****

Latvijas Universitātes Vadības un uzņēmējdarbības mācību centrs

(LU VUMC)

**IEVADS JAVA PROGRAMMĒŠANĀ**

**NODARBĪBA #7**

Mārtiņš Ceske/Vjačeslavs Vasiļevskis

2021

ESF projekts Nr. 8.4.1.0/16/l/001  
‘’Nodarbināto personu profesionālās kompetences pilnveide”



# 1. mājas darbs

# Reversais masīvs

### **Noteikumi:**

* Uzrakstiet metodi, kas kā parametru saņem masīvu un kā rezultātu atgriež masīvu kur elementi ir sakārtoti reversā kārtībā kā dotajam masīvam

padotais masīvs - {1, 2, 3}  
 rezultāts - {3, 2, 1}

### **Soļi:**

* Izveidojam jaunu klasi ar nosaukumu ArrayUtils un pievienojam statisku metodi reverse(int arr), kura pagaidām atgriezīs tukšu masīvu

public class ArrayUtils {  
   
 public static int[] reverse(int[] array) {  
 return new int[]{};  
 }  
   
 }

* Izveidojam jaunu klasi ar nosaukumu ArrayUtilsTest un pievienojam “main” metodi. Šo klasi mēs izmantosim, lai pārbaudītu ArrayUtils klases metodes.

public class ArrayUtilsTest {  
   
 public static void main(String[] args) {  
   
 }  
   
 }

* Tā kā metode “reverse” ir statiska, tad mēs varam to izsaukt bez objekta veidošanas. Izsaucam to klases ArrayUtilsTest metodē “main” un rezultātu ievietojam mainīgajā “result”

int[] result = ArrayUtils.reverse(new int[]{1, 2, 3});

* Uzrakstam mainīgā “result” pārbaudi. Mēs zinām, ka pareizs rezultātam ir jābūt tā, ka pirmais elements ir 3, otrais - 2, trešais – 1

System.out.print("Testējam klases ArrayUtils metodi 'reverse', darbojas ..");  
 if (result[0] == 3 && result[1] == 2 && result[2] == 1) {  
 System.out.println("PAREIZI");  
 } else {  
 System.out.println("NEPAREIZI");  
 }

* Palaižot šo “ArrayUtilsTest.main” iegūsim kļūdas paziņojumu, jo mēs taču vēl neesam realizējuši metodi “reverse”
* Testējam klases ArrayUtils metodi 'reverse', iegūsim kļūdu. Tā tam jābūt, jo neesam vēl uzrakstījuši pašu "reverse" metodi

..Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 0  
 at com.examples.lesson3.ArrayUtilsTest.main(ArrayUtilsTest.java:10)

**Metodes "reverse" implementācija**

* Mēs zinām, ka rezultāts būs masīvs, kuram izmērs būs tāds pats kā padotajam. Tātad mēs varam rakstīt:

int[] result = new int[array.length];  
 return result;

* Lai sagabātu elementus reversā kārtībā, mēs varam to nolasīt no dotā masīva sākot ar pēdējo elemtu.

int[] result = new int[array.length];  
  
 for (int i = array.length - 1; i >= 0; i--) {

//šeit i ir vienāds ar masīva lielāko pēdējā elementa indeksu  
 //pie katras iterācijas skaitītāju samazinām par vienu   
 / un iterējam līdz nonākam līdz masīva sākumam  
// ...   
 }

return result;

* Lai saglabātu vērtības citā masīvā mums ir nepieciešams jauns indeks skaitītājs “j”, nodefinēsim to skaitīsim uz priekšu ar katru cikla interāciju

for (int i = array.length - 1, j = 0; i >= 0; i--, j++) {   
 // ...  
 }

* Un beigās pievienojam ierakstīšanu jaunā masīvā pēc “j” indeksa un nolasīšanu pēc “i” indeksa

for (int i = array.length - 1, j = 0; i >= 0; i--, j++) {  
 result[j] = array[i];  
 }

* Rezultātam pēc veiksmīga testa ir jābūt šādam:

Testējam klases ArrayUtils metodi 'reverse', darbojas ..PAREIZI

# 2. mājas darbs

# Palindroms

### 

### **Noteikumi:**

* Uzrakstīt programmu, kas nosaka vai ievadītā simblu virkne ir palindroms.

**Piemēram:**

Racecar  
 Level  
 Sum summus mus

### **Soļi:**

* Izveidojiet klasi, kura realizē publisku metodi “isPalindrome(String text)”. Metode kā parametru saņem pārbaudāmo simbolu virkni un rezultātā atgriež true, ja teksts ir palindroms citādi - false.
* Izveidojiet testa klasi, kurā pārbaudāt metodes rezultātu
* Pārbaudes loģiku realizējiet izmantojiet masīvu.