

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА
ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту



Лабораторна робота N7

З дисципліни

«Алгоритмізація та програмування»

Виконав:

Студент групи КН-108

Воробель Адріан

Викладач:

Грабовська Н.Р.

Мета:

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

1. Вимоги

1. Розробити консольну програму для реалізації завдання обробки даних згідно Прикладні задачі. Список №2. 20 варіантів.
2. Для розміщення та обробки даних використовувати контейнери (колекції) і алгоритми з Java Collections Framework .
3. Забезпечити обробку колекції об'єктів: додавання, видалення, пошук, сортування згідно списку з Прикладні задачі. Список №2. 20 варіантів).
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.

1.1 Розробник

Воробель Адріан, КН-108, номер варіанту індивідуального завдання – 5.

1.2 Задача

Опис програми:

Клас-контейнер, створений з використанням Java Collections Framework, зокрема із сортуванням за допомогою компаратора

2.1 Засоби ООП

Клас-контейнер, domain-клас та його підклас, Main.

2.2 Ієрархія та структура класів

Клас TipaContainer – контейнер з реалізованими методами управління об'єктами контейнера.

TradePoint – domain-клас та його підклас WorkTime.

```

package com.company;

import java.beans.XMLDecoder;
import java.beans.XMLEncoder;
import java.io.*;
import java.util.ArrayList;
import java.util.Comparator;

public class TipaContainer {

    protected ArrayList<TradePoint> ShopBase = new ArrayList();
    protected TradePoint shop = new TradePoint();

    public void add(TradePoint object){ ShopBase.add(object); }

    public void remove(int index) { ShopBase.remove(index); }

    public void get(int index){
        if(index < ShopBase.size()) {
            shop = ShopBase.get(index);
            shop.time = shop.getWorkTime();
            System.out.println(shop.getName() + shop.getAddress() + shop.getSpecialization());
            System.out.println(shop.time.getMONDAY() + shop.time.getTUESDAY() + shop.time.getWEDNESDAY() + shop.time.getTHURSDAY() +
                shop.time.getFRIDAY() + shop.time.getSATURDAY() + shop.time.getSUNDAY());
            for (String i : shop.getNumbers()) {
                System.out.print(i);
            }
        } else if(index<0 || index>= ShopBase.size())
            System.out.println("invalid index");
    }

    public void getALL(){
        for (int i = 0; i < ShopBase.size(); i++) {
            shop = ShopBase.get(i);
            shop.time = shop.getWorkTime();
            System.out.println(shop.getName() + shop.getAddress() + shop.getSpecialization());
            System.out.println(shop.time.getMONDAY() + shop.time.getTUESDAY() + shop.time.getWEDNESDAY() + shop.time.getTHURSDAY() +
                shop.time.getFRIDAY() + shop.time.getSATURDAY() + shop.time.getSUNDAY());
            for (String y : shop.getNumbers()) {
                System.out.print(y);
            }
        }
    }

    public int search (String name){
        int find = -1;
        for (int i = 0; i < ShopBase.size(); i++) {
            shop = ShopBase.get(i);
            if(shop.getName().equals(name+"\n")) {
                find = i;
                break;
            }
        }
    }
}

```

```

1 package com.company;
2
3 public class TradePoint {
4
5     private String specialization;
6     private String name;
7     private String address;
8     private String numbers;
9     private String WorkTime;
10
11     public String getName() { return name; }
12
13     public void setName(String name) { this.name = name; }
14
15
16
17
18
19     public String getAddress() { return address; }
20
21     public void setAddress(String address) { this.address = address; }
22
23
24
25
26
27     public String getNumbers() { return numbers; }
28
29     public void setNumbers(String numbers) { this.numbers = numbers; }
30
31
32
33
34
35     public String getSpecialization() {
36         return specialization;
37     }
38     public void setSpecialization(String specialization) {
39         this.specialization = specialization;
40     }
41
42
43     public String getWorkTime() {
44         return WorkTime;
45     }
46     public void setWorkTime(String workTime) {
47         WorkTime = workTime;
48     }
49 }
50

```

```

package com.company;

public class WorkTime {

    public String MONDAY;
    public String TUESDAY;
    public String WEDNESDAY;
    public String THURSDAY;
    public String FRIDAY;
    public String SATURDAY;
    public String SUNDAY;

    public String getMONDAY() { return MONDAY; }
    public void setMONDAY(String MONDAY) { this.MONDAY = MONDAY + "MONDAY"+"\\n"; }

    public String getTUESDAY() { return TUESDAY; }
    public void setTUESDAY(String THUESDAY) { this.TUESDAY = THUESDAY + "THUESDAY"+"\\n"; }

    public String getWEDNESDAY() { return WEDNESDAY; }
    public void setWEDNESDAY(String WEDNESDAY) { this.WEDNESDAY = WEDNESDAY + "WEDNESDAY"+"\\n"; }

    public String getTHURSDAY() { return THURSDAY; }
    public void setTHURSDAY(String THURSDAY) { this.THURSDAY = THURSDAY + "THURSDAY"+"\\n"; }

    public String getFRIDAY() { return FRIDAY; }
    public void setFRIDAY(String FRIDAY) { this.FRIDAY = FRIDAY+"FRIDAY"+"\\n"; }

    public String getSATURDAY() { return SATURDAY; }
    public void setSATURDAY(String SATURDAY) { this.SATURDAY = SATURDAY+"SATURDAY"+"\\n"; }

    public String getSUNDAY() { return SUNDAY; }
    public void setSUNDAY(String SUNDAY) { this.SUNDAY = SUNDAY+ "SUNDAY"+"\\n"; }

}

```

3. Варіанти використання

Програма може використовуватись для тривалого зберігання об'єктів згідно з варіантом умови.

ВИСНОВКИ

У ході роботи я навчився працювати з компаратором, та використовувати Java Collections Framework та стандартні методи.