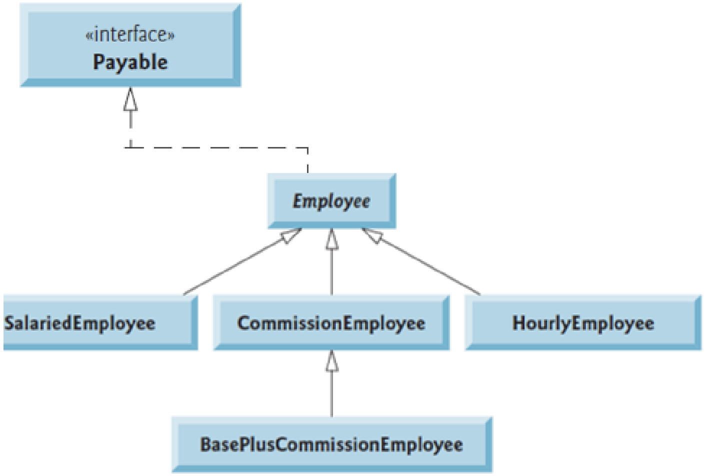
ADNAN MENDERES UNIVERSITY
CSE 203 Object-Oriented Programming

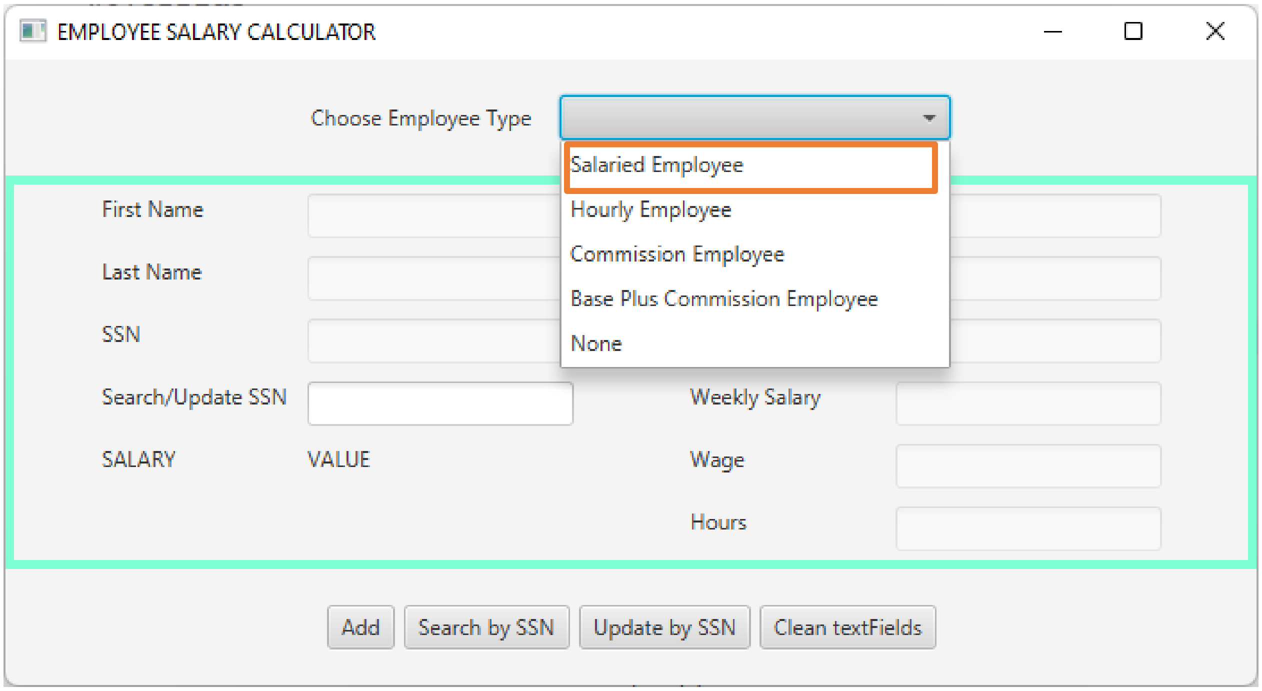
Project

- You should submit **whole java project file in zipped format** and **project report,** named project file as ***NameSurname_studentNo_Project***. Project report should be in .doc, .docx or .pdf format. Name of one student is enough for naming project
- Do your homeworks in **ECLIPSE IDE**.
- Do not use Turkish Characters(ç,ğ,ı,ö,ş,ü) for naming project, methods, classes.
- Late submissions are not allowed.
- Group working is allowed **up to 4 members**. **Indicate all project members's student No-Name surname in the report.**
- Copy homework will be evaluated as 0.
- Use Google Classroom for your questions.

