****

Latvijas Universitātes Vadības un uzņēmējdarbības mācību centrs

(LU VUMC)

**IEVADS JAVA PROGRAMMĒŠANĀ**

**NODARBĪBA #10**

Mārtiņš Ceske/Vjačeslavs Vasiļevskis

2021

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



# 1. mājas darbs

# Lietotāja datu pārbaudītājs

### **Noteikumi:**

* Uzrakstīt klases UserValidator, kas pārbauda klases User lietotāja vārdu, uzvārdu un vecumu
* Atļauts veidot lietotājus, kuru vārda un uzvārda garums ir no 3 līdz 15 simboliem
* Atļauts ievadīt cilvēka vecumu no 0 līdz 120

P.S. Uzdevuma realizācijā ir jāizmanto pašu veidots Exception tips.

### **Soļi:**

* Izveidojam jaunu klasi User ar nepieciešamajiem laukiem un konstruktoru un geteriem

public class User {  
  
 private final String name;  
 private final String surname;  
 private final int age;  
   
 public User(String name, String surname, int age) {  
 this.name = name;  
 this.surname = surname;  
 this.age = age;  
 }  
  
 public String getName() {  
 return name;  
 }  
   
 public String getSurname() {  
 return surname;  
 }  
   
 public int getAge() {  
 return age;  
 }  
   
 }

* Izveidojam jaunu kļūdas tipu

public class ValidationException extends RuntimeException {  
   
 public ValidationException(String message) {  
 super(message);  
 }  
 }

* Izveidojam klasi UserValidator un pievienojam metodi validate

public class UserValidator {  
 public void validate(User user) {  
   
 }  
}

* UserValidator klasē pieliekam pārbaudi simbolu virknes garumam. Gadījumā, ja pārbaude izgāžas, tad metam kļūdu.

private static final int MIN\_TEXT\_LENGTH = 3;  
 private static final int MAX\_TEXT\_LENGTH = 15;  
 private static final int MIN\_AGE = 0;  
 private static final int MAX\_AGE = 120;  
  
 public void validate(User user) {  
 if (hasInvalidLength(user.getName()) || hasInvalidLength(user.getSurname())) {  
 throw new ValidationException("Name and surname has to be 3 to 15 characters long");  
 }  
 }  
   
 private boolean hasInvalidLength(String text) {  
 return text.length() < MIN\_TEXT\_LENGTH || text.length() > MAX\_TEXT\_LENGTH;  
 }

* Pievienojam arī vecuma pārbaudi

private static final int MIN\_AGE = 0;  
 private static final int MAX\_AGE = 120;  
  
 private void validate(String name, String surname, int age) {  
 // . . .  
 if (ageCheck(age)){  
 throw new ValidationException("Age must be within human possibility : 0 - 120");  
 }  
 }  
  
 private boolean hasInvalidAge(int age) {  
 return age < MIN\_AGE || age > MAX\_AGE;  
 }

* Uzrakstam jaunu klasi UserCreator ar metodi “main”. Izveidojam vairākus objektus, gan ar derīgām vērtībām, gan nederīgām un pārbaudām tās ar UserValidator. Kas sanāca?

public class UserCreator {  
   
 public static void main(String[] args) {  
 UserValidator validator = new UserValidator();  
 validator.validate(new User("Martins", "Ceske", 37));  
 // ...  
 }  
 }

# 2. mājas darbs

# Noķer kļūdu

### 

### **Noteikumi:**

* Izmantojot User un UserValidator klases no pirmā uzdevuma.
* Uzrakstīt programmu, kas ļauj ievadīt lietotājus no konsoles (izmnto Scanner klasi)
* Programmai ir jāpaprasa ievadīt visi nepieciešamie lietotāja dati un jāpārbauda to pareizība
* Kļūdas gadījumā programmai nedrīkst izbeigt savu darbību (izmanto try-catch), bet gan pieklājīgi ir jāizvada konsolē lietotājam paziņojums par kļūdu
* Kļūdainu datu ievades gadījumā programmai ir jāļauj lietotājam atkārtoti ievadīt datus

P.S. Izmanto pirmā uzdevuma risinājumu