МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Кафедра інформаційних систем та технологій

Лабораторна робота №12

З дисципліни «Основи програмування»
Тема: Колекції. Множина HashSet. Асоціативні масиви Map.

Виконали: Гуменюк К.Е.
Тильна М.С. Любченко І.М.

Перевірив: Колеснік В.М.

Київ-2023

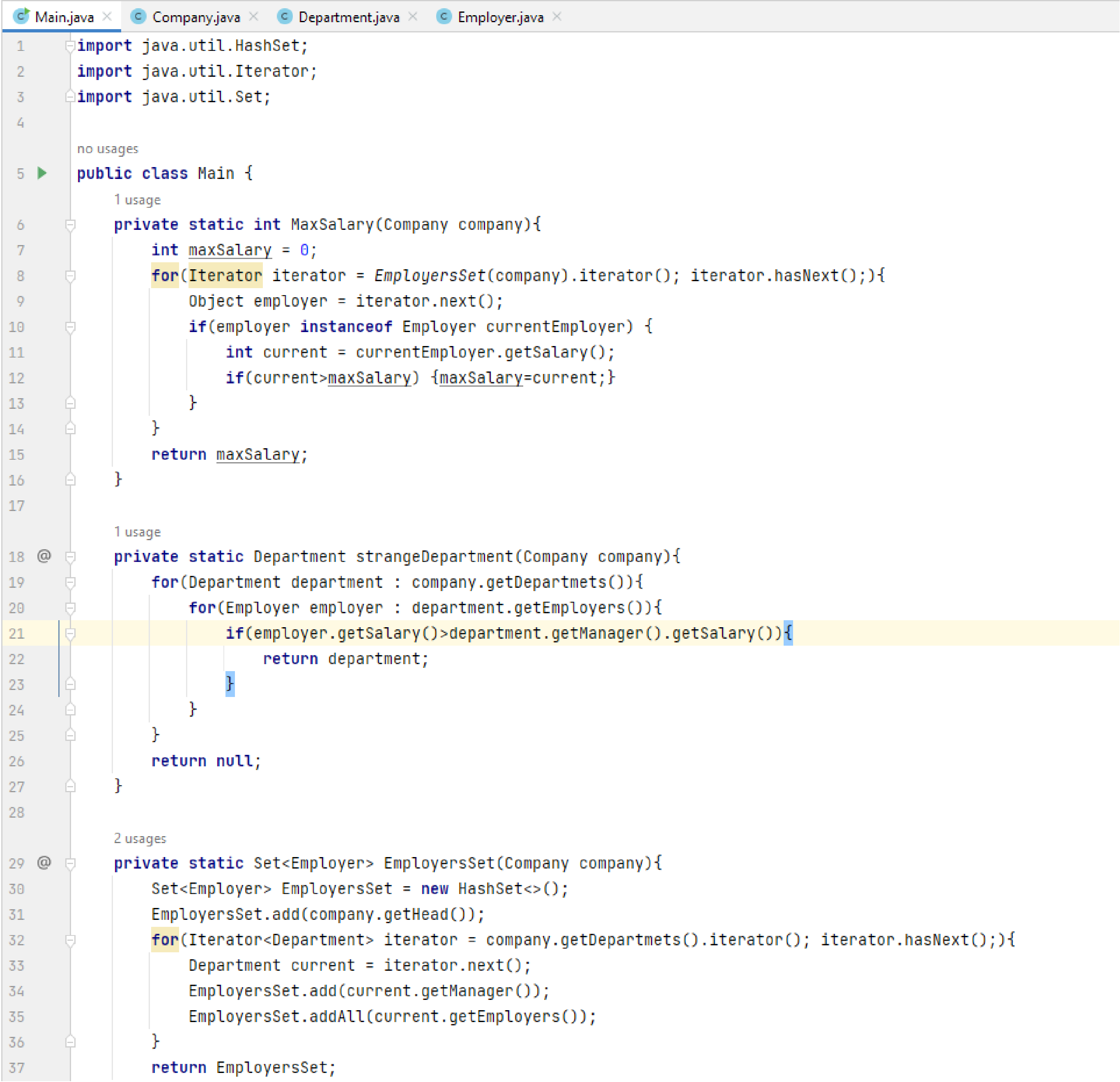
Тема: Колекції. Множина HashSet. Асоціативні масиви Map.