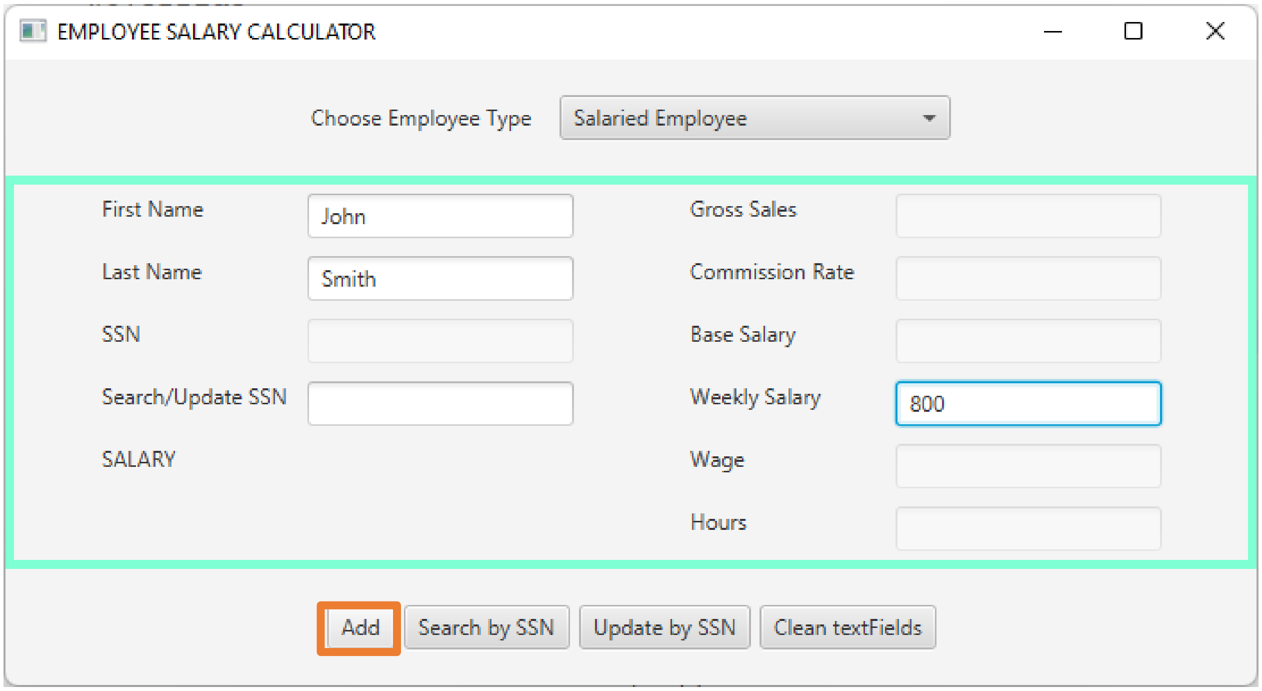
Problem Description:
CEO of Factory X make a request to your software company about design a program with user interface. The program should calculate Factory X's payment amount which will be made for all type of employees. Create **Payable** (interface), **Employee**, **SalariedEmployee**, **HourlyEmployee**, **CommissionEmployee**, **BasePlusCommissionEmployee,** and **Test (main)** class according to requirements given in LAB 5.
NOTE: Do not use invoice class from LAB 5, it is not needed. You may use other classes from LAB 5. If necessary, make some changes to provide true encapsulation (design all data fields as private and reach them just using get/set methods) and inheritance practice in your implementation.

Hierarchy between classes:





SALARY VALUE



Choose Employee Type: Salaried Employee

First Name: John

Last Name: Smith

Weekly Salary: 800

ADNAN MENDERES UNIVERSITY
CSE 203 Object-Oriented Programming

1- Your program should store, add, update and search salary information of all type employees as shown in figures above. Use **random access file(.dat) or text file (.txt)** for reading and writing an employee information.

- Program should perform **Add**, **Search by SSN**, **Update by SSN CleanTextFields** operations.
- SSN text field should be "read only" and **SSN** should be set in the code for each record, do not take from user.
- When program starts, read all records from random access file or text file and save each record in Employee []. (Employee array)
- Update both Employee [] and random access file or text file when **add** and **update** operations happen.
- **Add:** Select employee type from dropdown list. According to your selection, related text fields are enabled, unrelated text fields are disabled (Search/Update SSN text field always enabled). Enter employee information into enabled text fields, then press "add button". Your employee record will be written into file. Also add that record into your Employee[]
- **Search by SSN:** Enter SSN of employee whose information you want to reach, into "Search/Update SSN text field" . Press "Search by SSN button". Then all information related to the employee will be displayed, such as employee type, first name, last name, SSN, salary etc..
- **Update by SSN:** Enter SSN of employee whose information you want to update, into "Search/Update SSN text field". Enter other fields you want to update, then press "Update by SSN button**".** Employee record which is placed in file should be updated. Also, Employee[] should be updated.
- **Clean TextFields:** Cleans all text fields.

2- Submit a report (word or pdf) file that explains following questions with a cover page that contains your name and number. **In the report:**

- Indicate all project member's student No- Name surname in the report.
- Describe the problem including major steps, functions for solving the problem and input/output in your own words.
- Draw the UML diagram of your code.
- Describe how you test this program.