Міністерство освіти і науки України Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



Лабораторна робота №9

на тему: «Розробка власних контейнерів. Ітератори. Серіалізація/десеріалізація об’єктів. Бібліотека класів користувача»

з курсу: « Об’єктно-орієнтоване програмування »

Виконав:

ст. гр. КН-110

Козій Дмитро

Прийняв:

Старший викладач Гасько Р. Т.

Львів – 2017 р.

Лабораторна робота №9

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.

Код програми:

|  |  |
| --- | --- |
| import java.awt.\*; |  |
|  | import java.io.FileOutputStream; |
|  | import java.io.IOException; |
|  | import java.io.ObjectOutputStream; |
|  | import java.io.Serializable; |
|  | import java.util.Iterator; |
|  |  |
|  | class SelfCollection implements Serializable { |
|  | private static int arrSize = 15; |
|  | private static String arr[] = new String[arrSize]; |
|  | private static int end = 0; |
|  | @Override |
|  | public String toString(){ |
|  | StringBuilder string = new StringBuilder(); |
|  | for(int i = 0; i < end; i++) |
|  | string.append(arr[i]); |
|  | return string.toString(); |
|  | } |
|  | public void add(String string){ |
|  | if(end == arrSize - 1){ |
|  | arrSize \*= 2; |
|  | String newArr[] = new String[arrSize]; |
|  | for(int i = 0; i < end; i++) |
|  | newArr[i] = arr[i]; |
|  | newArr[end] = string; |
|  | arr = newArr; |
|  | } |
|  | else |
|  | arr[end++] = string; |
|  | } |
|  | public void clear(){ |
|  | for(int i = 0; i < end; i++){ |
|  | arr[i] = null; |
|  | } |
|  | } |
|  | public boolean remove(String string){ |
|  | for(int i = 0; i < end; i++){ |
|  | if(string.equals(arr[i])){ |
|  | for(;i + 1 < end; i++){ |
|  | arr[i] = arr[i+1]; |
|  | } |
|  | arr[--end] = null; |
|  | return true; |
|  | } |
|  | } |
|  | return false; |
|  | } |
|  | public Object[] toArray(){ |
|  | return arr; |
|  | } |
|  | public int size(){ |
|  | return end; |
|  | } |
|  | public boolean contains(String string){ |
|  | for(int i = 0; i < end; i++){ |
|  | if(string.equals(arr[i])){ |
|  | return true; |
|  | } |
|  | } |
|  | return false; |
|  | } |
|  | public boolean ContainsAll(Container container){ |
|  | for(String x : arr){ |
|  | if(!contains(x)) |
|  | return false; |
|  | } |
|  | return true; |
|  | } |
|  | public Iterator<String> iterator(){ |
|  | final int[] iterable = {0}; |
|  | return new Iterator<String>() { |
|  | @Override |
|  | public boolean hasNext() { |
|  | if(iterable[0] != end) |
|  | return true; |
|  | else return false; |
|  | } |
|  |  |
|  | @Override |
|  | public String next() { |
|  | return arr[iterable[0]++]; |
|  | } |
|  | }; |
|  | } |
|  | } |
|  |  |
|  | public class Main { |
|  | public static void main(String[] args) throws IOException { |
|  | SelfCollection var = new SelfCollection(); |
|  | for(int i = 0; i < 20; i++) |
|  | var.add(" " + i + 20); |
|  | var.remove(" 4"); |
|  | Iterator<String> gener = var.iterator(); |
|  | while(gener.hasNext()) |
|  | System.out.print(gener.next()); |
|  | FileOutputStream fos = new FileOutputStream("temp.out"); |
|  | ObjectOutputStream oos = new ObjectOutputStream(fos); |
|  | SelfCollection ts = new SelfCollection(); |
|  | oos.writeObject(var); |
|  | oos.flush(); |
|  | oos.close(); |
|  | } |
|  | } |

Висновок: На цій лабораторній роботі я навчився розробляти власні контейнери та використовувати ітератори.