Хід роботи

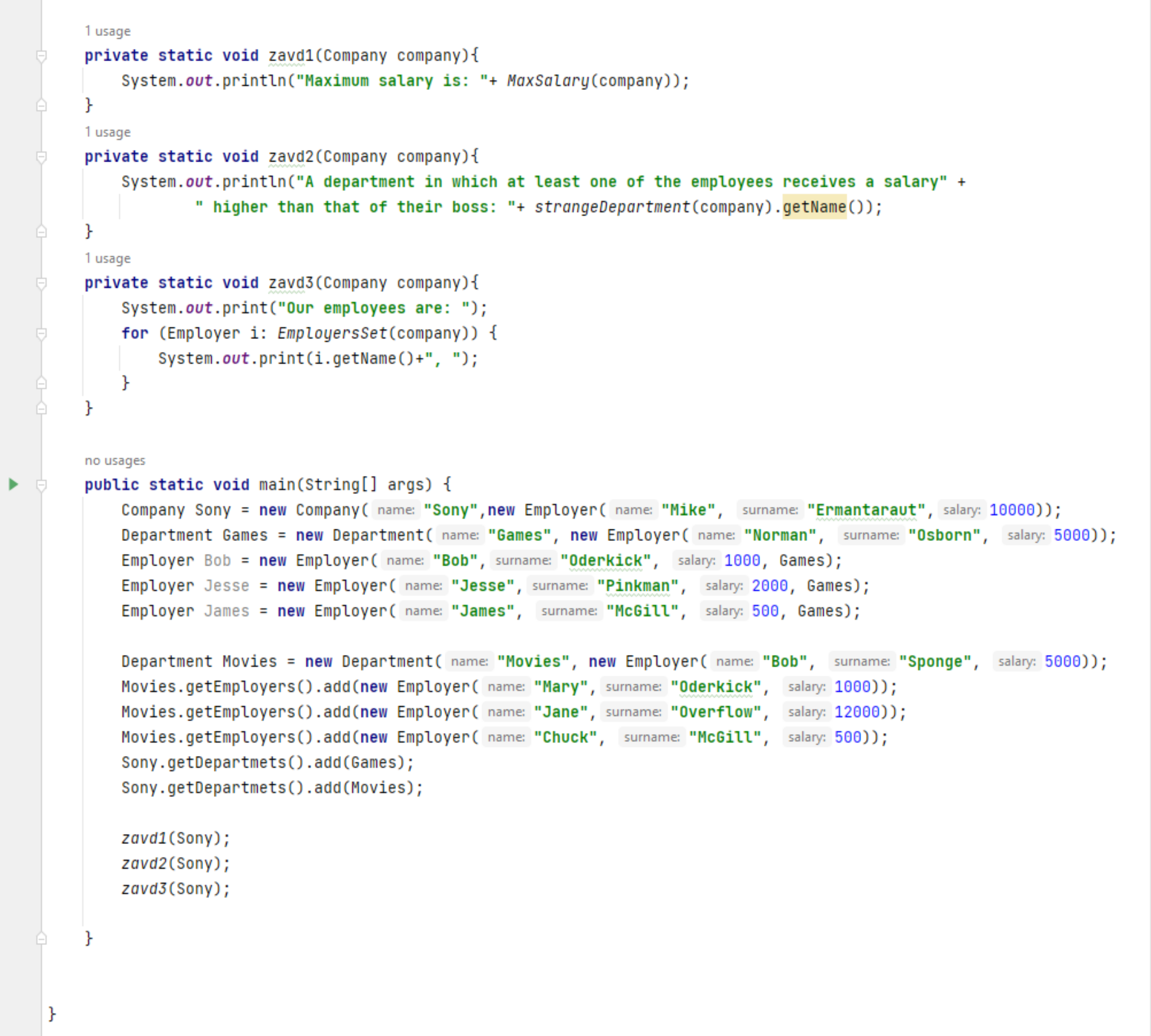
1. Ознайомитись з javadoc для наступних інтерфейсів,класів та методів:
   - Set
   - HashSet
   - Object.equals(), Object.hashCode()
   - Map
   - HashMap

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).

