Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Лабораторна робота №2

Колекції. Множина HashSet. Асоціативні масиви Map.

Виконали

студенти групи ІА-23:

Волошин Вадім Вікторович

Воронюк Євгеній Владиславович

Савонік Назар Анатолійович

Перевірив:

Колеснік Валерій Миколайович

Київ 2023

Хід роботи:

1. Ознайомитись з javadoc для наступних інтерфейсів,класів та методів:

- Set
- HashSet
- Object.equals(), Object.hashCode()
- Map
- HashMap

2. Виконати завдання лабораторної роботи №10, замінивши списки List (ArrayList та LinkedList) на множини Set (HashSet). Проаналізувати предметну область та на власний розсуд додати функціональність, для реалізації якої використати Map (TreeMap або HashMap).

3. Відповісти на контрольні питання.

Company.java {}

```
import java.util.*;

public class Company {
    public static void main(String args[]) {
        System.out.println("Company name: GnomIT");
        System.out.println("Director name: Michael Reeves");
        Set set = new HashSet();
        set.add("MARKETING");
        set.add("LEGAL");
        set.add("HR");
        set.add("OPERATIONS");

        System.out.println(set);
    }
}
```

```
Company name: GnomIT
Director name: Michael Reeves
[OPERATIONS, LEGAL, MARKETING, HR]

Process finished with exit code 0
```

Department.java {} *(допоміжний клас)*

```
import java.util.List;

public class Department {
    private String DepName;
    private List<String> EmployeeList;

    public Department(String DepName, List<String> EmployeeList) {
        this.DepName = DepName;
        this.EmployeeList = EmployeeList;
    }
```

```
    public void printEmployeeInfo() {
        System.out.print(this.DepName + ": ");
        for (String dep : this.EmployeeList) {
            System.out.print(dep + ", ");
        }
        System.out.println();
    }
}
```

DepList.java {}

```
import java.util.List;
import java.util.ArrayList;
import java.util.Collections;

public class DepList {
    public static void main(String[] args) {
        //Creating departments
        String[] departments = {"HR", "OPERATIONS", "MARKETING", "LEGAL"};

        //HR
        List<String> DepEmployeeList = new ArrayList<>();
        String[] EmployeeList = {"Tommy Versace", "Catherine Jones", "Tom
Ford"};
        Collections.addAll(DepEmployeeList, EmployeeList);
        Department hr = new Department(departments[0], DepEmployeeList);
        System.out.println("Head of HR: " + EmployeeList[0]);
        hr.printEmployeeInfo();

        //operations
        DepEmployeeList = new ArrayList<>();
        EmployeeList = new String[]{"James Elliot", "John White", "James
Alvaro"};
        Collections.addAll(DepEmployeeList, EmployeeList);
        Department operations = new Department(departments[1],
DepEmployeeList);
        System.out.println("\nHead of Operations: " + EmployeeList[0]);
        operations.printEmployeeInfo();

        //marketing
        DepEmployeeList = new ArrayList<>();
        EmployeeList = new String[]{"Tom Jones", "Nancy Smith", "Frank
Anthony"};
        Collections.addAll(DepEmployeeList, EmployeeList);
        Department marketing = new Department(departments[2],
DepEmployeeList);
        System.out.println("\nHead of Marketing: " + EmployeeList[0]);
        marketing.printEmployeeInfo();

        //legal
        DepEmployeeList = new ArrayList<>();
        EmployeeList = new String[]{"Ryan Gosling", "Harry Major", "Ethan
Hardy"};
        Collections.addAll(DepEmployeeList, EmployeeList);
        Department legal = new Department(departments[2], DepEmployeeList);
        System.out.println("\nHead of Legal: " + EmployeeList[0]);
        legal.printEmployeeInfo();
    }
}
```

```
Head of HR: Tommy Versace
HR: Tommy Versace, Catherine Jones, Tom Ford,

Head of Operations: James Elliot
OPERATIONS: James Elliot, John White, James Alvaro,

Head of Marketing: Tom Jones
MARKETING: Tom Jones, Nancy Smith, Frank Anthony,

Head of Legal: Ryan Gosling
MARKETING: Ryan Gosling, Harry Major, Ethan Hardy,

Process finished with exit code 0
```

