Университет ИТМО

Кафедра ВТ

Программирование

Лабораторная работа №5

Вариант 112

Группа P3110

Нгу Фыонг Ань

ПРОВЕРИЛ:

2018 год

**Задание:**

Реализовать на базе программы из [лабораторной работы №4](https://se.ifmo.ru/courses/#lab4) консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме. Номенклатуру сохраняемых в коллекции объектов необходимо заранее согласовать с преподавателем..

**Разработанная программа должна удовлетворять следующим требованиям:**

* Класс, коллекцией экземпляров которого управляет программа, должен реализовывать сортировку по умолчанию.
* Для хранения необходимо использовать коллекцию типа java.util.HashSet.
* При запуске приложения коллекция должна автоматически заполняться значениями из файла.
* Имя файла должно передаваться программе с помощью переменной окружения.
* Данные должны храниться в файле в формате xml.
* При остановке приложения текущее состояние коллекции должно автоматически сохраняться в файл.
* Чтение данных из файла необходимо реализовать с помощью класса java.util.Scanner.
* Запись данных в файл необходимо реализовать с помощью класса java.io.OutputStreamWriter.
* Все реализованные команды (см. ниже) должны быть задокументированы в формате javadoc.
* Формат задания объектов в командах - json.

**В интерактивном режиме программа должна поддерживать выполнение следующих команд:**

* remove\_lower {element}: удалить из коллекции все элементы, меньшие, чем заданный
* save: сохранить коллекцию в файл
* add\_if\_max {element}: добавить новый элемент в коллекцию, если его значение превышает значение наибольшего элемента этой коллекции
* load: перечитать коллекцию из файла

