Національний університет «Львівська Політехніка»

Кафедра Систем Штучного Інтелекту



Лабораторна робота №5

Виконав: студент КН-111

Зомчак Богдан

Перевірив:

Грабовська Н. Р.

**Львів-2019**

Тема:

Регулярні вирази. Перевірка даних. Обробка тексту.

Мета:

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

8. Автостанція. Запис в розкладі: номер рейсу; час відправлення; дні тижня; кількість вільних місць; маршрут - необмежений набір значень у вигляді “назва станції, час прибуття”.

import java.beans.XMLDecoder;

import java.beans.XMLEncoder;

import java.util.ArrayList;

import java.util.Locale;

import java.util.Scanner;

import java.io.\*;

import static java.lang.System.in;

public class Main {

public static int count = 0;

public static void main(String[] args) throws IOException {

Locale.setDefault(new Locale("uk", "UA"));

ArrayList<AutoStation> autostation = new ArrayList<>();

if (args.length == 1 && (args[0].equals("auto") || args[0].equals("-a"))) {

System.out.println("Auto mod enabled!");

try {

XMLDecoder decoder = new XMLDecoder(new FileInputStream("automod.xml"));

ArrayList<AutoStation> as2 = (ArrayList) decoder.readObject();

decoder.close();

for (int i = 0; i < as2.size(); i++) {

Output.printAutostation(0, as2);

System.out.println();

}

} catch (IOException ex) {

ex.printStackTrace();

}

}

Scanner scan = new Scanner(in);

while (true){

byte command;

int id;

Output.printMenu();

command = scan.nextByte();

switch (command) {

case 1:

autostation.add(new AutoStation());

count++;

System.out.println("Автостанція створена.");

System.out.println("Індекс: " + (count - 1));

break;

case 2:

if (!autostation.isEmpty()) {

System.out.println("Введіть індекс автостанції");

System.out.println("Можливі індекси: ");

amountOfId();

id = scan.nextInt();

scan.nextLine();

if (id < 0 || id > autostation.size() - 1) {

System.out.println("Id is impossible");

break;

} else {

System.out.print("Назва автостанції: ");

String toInput = scan.nextLine();

while (true) {

if (!Check.checkString(toInput)) {

System.out.println("Неправильно введено назву\nПовторіть спробу:");

toInput = scan.nextLine();

} else {

autostation.get(id).setNameOfStation(toInput);

break;

}

}

System.out.println("DD.MM.YYYY");

System.out.print("Дата: ");

toInput = scan.nextLine();

while (true) {

if (!Check.checkDate(toInput)) {

System.out.println("Неправильно введено дату\nПовторіть спробу:");

toInput = scan.nextLine();

} else {

autostation.get(id).setDate(toInput);

break;

}

}

System.out.print("Час відправлення: ");

toInput = scan.nextLine();

while (true) {

if (!Check.checkTime(toInput)) {

System.out.println("Неправильно введено час відправлення\nПовторіть спробу:");

toInput = scan.nextLine();

} else {

autostation.get(id).setTime(toInput);

break;

}

}

System.out.print("Час прибуття: ");

toInput = scan.nextLine();

while (true) {

if (!Check.checkTime(toInput)) {

System.out.println("Неправильно введено час прибуття\nПовторіть спробу:");

toInput = scan.nextLine();

} else {

autostation.get(id).setArrival(toInput);

break;

}

}

System.out.print("Кількість вільних місць: ");

toInput = scan.nextLine();

while (true) {

if (!Check.checkInt(toInput)) {

System.out.println("Неправильно введено кількість вільних місць\nПовторіть спробу:");

toInput = scan.nextLine();

} else {

autostation.get(id).setAmountOfFreePlaces(toInput);

break;

}

}

}

break;

}

case 3:

if (!autostation.isEmpty()) {

System.out.println("Введіть індекс автостанції");

System.out.println("Можливі індекси: ");

amountOfId();

id = scan.nextInt();

if (id < 0 || id > autostation.size() - 1) {

System.out.println("Id is impossible");

break;

} else {

autostation.remove(id);

System.out.println("Автостанцію з Id " + id + " видалено!");

break;

}

}

case 4:

if (!autostation.isEmpty()) {

System.out.println("Введіть індекс автостанції");

System.out.println("Можливі індекси: ");

amountOfId();

id = scan.nextInt();

if (id < 0 || id > autostation.size() - 1) {

System.out.println("Id is impossible");

break;

} else {

Output.printAutostation(id, autostation);

break;

}

}

case 5:

if (!autostation.isEmpty()) {

System.out.print("Введіть назву файлу: ");

scan.nextLine();

StringBuilder fileName = new StringBuilder(scan.nextLine()).append(".xml");

System.out.println("Виберіть директорію:");

String directory = Choice.setDirectory(scan) + "\\" + fileName;

try {

XMLEncoder encoder = new XMLEncoder(new BufferedOutputStream(new FileOutputStream(directory)));

encoder.writeObject(autostation);

encoder.close();

} catch (IOException ex) {

ex.printStackTrace();

}

break;

}

case 6:

System.out.println("Вкажіть розташування файлу:");

scan.nextLine();

String location = Choice.getDirectory(scan);

try{

XMLDecoder decoder = new XMLDecoder(new FileInputStream(location));