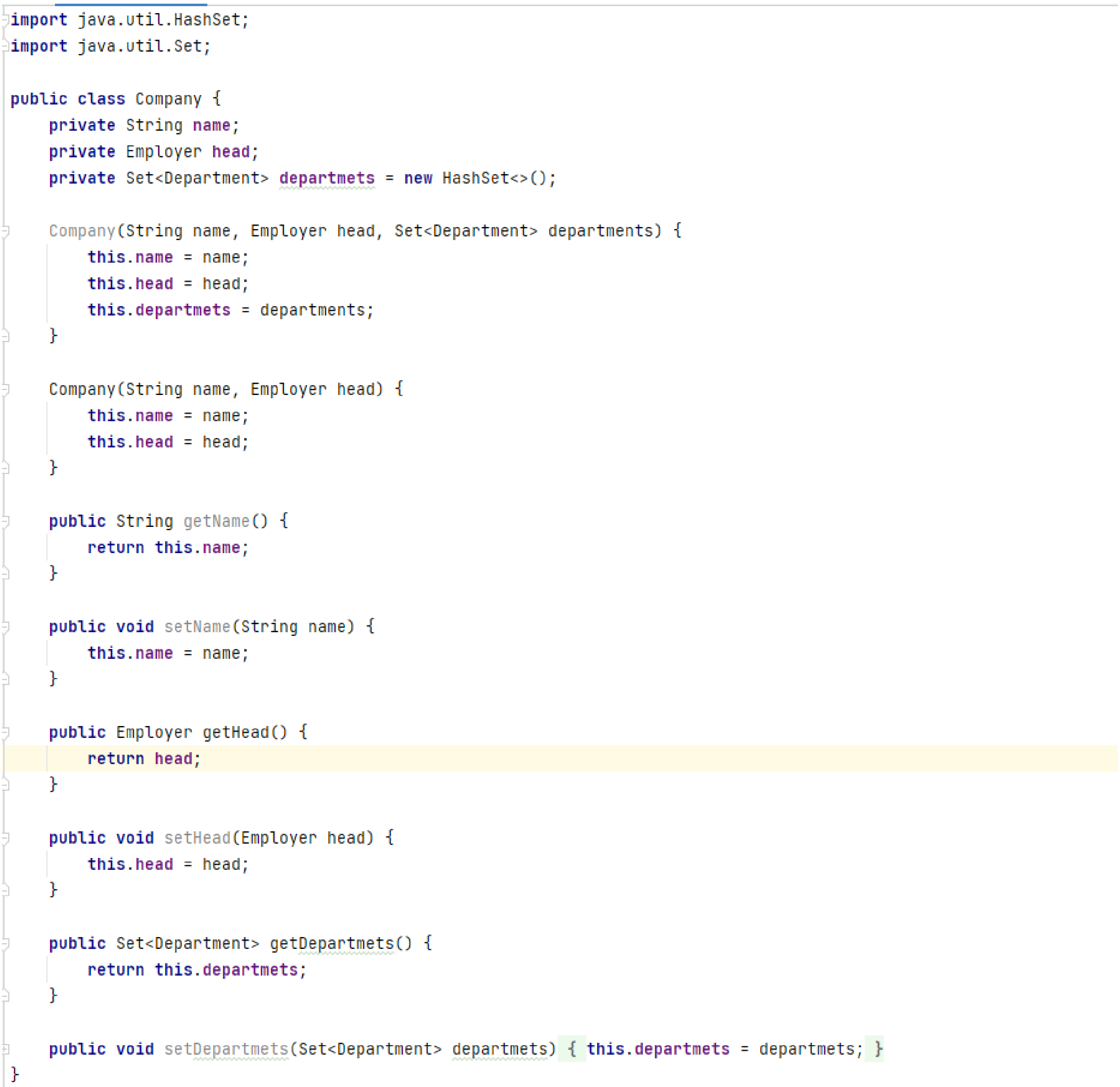
3. Відповісти на контрольні питання.