**Диаграмма классов:**

**Исходный код:**

***Файл lab5.java:***

**package Pack;**

**import java.io.\*;**

**import java.util.\*;**

**public class Lab5 {**

**public static void main(String[] args) throws FileNotFoundException, AgeException, IOException {**

**UnfamiliarMan Nick = new UnfamiliarMan("Nick", 23,5,25);**

**HashSet<WorkMan> workMan = new HashSet<WorkMan>();**

**String fileName = "C:\\Users\\phany\\OneDrive\\Documents\\NetBeansProjects\\Lab5\_959\\src\\Pack\\WorkMan.xml";**

**try {XmlFile.read(fileName, workMan);}**

**catch (FileNotFoundException ex) {**

**System.out.println("File input not found");**

**}**

**Scanner scanner = new Scanner(System.in);**

**while (true){**

**String s = scanner.nextLine();**

**if (s.equals("exit")) {**

**break;**

**}**

**if (s.equals("print")) {**

**for (WorkMan w : workMan) {**

**System.out.println(w.toString());**

**}**

**}**

**if (s.startsWith("remove\_lower ")) {**

**WorkMan w = WorkMan.jsonToObject(s.replaceFirst("remove\_lower ", ""));**

**for (Iterator<WorkMan> i = workMan.iterator(); i.hasNext();) {**

**WorkMan wm = i.next();**

**if (w.compareTo(wm)>0){**

**i.remove();**

**}**

**}**

**}**

**if (s.startsWith("add\_if\_max ")) {**

**WorkMan w = WorkMan.jsonToObject(s.replaceFirst("add\_if\_max ", ""));**

**boolean ok = true;**

**for (Iterator<WorkMan> i = workMan.iterator(); i.hasNext();) {**

**WorkMan wm = i.next();**

**if (w.compareTo(wm)<0) ok = false;**

**}**

**if (ok) workMan.add(w);**

**}**

**if (s.startsWith("save"))**

**XmlFile.write(fileName, workMan);**

**if (s.startsWith("load")) {**

**workMan = new HashSet<WorkMan>();**

**try {XmlFile.read(fileName, workMan);}**

**catch (FileNotFoundException ex) {**

**System.out.println("File input not found");**

**}**

**}**

**}**

**}**

**}**

***Файл* AgeException*.java***

**package Pack;**

**public class AgeException extends Exception{**

**@Override**

**public String getMessage(){**

**return "Age value is wrong";**

**}**

**}**

***Файл* SpeedException*.java***

**package Pack;**

**public class SpeedException extends Exception {**

**@Override**

**public String getMessage(){**

**return "Speed value if wrong";**

**}**

**}**

***Файл* Animal*.java:***

**package Pack;**

**public class Animal {**

**public String name;**

**public int age;**

**public int volum;**

**public int speed;**

**public Animal(String name , int age, int volum, int speed){**

**this.name = name;**

**this.age = age;**

**this.volum = volum;**

**this.speed = speed;**

**}**

**public String getName(){**

**return name;**

**}**

**public int getAge() {**

**return age;**

**}**

**public int getSpeed() {**

**return speed;**

**}**

**public int getVolum() {**

**return volum;**

**}**

**public void setVolum(int volum) {**

**this.volum = volum;**

**}**

**public void setName(String name){**

**this.name = name;**

**}**

**public void setSpeed(int speed){**

**this.speed = speed;**

**}**

**}**

***Файл Human.java:***

**package Pack;**

**public abstract class Human extends Animal{**

**public Human(String name, int age, int volum, int speed) {**

**super(name, age, volum, speed);**

**}**

**interface SuperRunSkill{**

**public void superrun();**

**}**

**SuperRunSkill redHaste = new SuperRunSkill() {**

**@Override**

**public void superrun(){**

**incSpeed(2);**

**}**

**};**

**SuperRunSkill blackHaste = new SuperRunSkill() {**

**@Override**

**public void superrun(){**

**decSpeed(2);**

**}**

**};**

**abstract void skill();**

**abstract void run();**

**abstract void jump();**

**abstract void punch();**

**@Override**

**public String toString() {**

**return "\*" + name + " " + age + " YEARS OLD ";**

**}**

**class Haste implements SuperRunSkill {**

**@Override**

**public void superrun(){**

**System.out.println("this is bottle rune special"+name);**

**}**

**}**

**public void setAge(int age) throws AgeException {**

**if (age <0) {**

**throw new AgeException();**

**}**

**this.age = age;**

**}**

**public void setSpeed(int speed) {**

**if (speed < 0){**

**throw new SpeedException();**

**}**

**this.speed = speed;**

**}**

**public void decSpeed(int time) {**

**this.speed = speed/time;**

**}**

**public void incSpeed(int time) {**

**this.speed = speed\*time;**

**}**

**protected void test(){**

**}**

**}**

***Файл SpeakSkill.java:***

**package Pack;**

**public interface SpeakSkill {**

**public void Speak();**

**}**

***Файл UnfamiliarMan.java:***

**package Pack;**

**public class UnfamiliarMan extends Human