ArrayList<AutoStation> as2 = (ArrayList)decoder.readObject();

decoder.close();

for(int i = 0; i<as2.size();i++){

Output.printAutostation(0, as2);

System.out.println();

}

}

catch (IOException ex){

ex.printStackTrace();

}

break;

case 7:

System.exit(0);

default:

System.out.println("Невідома команда");

}

}

}

public static void amountOfId(){

for (int i = 0; i < count ; i++) {

if(count!=i+1)

System.out.print(i +", ");

else

System.out.println(i + ".");

}

System.out.println();

}

} **public** **class** AutoStation {

**private** Integer runningNumber;

**private** String time;

**private** String date;

**private** String amountOfFreePlaces;

**private** String arrival;

**private** String nameOfStation;

**public** Integer getRunningNumber(){

**return** runningNumber;

}

**public** **void** setRunningNumber(Integer runningNumber){

**this**.runningNumber = runningNumber;

}

**public** String getTime(){

**return** time;

}

**public** **void** setTime(String time){

**this**.time = time;

}

**public** String getDate(){

**return** date;

}

**public** **void** setDate(String date){

**this**.date = date;

}

**public** String getAmountOfFreePlfces(){

**return** amountOfFreePlaces;

}

**public** **void** setAmountOfFreePlaces(String amountOfFreePlaces){

**this**.amountOfFreePlaces = amountOfFreePlaces;

}

**public** **void** setArrival(String runningNumber){

**this**.arrival = runningNumber;

}

**public** String getArrival(){

**return** arrival;

}

**public** **void** setNameOfStation(String runningNumber){

**this**.nameOfStation = runningNumber;

}

**public** String getNameOfStation(){

**return** nameOfStation;

}

} import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class Check {

public static boolean checkDate(String string){

Pattern p = Pattern.compile("^(3[01]|[12][0-9]|0[1-9]).(1[0-2]|0[1-9]).[0-9]{4}$");

Matcher m = p.matcher(string);

return m.matches();

}

public static boolean checkString(String string){

Pattern p = Pattern.compile("^[a-zA-Z]{3,50}$");

Matcher m = p.matcher(string);

return m.matches();

}

public static boolean checkTime(String string){

Pattern p = Pattern.compile("([0-1][0-9]|2[0-3]):([0-5][0-9])");

Matcher m = p.matcher(string);

return m.matches();

}

public static boolean checkInt(String string){

Pattern p = Pattern.compile("[0-9]{3,100}");

Matcher m = p.matcher(string);

return m.matches();

}

}

import java.io.File;

import java.util.Scanner;

class Choice

{

static String setDirectory(Scanner scanner)

{

String directory = "C:\\";

String choice;

while(true)

{

System.out.println("+---------------------+");

File dir = new File(directory);

if(dir.isDirectory())

{

for(File i: dir.listFiles())

{

System.out.println(i.getName());

}

}

else

{

System.out.println("Директорія не існує, виберіть ще раз!");

}

System.out.println("Виберіть директорію. Якщо це потрібна директорія введіть '/end'!\n Щоб повернутися назад введіть '/back!'");

choice = scanner.nextLine();

if(choice.equals("/end"))

{

return directory;

}

else if(choice.equals("/back"))

{

directory = dir.getParent();

}

else

{

directory += "\\" + choice;

}

}

}

static String getDirectory(Scanner scan)

{

String directory = "C:\\";

String choice;

while(true)

{

File dir = new File(directory);

if(dir.isDirectory())

{

for(File i: dir.listFiles())

{

if(i.isDirectory() || i.getName().contains(".xml"))

{

System.out.println(i.getName());

}

}

}

else if(dir.getName().contains(".xml") || dir.isFile())

{

return directory;

}

else

{

System.out.println("Дерикторія не існує!");

}

System.out.println("Виберіть директорію, а в ній файл. Щоб повернутися назад введіть '/back'!");

choice = scan.nextLine();

if(choice.equals("/back"))

{

directory = dir.getParent();

}

else

{

directory += "\\" + choice;

}

}

}

} **import** java.util.ArrayList;

**public** **class** Output {

**public** **static** **void** printMenu() {

System.***out***.println("1 - Створити автостанцію");

System.***out***.println("2 - Записати розклад");

System.***out***.println("3 - Видалити автостанцію");

System.***out***.println("4 - Показати розклад");

System.***out***.println("5 - ser");

System.***out***.println("6 - des");

System.***out***.println("7 - Вихід");

}

**static** **void** printAutostation(**int** id, ArrayList<AutoStation> autoStations) {

System.***out***.println("\n"+ "Назва станції: "+autoStations.get(id).getNameOfStation());

System.***out***.println("Дата: "+autoStations.get(id).getDate());

System.***out***.println("Час відправлення: "+autoStations.get(id).getTime());

System.***out***.println("Час прибуття: "+autoStations.get(id).getArrival());

System.***out***.println("Кількість вільних місць: "+autoStations.get(id).getAmountOfFreePlfces());

}

}

Висновок: Я ознайомився з принципами використання регулярних виразів для перевірки рядка на відповідність шаблону та з принципами використання регулярних виразів для обробки тексту.