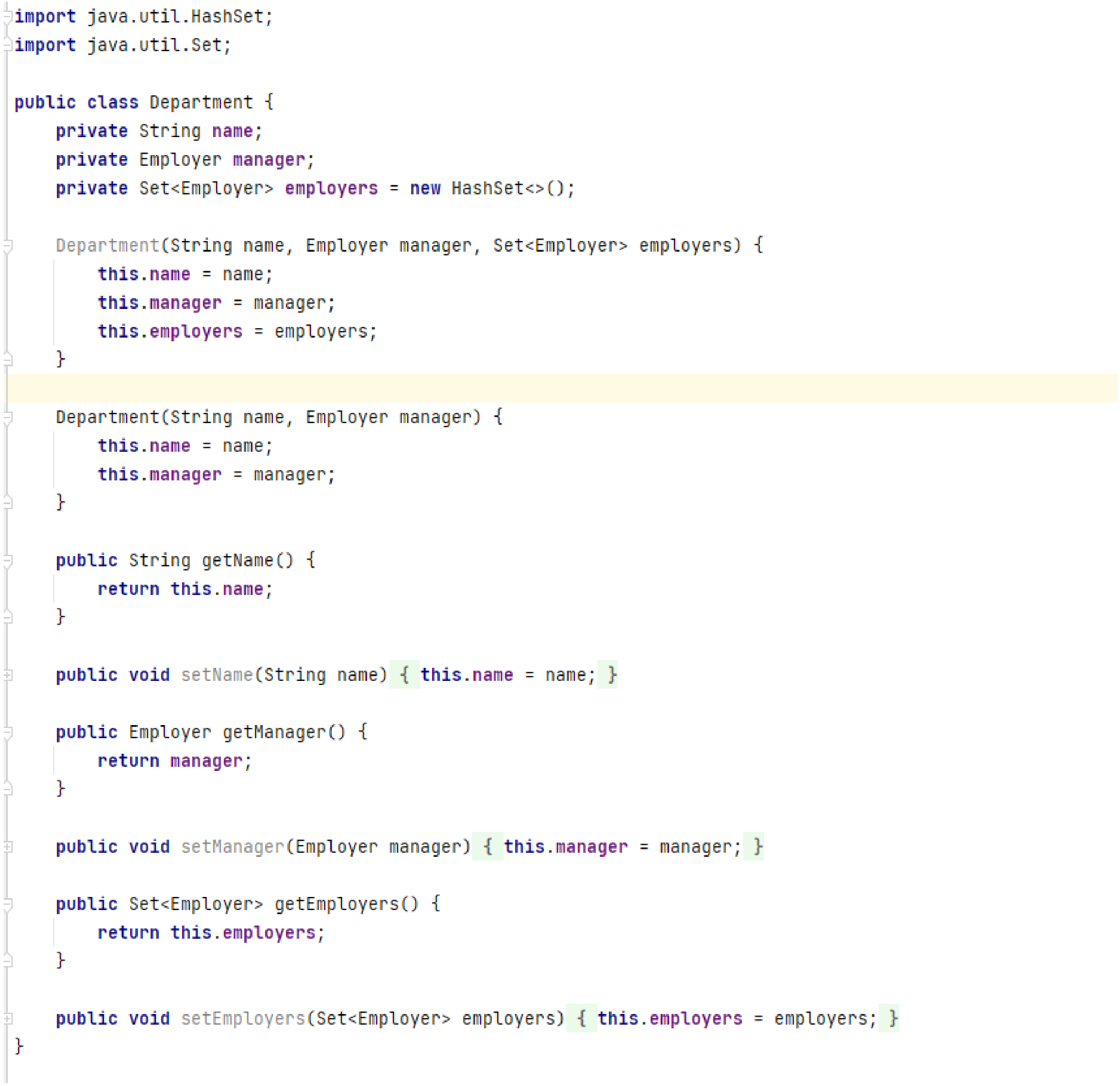
```java
import java.util.HashSet;
import java.util.Iterator;
import java.util.Set;

public class Main {
    private static int MaxSalary(Company company){
        int maxSalary = 0;
        for(Iterator iterator = EmployersSet(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.getDepartmets()){
            for(Employer employer : department.getEmployers()){
                if(employer.getSalary()>department.getManager().getSalary()){
                    return department;
                }
            }
        }
        return null;
    }

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



