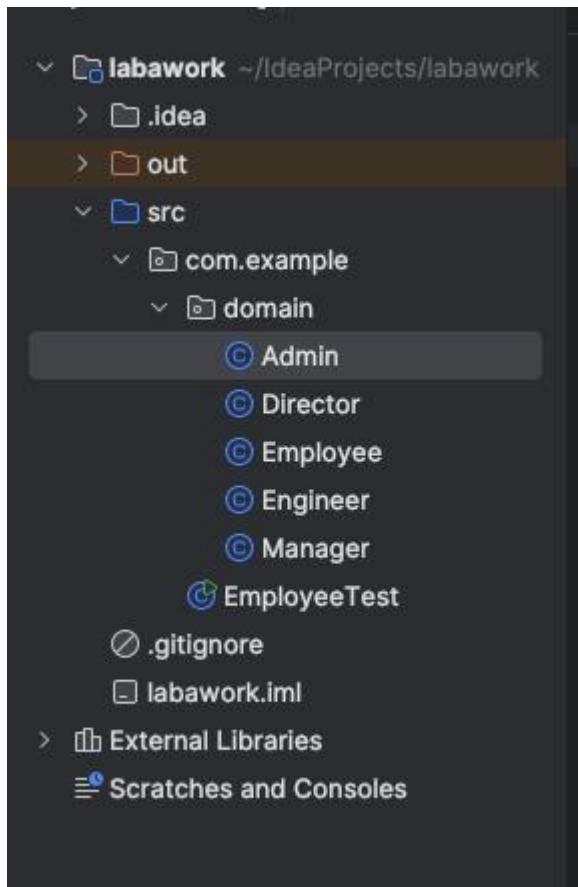


Лабораторная работа №3
Тарвердян Сима Арменовна
Группа 253-324

Код:



```
1 package com.example.domain;
2
3 public class Admin extends Employee { no usages
4
5     public Admin(int empId, String name, String ssn, double salary) { no usages
6         super(empId, name, ssn, salary);
7     }
8 }
```

The screenshot shows the code editor with the 'Admin.java' file open. The code defines a class 'Admin' that extends 'Employee'. It includes a constructor that takes 'empId', 'name', 'ssn', and 'salary' as parameters and calls the superclass's constructor. The code editor interface is visible on the left, showing the project structure and other files like 'EmployeeTest', '.gitignore', and 'labawork.iml'.

The screenshot shows the IntelliJ IDEA interface with the project structure on the left and the code editor on the right. The code editor displays the `Director` class from the `com.example.domain` package. The `Director` class extends `Manager` and has a constructor that takes `empId`, `name`, `ssn`, `salary`, and `deptName`. It also has a private field `budget` and a public method `getBudget()` that returns the budget.

```
1 package com.example.domain;
2
3 public class Director extends Manager { 3 usages
4
5     private double budget; 2 usages
6
7     public Director(int empId, String name, String ssn, double salary, 1 usage
8         String deptName, double budget) {
9         super(empId, name, ssn, salary, deptName);
10        this.budget = budget;
11    }
12
13    public double getBudget() { no usages
14        return budget;
15    }
16}
17
```

The screenshot shows the IntelliJ IDEA interface with the project structure on the left and the code editor on the right. The code editor displays the `Employee` class from the `com.example.domain` package. The `Employee` class has private fields `empId`, `name`, `ssn`, and `salary`. It has a constructor that initializes these fields and a method `raiseSalary(double increase)` that increases the salary by the specified amount if the increase is greater than zero. It also has methods to get the employee ID and name.

```
1 package com.example.domain;
2
3 public class Employee { 5 usages 4 inheritors
4
5     private int empId; 2 usages
6     private String name; 3 usages
7     private String ssn; 2 usages
8     private double salary; 3 usages
9
10    public Employee(int empId, String name, String ssn, double salary) { 3 usages
11        this.empId = empId;
12        this.name = name;
13        this.ssn = ssn;
14        this.salary = salary;
15    }
16
17    public void setName(String name) { no usages
18        this.name = name;
19    }
20
21    public void raiseSalary(double increase) { no usages
22        if (increase > 0) {
23            salary += increase;
24        }
25    }
26
27    public int getEmpId() { 1 usage
28        return empId;
29    }
30
31    public String getName() { 1 usage
32        return name;
33    }
34}
```

