Лунгол О.С

КІТ-188б

Лабораторна робота 15

[Колекції в Java](https://oop-khpi.gitlab.io/#tasks)

## Мета

* Ознайомлення з бібліотекою колекцій Java SE.
* Використання колекцій для розміщення об'єктів розроблених класів.

## Вимоги

1. Розробити консольну програму для реалізації завдання обробки даних згідно [прикладної області](https://oop-khpi.gitlab.io/%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F/task07/#_4).
2. Для розміщення та обробки даних використовувати контейнери (колекції) і алгоритми з [Java Collections Framework](https://docs.oracle.com/javase/8/docs/technotes/guides/collections/).
3. Забезпечити обробку колекції об'єктів: додавання, видалення, пошук, сортування згідно розділу [Прикладні задачі л.р. №10](https://oop-khpi.gitlab.io/%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F/task10/#_3).
4. Передбачити можливість довготривалого зберігання даних: 1) за допомогою стандартної серіалізації; 2) не використовуючи протокол серіалізації.
5. Продемонструвати розроблену функціональність в діалоговому та автоматичному режимах за результатом обробки параметрів командного рядка.

Container

package ua.khpi.oop.Lunhol15.Container;  
  
import ua.khpi.oop.Lunhol15.Head.HDR1;  
  
import java.io.\*;  
import java.util.Arrays;  
import java.util.*Iterator*;  
import java.util.LinkedList;  
import java.util.NoSuchElementException;  
  
public class Container implements *Serializable {* LinkedList*<*HDR1*>* linkedList = new LinkedList*<>()*;  
  
