Національний університет «Львівська Політехніка»

Кафедра Систем Штучного Інтелекту



Лабораторна робота №2

Виконав: студент КН-111

Зомчак Богдан

Перевірив:

Грабовська Н. Р.

**Львів-2019**

Тема:

Розробка власних контейнерів. Ітератори. Серіалізація/десеріалізація об’єктів. Бібліотека класів користувача

Мета:

Набуття навичок розробки власних контейнерів. Використання ітераторів. Тривале зберігання та відновлення стану об’єктів. Ознайомлення з принципами серіалізації/десеріалізації об’єктів. Використання бібліотек класів користувача.

import java.util.\*;

import static java.lang.System.in;

import java.io.\*;

import java.nio.file.Files;

import java.nio.file.Paths;

import java.nio.file.StandardOpenOption;

import java.lang.\*;

public class Main {

public static int count = 0;

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

Scanner scan = new Scanner(in);

ArrayList<MyContainer> listOfContainers = new ArrayList<>();

while (true){

Output.menu();

int id;

int indexToDelete;

int command = scan.nextInt();

switch (command){

case 1:

listOfContainers.add(new MyContainer());

System.out.println("New Container is added, id = " +count);

count++;

break;

case 2:

if(!listOfContainers.isEmpty()){

System.out.println("Enter id of Container");

System.out.println("Possible id - ");

amountOfId();

id = scan.nextInt();

System.out.println("Enter number of strings that you want to write:");

int numOfStrings = scan.nextInt();

System.out.println("Enter " + numOfStrings + " strings:");

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

listOfContainers.get(id).Str.add(scan.nextLine());

}

listOfContainers.get(id).Str.remove(0);

System.out.println("Done!");

break;

}

else {

System.out.println("Container is empty");

break;

}

case 3:

if(!listOfContainers.isEmpty()) {

System.out.println("Enter id of Container");

System.out.println("Possible id - ");

amountOfId();

id = scan.nextInt();

System.out.print("Container data: ");

System.out.println(listOfContainers.get(id).Str.toString());

break;

}

else{

System.out.println("Container is empty");

break;

}

case 4:

if(!listOfContainers.isEmpty()) {

System.out.println("Enter id of Container");

System.out.println("Possible id - ");

amountOfId();

id = scan.nextInt();

listOfContainers.get(id).Str.clear();

System.out.println("Your container is cleared");

break;

}

else {

System.out.println("Container is empty");

break;

}

case 5:

if(!listOfContainers.isEmpty()) {

System.out.println("Enter id of Container");

System.out.println("Possible id - ");

amountOfId();

id = scan.nextInt();

System.out.println("Enter your string:");

listOfContainers.get(id).Str.add(scan.nextLine());

listOfContainers.get(id).Str.add(scan.nextLine());

listOfContainers.get(id).Str.remove(listOfContainers.get(id).Str.size()-2);

System.out.println("Done!");

break;

}

else {

System.out.println("Container is empty");

break;

}

case 6:

if(!listOfContainers.isEmpty()) {

System.out.println("Enter id of Container");

System.out.println("Possible id - ");

amountOfId();

id = scan.nextInt();

System.out.println("Enter index of string you want to delete:");

indexToDelete = scan.nextInt();

if (indexToDelete < 0 || indexToDelete > listOfContainers.get(id).Str.size() - 1) {

System.out.println("Wrong index! Try again:");

} else {

listOfContainers.get(id).Str.remove(indexToDelete);

System.out.println("Success! String by index " + indexToDelete + " deleted");

break;

}

}

else {

System.out.println("Container is empty");

break;

}

case 7:

if(!listOfContainers.isEmpty()) {

System.out.println("Enter id of Container");

System.out.println("Possible id - ");

amountOfId();

id = scan.nextInt();

for (Object i : listOfContainers.get(id).Str.toArray()) {

System.out.println(i);

}

break;

}

else {

System.out.println("Container is empty");

break;

}

case 8:

if(!listOfContainers.isEmpty()) {

System.out.println("Enter id of Container");

System.out.println("Possible id - ");

amountOfId();

id = scan.nextInt();

System.out.println(listOfContainers.get(id).Str.size());

break;

}

else {

System.out.println("Container is empty");

break;

}

case 9:

if(!listOfContainers.isEmpty()) {

System.out.println("Enter id of Container");

System.out.println("Possible id - ");

amountOfId();

id = scan.nextInt();

Files.write(Paths.get("file.txt"), listOfContainers.get(0).Str, StandardOpenOption.CREATE);

System.out.println("Container saved to file.txt");

break;

}

else {

System.out.println("Container is empty");

break;

}

case 10:

if(!listOfContainers.isEmpty()) {

System.out.println("Enter id of Container");

System.out.println("Possible id - ");

amountOfId();

id = scan.nextInt();

File file = new File("file.txt");

Scanner fr = new Scanner(file);

while (fr.hasNextLine()) {

listOfContainers.get(id).Str.add(fr.nextLine());

}

System.out.println("Uploaded from file.fxt to container");

break;

}

else {

System.out.println("Container is empty");

break;

}

case 11:

if(!listOfContainers.isEmpty()) {

System.out.println("Enter id of Container");

System.out.println("Possible id - ");

amountOfId();

id = scan.nextInt();

Collections.sort(listOfContainers.get(id).Str);

break;

}

else {

System.out.println("Container is empty");

break;

}

case 12:

if(!listOfContainers.isEmpty()) {

System.out.println("Enter id of Container");

System.out.println("Possible id - ");

amountOfId();

id = scan.nextInt();

for (String i : listOfContainers.get(id).Str) {

System.out.println(i);

}

break;

}

else {

System.out.println("Container is empty");

break;

}

case 13:

System.exit(0);

default:

System.out.println("Wrong command");

break;

}

}

}

public static void amountOfId(){

System.out.println("Id:");

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;

import java.util.ArrayList;

public class MyContainer implements Serializable{

public ArrayList<String> Str = new ArrayList<>();

}

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

**public** **static** **void** menu(){

System.***out***.println("\n\tMENU\t");

System.***out***.println("1 - make container");

System.***out***.println("2 - fill container");

System.***out***.println("3 - show container");

System.***out***.println("4 - clear container");

System.***out***.println("5 - add element");

System.***out***.println("6 - remove string");

System.***out***.println("7 - convert to array");

System.***out***.println("8 - current size");

System.***out***.println("9 - write to file (serialize)");

System.***out***.println("10 - read from file (deserialize)");

System.***out***.println("11 - sort");

System.***out***.println("12 - iterate through container (foreach)");

System.***out***.println("13 - exit");

}

}

Висновок: Я набув навичок розробки власних контейнерів, використання ітераторів, тривале зберігання та відновлення стану об’єктів. Ознайомився з принципами серіалізації/десеріалізації об’єктів. Навчився використовувати бібліотеки класів користувача.