implements SpeakSkill{**

**public UnfamiliarMan(String name, int age, Position position, int volum, int speed) {**

**super(name, age, position, volum, speed);**

**}**

**@Override**

**public void skill() {**

**System.out.println("SKILL: Punch, Run, Jump");**

**}**

**@Override**

**public void punch(){**

**System.out.println("PUNCH WITH " + speed\*1.5 + "N");**

**}**

**@Override**

**public void jump() {**

**System.out.println("JUMP TO " + speed\*0.8 + "M");**

**}**

**@Override**

**public void run() {**

**System.out.println("RUN WITH THE SPEED OF " + speed + "KM/H" );**

**}**

**@Override**

**public void Speak(){**

**System.out.println("PLEASE HELP ME TO FIND MY FRIEND");**

**}**

**}**

***Файл WorkMan.java:***

**package Pack;**

**import org.json.simple.\*;**

**import org.json.simple.parser.\*;**

**import java.util.\*;**

**public class WorkMan extends Human implements WorkSkill, SpeakSkill, Comparable<WorkMan>{**

**String job;**

**boolean love;**

**public WorkMan(String name, int age, int volum, int speed, String job, boolean love) {**

**super(name, age, volum, speed);**

**this.job = job;**

**this.love = love;**

**}**

**public String getJob(){**

**return job;**

**}**

**public boolean getLove(){**

**return love;**

**}**

**public void setJob(String job){**

**this.job = job;**

**}**

**public void setLove(boolean love){**

**this.love = love;**

**}**

**@Override**

**public void skill() {**

**System.out.println("SKILL: Punch, Run, Jump");**

**}**

**@Override**

**public void punch(){**

**System.out.println("PUNCH WITH " + speed + "N");**

**}**

**@Override**

**public void jump() {**

**System.out.println("JUMP TO " + speed\*0.4 + "M");**

**}**

**@Override**

**public void run() {**

**System.out.println("RUN WITH THE SPEED OF " + speed\*0.8 + "KM/H" );**

**}**

**@Override**

**public void WorkAs(){**

**System.out.println(name + " IS WORKING AS " + job);**

**}**

**@Override**

**public void Speak(){**

**if (love) System.out.println(name + " LOVES WORKING AS " + job); else**

**System.out.println(name + " HATES WORKING AS " + job);**

**}**

**@Override**

**public String toString() {**

**return name + " " + age + " " + volum + " " + speed + " " + job + " " + love;**

**}**

**@Override**

**public int compareTo( WorkMan w){**

**return (this.getName()+this.getAge()).compareTo(w.getName()+w.getAge());**

**}**

**public static WorkMan jsonToObject(String s) throws AgeException{**

**WorkMan w = new WorkMan("", 0, 0, 0, "", false);**

**try {**

**// 1. Create a JSONParser**

**JSONParser jsP = new JSONParser();**

**// 2. Parser string JSON into a JSONObject**

**JSONObject jsO = (JSONObject) jsP.parse(s);**

**// 3. Get values from jsonObject through Key**

**String name = (String) jsO.get("name");**

**if (name != null) w.setName(name);**

**try{**

**int age = Integer.parseInt((String) jsO.get("age"));**

**try {w.setAge(age);**

**} catch (AgeException ex){**

**System.out.println("Age exception");**

**}**

**} catch (NumberFormatException ex) {}**

**try{**

**int volum = Integer.parseInt((String) jsO.get("volum"));**

**w.setVolum(volum);**

**} catch (NumberFormatException ex) {}**

**try{**

**int speed = Integer.parseInt((String) jsO.get("speed"));**

**w.setSpeed(speed);**

**} catch (NumberFormatException ex) {}**

**String job = (String) jsO.get("job");**

**if (job != null) w.setJob(job);**

**String l = (String) jsO.get("love");**

**if (l != null) {**

**boolean love;**

**love = ("true".equals(l));**

**w.setLove(love);**

**}**

**System.out.println(w.toString());**

**} catch (ParseException e) {**

**e.printStackTrace();**

**}**

**return w;**

**}**

**}**

***Файл WorkSkill.java:***

**package Pack;**

**public interface WorkSkill {**

**public void WorkAs();**

**}**

**Вывод программы:**

print

Phill 25 26 60 Miner false

Nico 34 12 54 Teacher false

Nick 34 50 100 Farmer true

Alice 14 20 60 Student true

Flora 24 30 40 Florist true

Lily 34 65 32 Baker true

remove\_lower {"name":"Minie", "age":"25"}

Minie 25 0 0 false

print

Phill 25 26 60 Miner false

Nico 34 12 54 Teacher false

Nick 34 50 100 Farmer true

save

load

print

Phill 25 26 60 Miner false

Nico 34 12 54 Teacher false

Nick 34 50 100 Farmer true

**Вывод:** в ходе лабораторной работы мне удалось реализовать на базе программы из лабораторной работы №4 консольное приложение, которое реализует управление коллекцией объектов в интерактивном режиме, используя пакеты java.util.HashSet, java.util.Scanner, java.io.OutputStreamWriter и формат json.