in);

        //Ievade
        for(int i=0; i<numbers.length;i++) {
            System.out.println("Ievadiet " + (i + 1) + ". skaitli: ");
            numbers[i] = scan.nextInt();
        }
        scan.close();

        //Izvade
        System.out.println("Jūsu ievadītie skaitļi:");
        for(int num : numbers) {
            System.out.print(num+" ");
        }
    }
}
```

2.UZDEVUMS

Lietotājs ievada 7 skaitļus. Saglabājiet tos masīvā un pēc tam parādiet visu masīva elementu summu un aritmētisko vidējo vērtību.

2.UZDEVUMS

```java
import java.util.Scanner;

public class uzd2 {
    public static void main(String[] args) {
        int[] numbers = new int[7];
        int sum = 0;
        Scanner scanner = new Scanner(System.in);
       //Ievade
        for (int i = 0; i < numbers.length; i++) {
            System.out.print("Ievadiet skaitli " + (i + 1) + ": ");
            numbers[i] = scanner.nextInt();
            sum += numbers[i];
        }

        // Aprēķina aritmētisko vidējo
        double average = (double) sum / numbers.length;

     // Izvada rezultātus
        System.out.println("Summa: " + sum);
        System.out.println("Viedejais artmētiskais: " + average);

    }
}
```

3.UZDEVUMS

Izveidojiet masīvu no 10 nejaušiem veseliem skaitļiem no 1 līdz 100. Atrodiet un izdrukājiet šī masīva maksimālo un minimālo vērtību un pēc tam izdrukājiet visus masīva elementus apgrieztā secībā.

3.UZDEVUMS

```java
import java.util.Random;

public class uzd3 {
    public static void main(String[] args) {
        int[] arr = new int[10];
        Random rand = new Random();

        // Aizpilda masīvu ar nejaušiem skaitļiem no 1 līdz 100
        for (int i = 0; i < arr.length; i++) {
            arr[i] = rand.nextInt(100) + 1;
        }

        int max = arr[0];
        int min = arr[0];

        // Atrod max un min
        for (int i = 1; i < arr.length; i++) {
            if (arr[i] > max) {
                max = arr[i];
            }
            if (arr[i] < min) {
                min = arr[i];
            }
        }

        // Izvada Max un Min
        System.out.println("\nMax: " + max);
        System.out.println("Min: " + min);

        // Izvada masīva elementus apgrieztā secībā
        System.out.println("Masīvs apgrieztā secībā:");
        for (int i = arr.length - 1; i >= 0; i--) {
            System.out.print(arr[i] + " ");
        }
    }
}
```

AVOTI

ChatGPT - https://chatgpt.com/
Prezentācija - https://skolo.lv/
JavaRush -https://javarush.com
YandexPrakticum -https://practicum.yandex.ru
GeeksForGeeks - https://www.geeksforgeeks.org

THANK YOU