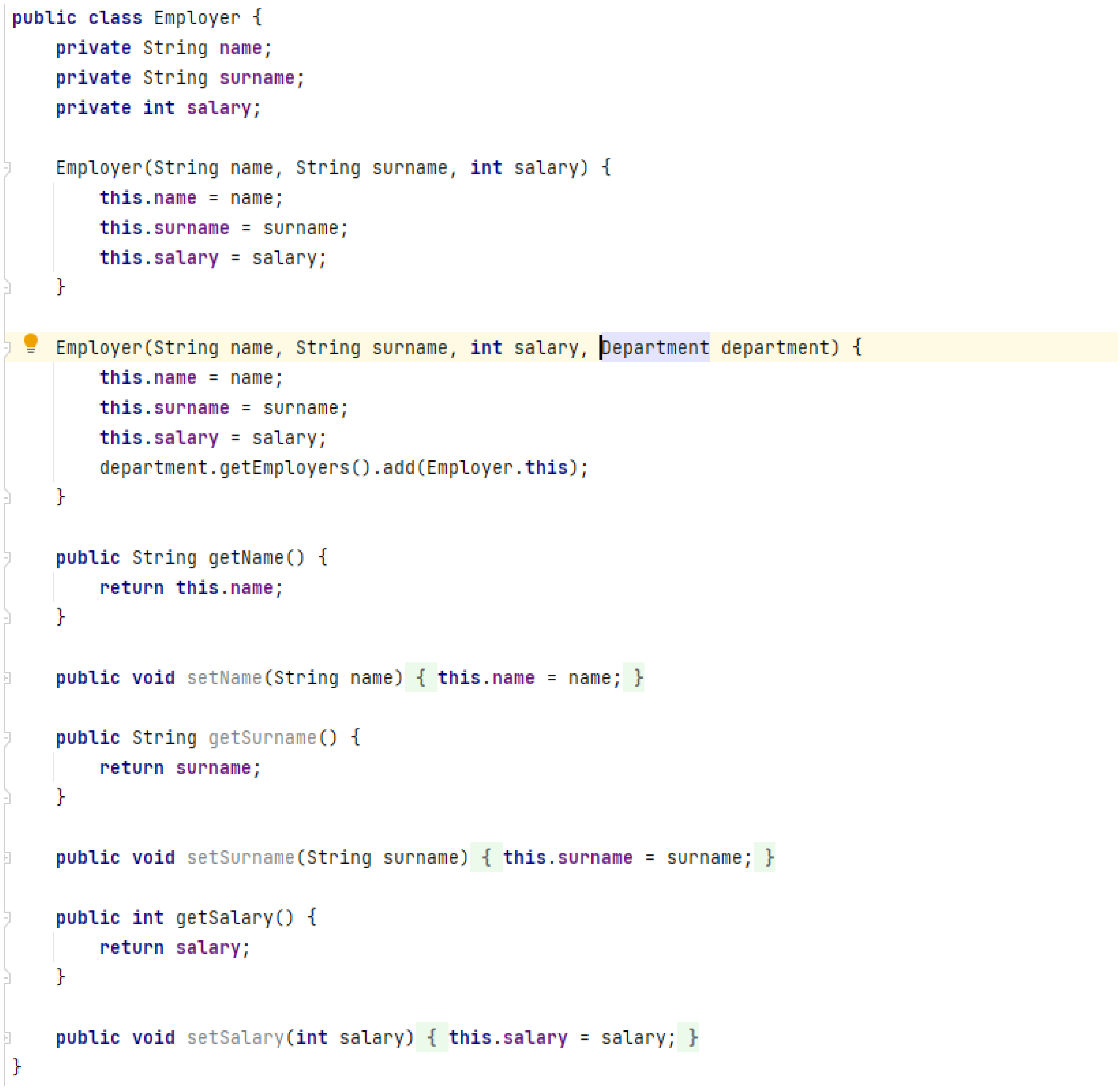
```
    1 usage
    private static void zavd1(Company company){
        System.out.println("Maximum salary is: "+ MaxSalary(company));
    }
    1 usage
    private static void zavd2(Company company){
        System.out.println("A department in which at least one of the employees receives a salary" +
                " higher than that of their boss: "+ strangeDepartment(company).getName());
    }
    1 usage
    private static void zavd3(Company company){
        System.out.print("Our employees are: ");
        for (Employer i: EmployersSet(company)) {
            System.out.print(i.getName()+", ");
        }
    }

