Міністерство освіти і науки України

Національний технічний університет

«Харківський політехнічний інститут»

Кафедра «Обчислювальна техніка та програмування»

ЗВІТ

Про виконання лабораторної роботи 11

“Регулярні вирази. Перевірка даних”

Виконавець:

студент гр. КН-920В

Олексієнко Микита

Харків 2022

Лабораторна робота №11.

**Регулярні вирази. Перевірка даних**

**Мета**

Ознайомлення з принципами використання регулярних виразів для перевірки рядка на відповідність шаблону.

**1.Вимоги**

**1.1 Розробник:**

Олексієнко Микита Віталійович

студент групи КН-920В

1. варіант

**1.2 Загальне завдання**

Продемонструвати ефективне (оптимальне) використання регулярних виразів для перевірки коректності (валідації) даних, що вводяться, перед записом в domain-об'єкти відповідно до призначення кожного поля для заповнення розробленого контейнера:

* при зчитуванні даних з текстового файла в автоматичному режимі;
* при введенні даних користувачем в діалоговому режимі.

**1.3 Прикладні задачі**

Автомобіль: марка, рік випуску, технічні характеристики у вигляді "параметр - значення" (серед яких витрата палива в міському і заміському циклі), технічний стан, ціна.

**2. Опис програми**

**2.1 Засоби ООП:**

У лабороторній роботі 11 , я використовую клас Program, який має у собі лише запуск меню, розробленого у лабороторної роботі 8. Ця работа передбачає удосконалення власного контейнера, використанням регулярних виразів для забезпечення правильного заповнення domain-об’єктів.

Попередньо оголошено клас Auto, який має поля з індивідуального завдання та методи обробки ціх полей та клас Menu для роботи з користувачем.

**2.2 Ієрархія та структура класів:**

* Auto.java
* Container.java
* Iterator.java ( interface )
* LinkedList.java
* Menu.java
* Program.java

**2.3 Важливі фрагменти програми:**

Container.java ( додаткові методи ) :

public boolean loadFromFile(String path) throws IOException {  
 String result = null;  
  
 DataInputStream reader = new DataInputStream(new FileInputStream(path));  
 int nBytesToRead = reader.available();  
  
 if(nBytesToRead > 0) {  
 byte[] bytes = new byte[nBytesToRead];  
 reader.read(bytes);  
 result = new String(bytes);  
 }  
  
 for ( String line : result.split("\n") ) {  
 Auto ADDED\_CAR = new Auto();  
 int i = 0;  
 String[] words = line.split(", ");  
  
 Pattern MODEL\_NAME\_PATTERN = Pattern.compile("[A-ZА-Я]+");  
 Matcher MODEL\_NAME\_MATCHER = MODEL\_NAME\_PATTERN.matcher(words[i]);  
  
 if ( MODEL\_NAME\_MATCHER.matches() ) {  
 ADDED\_CAR.setModel( words[i++] );  
 } else return ERROR\_MESS();  
  
 Pattern YEAR\_PATTERN = Pattern.compile("\\d\\d\\d\\d");  
 Matcher YEAR\_MATCHER = YEAR\_PATTERN.matcher(words[i]);  
  
 if ( YEAR\_MATCHER.find() ) {  
 ADDED\_CAR.setReleaseYear( Integer.parseInt(words[i++]) );  
 } else return ERROR\_MESS();  
  
 Pattern URBAN\_PATTERN = Pattern.compile("\\d\\d\\d");  
 Matcher URBAN\_MATCHER = URBAN\_PATTERN.matcher(words[i]);  
  
 if ( URBAN\_MATCHER.find() ) {  
 ADDED\_CAR.setUrbanFuel( Integer.parseInt(words[i++]) );  
 } else return ERROR\_MESS();  
  
 Pattern SUBURBAN\_PATTERN = Pattern.compile("\\d\\d\\d");  
 Matcher SUBURBAN\_MATCHER = SUBURBAN\_PATTERN.matcher(words[i]);  
  
 if ( SUBURBAN\_MATCHER.find() ) {  
 ADDED\_CAR.setSubUrbanFuel( Integer.parseInt(words[i++]) );  
 } else return ERROR\_MESS();  
  
