Task 06.02.2025 – Employee Management System

A company manages its employees in a hierarchical structure. There is a base class Employee, from which different specializations inherit:

- **Manager** → Responsible for a budget
- **Developer** → Has a programming language as a specialization
- **Intern** → Has a training year

The goal is to develop an employee management system that utilizes polymorphism. Methods will be overridden and overloaded, and instanceof will be used to ensure that specialized methods from subclasses can be accessed.

1. Task: Employee Base Class

- **Requirements:**
  - The class Employee should be abstract.
  - It contains general attributes for all employees:
    - protected String name
    - protected double salary
    - protected String department
  - Create a constructor that initializes all attributes.
  - It contains the following methods:
    - public void display() → Outputs general information.
    - public abstract void work(); → Must be overridden by subclasses.
    - Overloaded method: public void display(boolean detailed) → Outputs more information depending on the parameter.

2. Task: Subclasses Manager, Developer, Intern

- **Class Manager (inherits from Employee)**

Additional Attributes:

- private double budget;

Methods:

- Override the constructor → Use super() and extend it.
- Override work() → "Manager manages a budget of XYZ EUR"
- Override display() to also show the budget.
- Additional method:
    - public void calcPaycheck(Employee e) →
        - Depending on the employee type (Manager, Developer, or Intern), print their salary, name, and job title to the console.
- Consideration:
    - What additional steps are needed to access the name of the employee if it were private?

🔹 **Class Developer (inherits from Employee)**

Additional Attributes:

- private String programmingLanguage;

Methods:

- Override the constructor → Use super() and extend it.
- Override work() → "Developer is coding in XYZ"
- Override display() → Include the programming language.
- Additional method:
    - public void debug(String project) → "Developer is debugging project XYZ"

🔹 **Class Intern (inherits from Employee)**

Additional Attributes:

- private int trainingYear;

Methods:

- Override the constructor → Use super() and extend it.
- Override work() → "Intern is learning new tasks"
- Override display() → Include the training year.
- Additional method:

- public void learn() → "Intern is attending a seminar"

3. Task: Managing Employees

Create a Company class with a main() method:

- Create an array containing mixed Employee objects (Manager, Developer, Intern).
- Iterate over the array using a loop.
- Call display() and work() on each object.
- Check if the object is a Manager, Developer, or Intern using instanceof.
  - If so, call the appropriate specific method (e.g., Intern.learn()).
- Create a second loop and pass all objects in the array through a Manager's calcPaycheck(Employee e) method, which prints the respective salary details again.