    no usages
    public static void main(String[] args) {
        Company Sony = new Company( name: "Sony",new Employer( name: "Mike",  surname: "Ermantaraut", salary: 10000));
        Department Games = new Department( name: "Games", new Employer( name: "Norman",  surname: "Osborn",  salary: 5000));
        Employer Bob = new Employer( name: "Bob", surname: "Oderkick",  salary: 1000, Games);
        Employer Jesse = new Employer( name: "Jesse", surname: "Pinkman",  salary: 2000, Games);
        Employer James = new Employer( name: "James",  surname: "McGill",  salary: 500, Games);

        Department Movies = new Department( name: "Movies", new Employer( name: "Bob",  surname: "Sponge",  salary: 5000));
        Movies.getEmployers().add(new Employer( name: "Mary", surname: "Oderkick",  salary: 1000));
        Movies.getEmployers().add(new Employer( name: "Jane", surname: "Overflow",  salary: 12000));
        Movies.getEmployers().add(new Employer( name: "Chuck",  surname: "McGill",  salary: 500));
        Sony.getDepartmets().add(Games);
        Sony.getDepartmets().add(Movies);

        zavd1(Sony);
        zavd2(Sony);
        zavd3(Sony);

    }


}
```



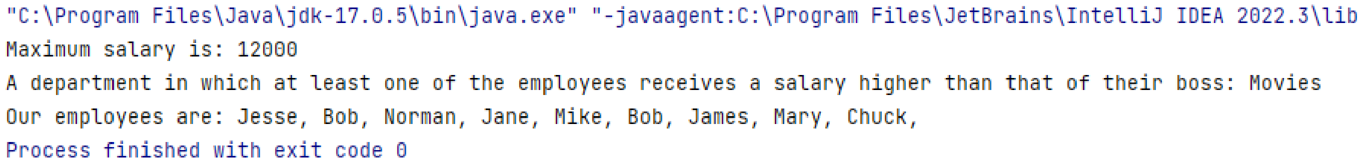
```
import java.util.HashSet;
import java.util.Set;

public class Company {
    private String name;
    private Employer head;
    private Set<Department> departmets = new HashSet<>();

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

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

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

    public void setName(String name) {
        this.name = name;
    }

    public Employer getHead() {
        return head;
    }

    public void setHead(Employer head) {
        this.head = head;
    }

    public Set<Department> getDepartmets() {
        return this.departmets;
    }

    public void setDepartmets(Set<Department> departmets) { this.departmets = departmets; }
}
```



```java
import java.util.HashSet;
import java.util.Set;

public class Department {
    private String name;
    private Employer manager;
    private Set<Employer> employers = new HashSet<>();

    Department(String name, Employer manager, Set<Employer> employers) {
        this.name = name;
        this.manager = manager;
        this.employers = employers;
    }

    Department(String name, Employer manager) {
        this.name = name;
        this.manager = manager;
    }

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

    public void setName(String name) { this.name = name; }

    public Employer getManager() {
        return manager;
    }

    public void setManager(Employer manager) { this.manager = manager; }

    public Set<Employer> getEmployers() {
        return this.employers;
    }

    public void setEmployers(Set<Employer> employers) { this.employers = employers; }
}
```



```java
public class Employer {
    private String name;
    private String surname;
    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 void setName(String name) { this.name = name; }

    public String getSurname() {
        return surname;
    }

    public void setSurname(String surname) { this.surname = surname; }

    public int getSalary() {
        return salary;
    }

    public void setSalary(int salary) { this.salary = salary; }
}
```



```
"C:\Program Files\Java\jdk-17.0.5\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2022.3\lib
Maximum salary is: 12000
A department in which at least one of the employees receives a salary higher than that of their boss: Movies
Our employees are: Jesse, Bob, Norman, Jane, Mike, Bob, James, Mary, Chuck,
Process finished with exit code 0
```

Висновки

Під час цієї лабораторної роботи ми ознайомитись з javadoc для наступних інтерфейсів,класів та методів:

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

та використали їх на практиці