 ADDED\_CAR.setTechnicalCondition(Boolean.valueOf(words[i++]));  
  
 Pattern PRICE\_PATTERN = Pattern.compile("\\d\\d\\d\\d\\d");  
 Matcher PRICE\_MATCHER = PRICE\_PATTERN.matcher(words[i]);  
  
 if ( PRICE\_MATCHER.find() ) {  
 ADDED\_CAR.setPrice( Integer.parseInt(words[i++]) );  
 } else return ERROR\_MESS();  
  
 this.add(ADDED\_CAR);  
 }  
  
 System.out.println("loadFromFile - SUCCESS");  
 return true;  
}  
  
private static boolean ERROR\_MESS () {  
 System.err.println("Error. Incorrect input.");  
 return false;  
}

Menu.java ( перероблений метод CarAdding ) :

private static boolean carAdding () {  
 Auto ADDED\_CAR = new Auto();  
  
 Scanner scan = new Scanner(System.in);  
  
 System.out.println("Please type your car model:");  
 String model = scan.nextLine();  
  
 Pattern MODEL\_NAME\_PATTERN = Pattern.compile("[A-ZА-Я]+");  
 Matcher MODEL\_NAME\_MATCHER = MODEL\_NAME\_PATTERN.matcher(model);  
  
 if ( MODEL\_NAME\_MATCHER.find(1) ) {  
 ADDED\_CAR.setModel( model );  
 } else return ERROR\_MESS();  
  
 System.out.println("Please type your car release car:");  
 String year = scan.nextLine();  
  
 Pattern YEAR\_PATTERN = Pattern.compile("\\d\\d\\d\\d");  
 Matcher YEAR\_MATCHER = YEAR\_PATTERN.matcher(year);  
  
 if ( YEAR\_MATCHER.find() ) {  
 ADDED\_CAR.setReleaseYear( Integer.parseInt(year) );  
 } else return ERROR\_MESS();  
  
 System.out.println("Please type your car urban fuel:");  
 String urbanFuel = scan.nextLine();  
  
 Pattern URBAN\_PATTERN = Pattern.compile("\\d\\d\\d");  
 Matcher URBAN\_MATCHER = URBAN\_PATTERN.matcher(urbanFuel);  
  
 if ( URBAN\_MATCHER.find() ) {  
 ADDED\_CAR.setUrbanFuel( Integer.parseInt(urbanFuel) );  
 } else return ERROR\_MESS();  
  
 System.out.println("Please type your car suburban fuel:");  
 String suburbanFuel = scan.nextLine();  
  
 Pattern SUBURBAN\_PATTERN = Pattern.compile("\\d\\d\\d");  
 Matcher SUBURBAN\_MATCHER = SUBURBAN\_PATTERN.matcher(suburbanFuel);  
  
 if ( SUBURBAN\_MATCHER.find() ) {  
 ADDED\_CAR.setSubUrbanFuel( Integer.parseInt(suburbanFuel) );  
 } else return ERROR\_MESS();  
  
 System.out.println("Please choose your car state:");  
 System.out.println("[1] Good");  
 System.out.println("[2] Bad");  
  
 String technicalCondition = scan.nextLine();  
 ADDED\_CAR.setTechnicalCondition(Integer.parseInt(technicalCondition) == 1);  
  
 System.out.println("Please type your car price:");  
 String price = scan.nextLine();  
  
 Pattern PRICE\_PATTERN = Pattern.compile("\\d\\d\\d\\d\\d");  
 Matcher PRICE\_MATCHER = PRICE\_PATTERN.matcher(price);  
  
 if ( PRICE\_MATCHER.find() ) {  
 ADDED\_CAR.setPrice( Integer.parseInt(price) );  
 } else return ERROR\_MESS();  
  
 container.add(ADDED\_CAR);  
 return true;  
}

private static boolean ERROR\_MESS () {  
 System.err.println("Error. Incorrect input.");  
 return false;  
}

**3. Висновок:**

Ознайомлено з принципами використання регулярних виразів для перевірки рядка на відповідність шаблону.