Національний університет «Львівська Політехніка»

Кафедра Систем Штучного Інтелекту



Лабораторна робота №4

Виконав: студент КН-111

Зомчак Богдан

Перевірив:

Грабовська Н. Р.

**Львів-2019**

Тема:

Параметризація в Java. Обробка параметризованих контейнерів

Мета:

Вивчення принципів параметризації в Java. Розробка параметризованих класів та методів. Розширення функціональності параметризованих класів.

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.io.\*;

import java.util.Locale;

import java.util.Scanner;

import static java.lang.System.in;

public class Main {

public static int count =0;

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

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

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

}

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

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

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.size() != 0) {

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

autostation.add(new AutoStation());

id = autostation.size() - 1;

count++;

autostation.get(id).setNameOfStation("Lviv");

autostation.get(id).setRunningNumber(22);

autostation.get(id).setArrival("15:47");

autostation.get(id).setDate("12.02.2019");

autostation.get(id).setAmountOfFreePlaces(47);

autostation.get(id).setTime("15:32");

break;

} else {

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;

}

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

scan.nextLine();

autostation.get(id).setNameOfStation(scan.nextLine());

System.out.println("Номер рейсу: ");

autostation.get(id).setRunningNumber(scan.nextInt());

scan.nextLine();

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

autostation.get(id).setTime(scan.nextLine());

System.out.println("Прибуття: ");

autostation.get(id).setArrival(scan.nextLine());

System.out.println("Дата: ");

autostation.get(id).setDate(scan.nextLine());

System.out.println("Вільні місця: ");

autostation.get(id).setAmountOfFreePlaces(scan.nextInt());

break;

}

}

case 3:

if (autostation.size() != 0) {

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.size() != 0) {

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;

}

} else {

System.out.println("Розклад пустий");

break;

}

case 5:

try (ObjectOutputStream oos = new ObjectOutputStream(

new FileOutputStream("tempfile.bat"))) {

LinkedList<AutoStation> c = autostation;

oos.writeObject(c);

} catch (Exception ex) {

System.out.println(ex.getMessage());

}

System.out.println("Дані записано!");

break;

case 6:

try(ObjectInputStream ois = new ObjectInputStream(

new FileInputStream("tempfile.bat"))) {

@SuppressWarnings("unchecked")

LinkedList<AutoStation> ca = (LinkedList<AutoStation>) ois.readObject();

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

autostation.add(ca.get(i));

}

System.out.println("Дані успішно зчитані з файлу");

count++;

}

catch(Exception ex) {

System.out.println(ex.getMessage());

}

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();

}

} **import** java.io.Serializable;

**public** **class** AutoStation **implements** Serializable {

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** Integer runningNumber;

**private** String time;

**private** String date;

**private** **int** 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** Integer getAmountOfFreePlfces(){

**return** amountOfFreePlaces;

}

**public** **void** setAmountOfFreePlaces(**int** 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.io.Serializable;

import java.util.Iterator;

public class LinkedList<T> implements Iterable<T>, Serializable {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private class Node implements Serializable {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private T data;

private Node next;

private T getData() {

return data;

}

private Node getNext() {

return next;

}

private void setData(T data) {

this.data = data;

}

private void setNext(Node next) {

this.next = next;

}

}

private Node root;

public LinkedList() {

root = new Node();

}

public void add(T data) {

Node last = root;

while(last.getNext() != null)

last = last.getNext();

last.setNext(new Node());

last.getNext().setData(data);

}

public T get(int index) {

Node result = root;

for (int i = -1; i < index; i++) {

if (result.getNext() == null)

return null;

else

result = result.getNext();

}

return result.getData();

}

public boolean remove(int index) {

Node result = root;

if (size() != 0 && index>=0) {

for (int i = -1; i < index-1; i++)

if (result.getNext() == null)

return false;

else

result = result.getNext();

if (result.getNext() == null)

return false;

else if (result.getNext().getNext() == null) {

result.setNext(null);

return true;

}

else {

result.setNext(result.getNext().getNext());

return true;

}

}

else return false;

}

public int size() {

Node last = root;

int count = 0;

while(last.getNext() != null) {

count++;

last = last.getNext();

}

return count;

}

public void clear() {

root = new Node();

}

public String toString() {

if (size() == 0) return "[ - ]";

StringBuilder result = new StringBuilder();

result.append("[ ");

for (int i = 0; i < size()-1; i++) {

result.append(get(i)).append(", ");

}

result.append(get(size()-1)).append(" ]");

return new String(result);

}

public T[] toArray() {

@SuppressWarnings("unchecked")

T[] result = (T[]) new Object[size()];

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

result[i] = get(i);

}

return result;

}

@Override

public Iterator<T> iterator() {

return new Iterator<T>() {

int counter;

{

counter = 0;

}

@Override

public boolean hasNext() {

return counter < size();

}

@Override

public T next() {

return get(counter++);

}

};

}

} **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, LinkedList<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("Маршрут:"+autoStations.get(id).getRunningNumber());

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

}

**static** **void** printApplication(AutoStation autoStation){

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

System.***out***.println("Дата: " + autoStation.getDate());

System.***out***.println("Час відправлення: " + autoStation.getTime());

System.***out***.println("Час прибуття: " + autoStation.getArrival());

System.***out***.println("Маршрут:" + autoStation.getRunningNumber());

System.***out***.println("Кількість вільних місць" + autoStation.getAmountOfFreePlfces());

}

}

Висновок: Я вивчив принципи параметризації в Java. Навчився розробляти параметризовані класи та методи та розширювати функціональність параметризованих класів.