EmployeeList.java {} *(task 3 here)*

```
import java.util.*;

public class EmployeeList {
    public static void main(String[] args) {

        HashSet<Employee> employeeL = new HashSet<>();
        employeeL.add(new Employee("Tom Jones", 21000, "Marketing", true,
false));
        employeeL.add(new Employee("Tommy Versace", 19000, "HR", true,
false));
        employeeL.add(new Employee("Tom Ford", 17000, "HR", false, false));
        employeeL.add(new Employee("Ryan Gosling", 34000, "Legal", true,
false));
        employeeL.add(new Employee("Harry Major", 20000, "Legal", false,
false));
        employeeL.add(new Employee("Ethan Hardy", 30000, "Legal", false,
false));
        employeeL.add(new Employee("Nancy Smith", 15000, "Marketing", false,
false));
        employeeL.add(new Employee("Catherine Jones", 18000, "HR", false,
false));
        employeeL.add(new Employee("James Elliot", 33000, "Operations", true,
false));
        employeeL.add(new Employee("James Alvaro", 31000, "Operations",
false, false));
        employeeL.add(new Employee("John White", 35000, "Operations", false,
false));
        employeeL.add(new Employee("Frank Anthony", 22000, "Marketing",
false, false));
        employeeL.add(new Employee("Michael Reeves", 45000, "Head of
company", false, true));

        //TASK3
        System.out.println("List of employees: " + employeeL);
    }
}
class Employee implements Comparable<Employee> {
    String name;
    Integer salary;
```

```
    String department;
    Boolean isDirector;
    Boolean isHead;

    public Employee(String n, Integer salary, String d, Boolean isHead,
Boolean isDirector) {
        name = n;
        this.salary = salary;
        department = d;
        this.isHead = isHead;
        this.isDirector = isDirector;
    }

    public double getSalary() {
        return salary;
    }

    public Boolean getisDirector() {
        return isDirector;
    }

    public String toString() {
        return "(" + name + ", " + ", " + salary + ", " + department + ", " +
isHead + ", " + isDirector + ")";
    }


    @Override
    public int compareTo(Employee empl) {
        return this.name.compareTo(empl.name);
    }
}
```

```
List of employees:
[
(Catherine Jones, , 18000, HR, false, false),
(Nancy Smith, , 15000, Marketing, false, false),
(Ryan Gosling, , 34000, Legal, true, false),
(John White, , 35000, Operations, false, false),
(James Elliot, , 33000, Operations, true, false),
(Tommy Versace, , 19000, HR, true, false),
(Tom Ford, , 17000, HR, false, false),
(Frank Anthony, , 22000, Marketing, false, false),
(Harry Major, , 20000, Legal, false, false),
(Michael Reeves, , 45000, Head of company, false, true),
(Ethan Hardy, , 30000, Legal, false, false),
(Tom Jones, , 21000, Marketing, true, false),
(James Alvaro, , 31000, Operations, false, false)]

Process finished with exit code 0
```

Task1.java {}

```java
import java.util.*;

public class Task1 {
    public static void main(String[] args) {
        HashMap<String, String> map = new HashMap<>();
        map.put("21000", "Tom Jones");
        map.put("22000", "Frank Anthony");
        map.put("34000", "Ryan Gosling");
        map.put("18000", "Catherine Jones");
        map.put("33000", "James Elliot");
        map.put("19000", "Tommy Versace");
        map.put("17000", "Tom Ford");
        map.put("20000", "Harry Major");
        map.put("30000", "Ethan Hardy");
        map.put("31000", "James Alvaro");
        map.put("35000", "John White");
        map.put("15000", "Nancy Smith");
        map.put("45000", "Michael Reeves");

        TreeMap<String, String> sorted = new TreeMap<>();

        sorted.putAll(map);
        for(Map.Entry<String, String> entry : map.entrySet()) {
            System.out.println("Salary: " + entry.getKey() + "; Name: " +
entry.getValue());
        }
    }
}
```

```
Salary: 18000; Name: Catherine Jones
Salary: 19000; Name: Tommy Versace
Salary: 20000; Name: Harry Major
Salary: 21000; Name: Tom Jones
Salary: 17000; Name: Tom Ford
Salary: 15000; Name: Nancy Smith
Salary: 22000; Name: Frank Anthony
Salary: 34000; Name: Ryan Gosling
Salary: 35000; Name: John White
Salary: 33000; Name: James Elliot
Salary: 30000; Name: Ethan Hardy
Salary: 31000; Name: James Alvaro
Salary: 45000; Name: Michael Reeves

Process finished with exit code 0
```

Task2.java {}

```java
import java.util.*;

public class Task2 {

    public static void main(String[] args) {
        HashMap<Integer, String> map = new HashMap<>();
        map.put(21000, "Tom Jones, Marketing");
        map.put(22000, "Frank Anthony, Marketing");
        map.put(34000, "Ryan Gosling, Legal");
        map.put(18000, "Catherine Jones, HR");
        map.put(33000, "James Elliot, Operations");
        map.put(19000, "Tommy Versace, HR");
        map.put(17000, "Tom Ford, HR");
        map.put(20000, "Harry Major, Legal");
        map.put(30000, "Ethan Hardy, Legal");
        map.put(31000, "James Alvaro, Operations");
        map.put(35000, "John White, Operations");
        map.put(15000, "Nancy Smith, Marketing");
        TreeMap<Integer, String> sorted = new TreeMap<>();

        sorted.putAll(map);
        for (Map.Entry<Integer, String> entry : sorted.entrySet())
            System.out.println("Salary: " + entry.getKey() + "; Name and
department: " + entry.getValue());
    }
}
```

```
Salary: 15000; Name and department: Nancy Smith, Marketing
Salary: 17000; Name and department: Tom Ford, HR
Salary: 18000; Name and department: Catherine Jones, HR
Salary: 19000; Name and department: Tommy Versace, HR
Salary: 20000; Name and department: Harry Major, Legal
Salary: 21000; Name and department: Tom Jones, Marketing
Salary: 22000; Name and department: Frank Anthony, Marketing
Salary: 30000; Name and department: Ethan Hardy, Legal
Salary: 31000; Name and department: James Alvaro, Operations
Salary: 33000; Name and department: James Elliot, Operations
Salary: 34000; Name and department: Ryan Gosling, Legal
Salary: 35000; Name and department: John White, Operations

Process finished with exit code 0
```

Висновок: у даній лабораторній роботі ми повторили завдання лабораторної роботи №10, але замінили списки List (ArrayList та LinkedList) на множини Set (HashSet) і використали Map для додавання нової функціональності. Ми також порівняли інтерфейси Set, Collection, List, HashSet від TreeSet теоретично та отримали відомості про хеш-функцію, хеш, хеш-код.