



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 №11**  
**«Основи програмування»**  
**Тема: “Колекції. Множина TreeSet”**

Виконали:

Студенти групи ІА-24

Любченко.І.М

Гуменюк.К.Е

Тильна.М.С

### Хід роботи:

1. Ознайомитись з javadoc для наступних інтерфейсів та класів:

- Set
- TreeSet
- Comparable
- Comparator
- SortedSet
- NavigableSet

2. Виконати завдання попередньої лабораторної роботи, замінивши списки List (ArrayList та LinkedList) на множини Set (TreeSet). При цьому необхідно щонайменше один раз використати Comparable та щонайменше один раз – Comparator.

3. Відповісти на контрольні питання.

## Company.java

```
import java.util.Set;
import java.util.TreeSet;

public class Company {
    private final String name;
    private final Employer head;
    private final Set<Department> departments = new TreeSet<>();

    Company(String name, Employer head, Set<Department> departments) {
        this.name = name;
        this.head = head;
    }

    Company(String name, Employer head) {
        this.name = name;
        this.head = head;
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
    }

    public Employer getHead() {
        return head;
    }

    public void setHead(Employer head) {
    }

    public Set<Department> getDepartments() {
        return departments;
    }

    @Override
    public String toString() {
        return "name: " + name;
    }

    public void setDepartments(Set<Department> departments) {
```

```
}  
}
```

## Department.java

```
import java.util.Set;  
import java.util.TreeSet;  
  
public class Department implements Comparable<Department>{  
    private final String name;  
    private final Employer manager;  
    private final Set<Employer> employers= new TreeSet<>();  
  
    Department(String name, Employer manager, Set<Employer> employers){  
        this.name = name;  
        this.manager = manager;  
    }  
  
    Department(String name, Employer manager){  
        this.name = name;  
        this.manager = manager;  
    }  
  
    public String getName(){  
        return this.name;  
    }  
  
    public Employer getManager(){  
        return manager;  
    }  
  
    public Set<Employer> getEmployers(){  
        return this.employers;  
    }  
  
    @Override  
    public String toString() {  
        return "Department name: '" + name + '\'';  
    }  
  
    @Override  
    public int compareTo(Department o) {  
        if(this.employers.size() == o.employers.size()){  
            return 0;  
        } else if(this.employers.size() < o.employers.size()){  
            return -1;  
        } else {  
            return 1;  
        }  
    }  
}
```

## Employer.java

```
public class Employer implements Comparable<Employer>{
    final private String name;
    final private String surname;
    final private int salary;

    Employer(String name, String surname, int salary){
        this.name = name;
        this.surname = surname;
        this.salary = salary;
    }

    Employer(String name, String surname, int salary, Department
department){
        this.name = name;
        this.surname = surname;
        this.salary = salary;
        department.getEmployers().add(Employer.this);
    }

    public String getName(){
        return this.name;
    }

    public String getSurname(){
        return surname;
    }

    public int getSalary(){
        return salary;
    }

    @Override
    public String toString() {
        return name + " " + surname ;
    }

    @Override
    public int compareTo(Employer o) {
        return Integer.compare(this.salary, o.salary);
    }
}
```

## Main.java

```
import java.util.*;

public class Main {
    private static int MaxSalary(Company company){
        int maxSalary = 0;
        for(Iterator iterator = EmployersList(company).iterator();
iterator.hasNext();){
```

```

        Object employer = iterator.next();
        if(employer instanceof Employer currentEmployer) {
            int current = currentEmployer.getSalary();
            if(current>maxSalary) {maxSalary=current;}
        }
    }
    return maxSalary;
}

private static Department strangeDepartment(Company company){
    for(Department department : company.getDepartments()){
        for(Employer employer : department.getEmployers()){
            if(employer.getSalary()>department.getManager().getSalary()){
                return department;
            }
        }
    }
    return null;
}

private static Set<Employer> EmployersList(Company company){
    Set<Employer> EmployersList = new TreeSet<>();
    EmployersList.add(company.getHead());
    for(Iterator<Department> iterator =
company.getDepartments().iterator(); iterator.hasNext();){
        Department current = iterator.next();
        EmployersList.add(current.getManager());
        EmployersList.addAll(current.getEmployers());
    }
    return EmployersList;
}

private static void zavd1(Company company){
    System.out.println(MaxSalary(company));
}
private static void zavd2(Company company){
    System.out.println(strangeDepartment(company).toString());
}
private static void zavd3(Company company){
    for (Employer i: EmployersList(company)) {
        System.out.print(i.toString()+" ");
    }
}

public static void main(String[] args) {
    Company Sony = new Company("xbox",new Employer("Ivan",
"Zelynskii",10000));
    Department Games = new Department("Games", new Employer("Nazarii",
"Nover", 5000));
    Employer Bob = new Employer("Kyril","Humenyuk", 1000, Games);
    Employer Jesse = new Employer("Masha","Tylna", 1000, Games);
    Employer James = new Employer("Ioann", "Liubchenko", 1000, Games);

    Department Movies = new Department("Movies", new
Employer("Oleksandr", "Zelenskyi", 9000));
    Movies.getEmployers().add(new Employer("Masha","Tylna", 10000));
    Movies.getEmployers().add(new Employer("Kyril","Humenyuk", 10000));
    Movies.getEmployers().add(new Employer("Ioann", "Liubchenko",
10000));
    Movies.getEmployers().add(new Employer("Oleksandr", "Zelenskyi",

```

```
200000));  
  
Sony.getDepartments().add(Movies);  
Sony.getDepartments().add(Games);  
  
zavd1(xbox);  
zavd2(xbox);  
zavd3(xbox);  
  
}  
}
```

## **Висновок:**

Ми Ознайомились з javadoc для наступних інтерфейсів та класів: Set, TreeSet, Comparable Comparator, SortedSet, NavigableSet. Переробили минулу лабораторну, змінивши списки List ArrayList - LinkedList на множини Set-TreeSet. Зрозуміли як використовувати Comparable та Comparator. Опрацювали контрольні запитання і важливі моменти. А також попрацювали в команді.