 public boolean notEmpty*() {* return linkedList.size*()* > 0;  
 *}* public void clear*() {* linkedList.clear*()*;  
 *}* public void add*(*final HDR1 *hdr1) {* addLast*(hdr1)*;  
 *}* public void saveAll*() {* try *{* File file = new File*(*"save.txt"*)*;  
 if *(*!file.exists*()) {* file.createNewFile*()*;  
 *}* PrintWriter pw = new PrintWriter*(*file*)*;  
 System.*out*.println*()*;  
 pw.println*(*linkedList.size*())*;  
 for *(*int i = 0; i < linkedList.size*()*; i++*) {* pw.println*(*getElementByIndex*(*i*))*;  
 *}* pw.close*()*;  
 *}* catch *(*IOException *e) {* System.*out*.println*(*"Error" + *e)*;  
 *}  
 }* public void saveRec*() {* try *{* File file = new File*(*"save.txt"*)*;  
 if *(*!file.exists*()) {* file.createNewFile*()*;  
 *}* PrintWriter pw = new PrintWriter*(*file*)*;  
 HDR1 temp;  
 System.*out*.println*()*;  
 pw.println*(*linkedList.size*())*;  
 for *(*int i = 0; i < linkedList.size*()*; i++*) {* temp = getElementByIndex*(*i*)*;  
 pw.println*(*temp.getNumberPassport*())*;  
 pw.println*(*temp.getEducation*())*;  
 pw.println*(*temp.getPayment*())*;  
 pw.println*(*temp.getData*())*;  
 pw.println*(*temp.getPosition*())*;  
 pw.println*(*temp.getDepartment*())*;  
 pw.println*(*temp.getWorkingConditions*())*;  
 pw.println*(*temp.getMarks*())*;  
 *}* pw.close*()*;  
 *}* catch *(*IOException *e) {* System.*out*.println*(*"Error" + *e)*;  
 *}  
 }* public void readAll*() {* try *(*BufferedReader br = new BufferedReader*(*new FileReader*(*"save.txt"*))) {* Object temp;  
 String line;  
 line = br.readLine*()*;  
 int count = Integer.*parseInt(*line*)*;  
 for *(*int i = 0; i < count; i++*) {* line = br.readLine*()*;  
 temp = line;  
 add*((*HDR1*)* temp*)*;  
 *}  
 }* catch *(*IOException *ex) {  
 ex*.printStackTrace*()*;  
 *}  
 }* public void readRec*()* throws IOException *{* BufferedReader br = null;  
 HDR1 temp = new HDR1*()*;  
 try *{* br = new BufferedReader*(*new FileReader*(*"save.txt"*))*;  
 String line;  
 line = br.readLine*()*;  
 int count = Integer.*parseInt(*line*)*;  
 for *(*int i = 0; i < count; i++*) {* line = br.readLine*()*;  
 temp.setNumberPassport*(*Integer.*parseInt(*line*))*;  
 line = br.readLine*()*;  
 temp.setEducation*(*line*)*;  
 line = br.readLine*()*;  
 temp.setPayment*(*Integer.*parseInt(*line*))*;  
 line = br.readLine*()*;  
 temp.setData*(*line*)*;  
 line = br.readLine*()*;  
 temp.setPosition*(*line*)*;  
 line = br.readLine*()*;  
 temp.setDepartment*(*line*)*;  
 line = br.readLine*()*;  
 temp.setWorkingConditions*(*line*)*;  
 line = br.readLine*()*;  
 temp.setMarks*(*Integer.*parseInt(*line*))*;  
 add*(* new HDR1*(*temp*))*;  
 *}  
 }* catch *(*IOException *ex) {  
 ex*.printStackTrace*()*;  
 *}* finally *{* br.close*()*;  
 *}  
 }* public Object*[]* toArray*() {* return linkedList.toArray*()*;  
 *}* public void addLast*(*final HDR1 *obj) {* linkedList.addLast*(obj)*;  
 *}* public void addFirst*(*final HDR1 *obj) {* linkedList.addFirst*(obj)*;  
 *}* public int size*() {* return linkedList.size*()*;  
 *}* public HDR1 getElementByIndex*(*final int *index) {* return linkedList.get*(index)*;  
 *}* public void sortByMarks*() {* linkedList.sort*(*HDR1.*compareByMarks)*;  
 *}* public void sortByPassport*() {* linkedList.sort*(*HDR1.*compareByPassport)*;  
 *}* public void sortByPayment*() {* linkedList.sort*(*HDR1.*compareByPayment)*;  
 *}* public int search*(*HDR1 *obj) {* int index = 0;  
 for *(*HDR1 hdr1 : linkedList*) {* if *(*hdr1 == *obj) {* return index;  
 *}  
 }* return -1;  
 *}* public void saveSerializable*() {* try *{* ObjectOutputStream oo = new ObjectOutputStream*(*new FileOutputStream*(*"save.ser"*))*;  
 oo.writeObject*(*linkedList*)*;  
 oo.close*()*;  
 *}* catch *(*IOException *ioe) {* System.*out*.println*(*"Error"*)*;  
 *}  
 }* public void readSerializable*() {* try *{* FileInputStream fileInput = new FileInputStream*(*"save.ser"*)*;  
 ObjectInputStream objectInput = new ObjectInputStream*(*fileInput*)*;  
 LinkedList*<*HDR1*>* temp ;  
 temp = *(*LinkedList*<*HDR1*>)* objectInput.readObject*()*;  
 System.*out*.println*(*temp*)*;  
 objectInput.close*()*;  
 *}* catch *(*Exception *exc) {* System.*out*.println*(*"Error"*)*;  
 *}  
 }* public void show*() {* int count = 0;  
 for *(*HDR1 s : linkedList*) {* System.*out*.println*(*"#" + *(*++count*))*;  
 System.*out*.println*(*s.toString*())*;  
 *}  
 }  
  
}*

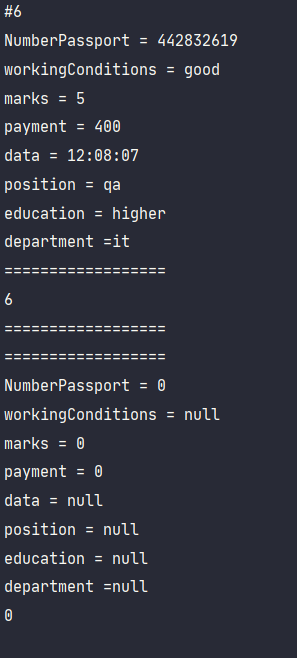


Рисунок 1 – Результат роботи auto

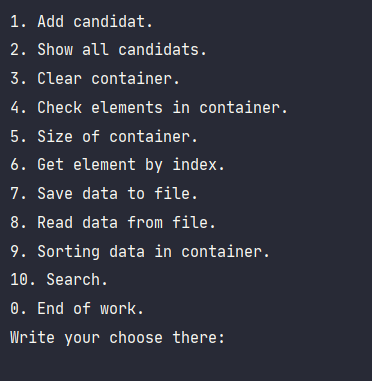


Рисунок 2 – Результат роботи menu

Висновок

В данні лабораторній роботі ми ознайомились з бібліотекою колекцій Java SE. Використали колекції для розміщення об'єктів розроблених класів.