The screenshot shows the IntelliJ IDEA interface with the Employee class selected in the project tree. The code editor displays the following Java code:

```
public class Employee { 5 usages 4 inheritors
    public void setName(String name) { no usages
        this.name = name;
    }

    public void raiseSalary(double increase) { no usages
        if (increase > 0) {
            salary += increase;
        }
    }

    public int getEmpId() { 1 usage
        return empId;
    }

    public String getName() { 1 usage
        return name;
    }

    public String getSsn() { 1 usage
        return ssn;
    }

    public double getSalary() { 1 usage
        return salary;
    }
}
```

The screenshot shows the IntelliJ IDEA interface with the Engineer class selected in the project tree. The code editor displays the following Java code:

```
package com.example.domain;
public class Engineer extends Employee { no usages
    public Engineer(int empId, String name, String ssn, double salary) { no usages
        super(empId, name, ssn, salary);
    }
}
```

The screenshot shows the IntelliJ IDEA interface with the Manager class selected in the project tree. The code editor displays the following Java code:

```
package com.example.domain;
public class Manager extends Employee { no usages 1 inheritor
    private String deptName; 2 usages
    public Manager(int empId, String name, String ssn, double salary, String deptName) { no usages
        super(empId, name, ssn, salary);
        this.deptName = deptName;
    }

    public String getDeptName() { no usages
        return deptName;
    }
}
```

The screenshot shows two code editors in IntelliJ IDEA. Both editors have the same file open: `EmployeeTest.java`. The top editor has the code starting from line 9, and the bottom editor has the full code from line 9 to line 48.

```
1 package com.example;
2
3 import com.example.domain.Admin;
4 import com.example.domain.Director;
5 import com.example.domain.Engineer;
6 import com.example.domain.Manager;
7 import com.example.domain.Employee;
8
9 public class EmployeeTest {
10
11     public static void main(String[] args) {
12
13         Engineer eng = new Engineer(
14             empId: 101, name: "Jane Smith", ssn: "012-34-5678", salary: 120_345.27);
15
16         Manager mgr = new Manager(
17             empId: 207, name: "Barbara Johnson", ssn: "054-12-2367", salary: 109_501.36, deptName: "US Marketing");
18
19         Admin adm = new Admin(
20             empId: 304, name: "Bill Munroe", ssn: "108-23-2367", salary: 75_002.34);
21
22         Director dir = new Director(
23             empId: 12, name: "Susan Wheeler", ssn: "099-45-2340", salary: 120_567.36,
24             deptName: "Global Marketing", budget: 1_000_000.00);
25
26         printEmployee(eng);
27         printEmployee(mgr);
28         printEmployee(adm);
29         printEmployee(dir);
30     }
31 }
32
33 private static void printEmployee(Employee emp) { 4 usages
34     System.out.println("Employee ID: " + emp.getEmpId());
35     System.out.println("Employee Name: " + emp.getName());
36     System.out.println("Employee Soc Sec #: " + emp.getSsn());
37     System.out.println("Employee salary: " + emp.getSalary());
38     System.out.println("-----");
39 }
40 }
```

```
/Users/twers1/Library/Java/JavaVirtualMachines/openjdk-25.0.1/Contents/Home/bin/java -javaagent:/Ap  
Employee ID: 101  
Employee Name: Jane Smith  
Employee Soc Sec #: 012-34-5678  
Employee salary: 120345.27  
-----  
Employee ID: 207  
Employee Name: Barbara Johnson  
Employee Soc Sec #: 054-12-2367  
Employee salary: 109501.36  
-----  
Employee ID: 304  
Employee Name: Bill Munroe  
Employee Soc Sec #: 108-23-2367  
Employee salary: 75002.34  
-----  
Employee ID: 12  
Employee Name: Susan Wheeler  
Employee Soc Sec #: 099-45-2340  
Employee salary: 120567.36  
-----  
Process finished with exit code 0
```