National University of Computer and Emerging Sciences

Software Construction and Development

Lab Manual 1

Muhammad Hassan Raza

Fall 2024

Department of Software Engineering
FAST-NU, Lahore, Pakistan

Problem 1: Bank Account Management

Description: You need to create a program that simulates bank account operations with additional functionality.

1. Create a base class `Transaction` with the following properties:

1. `transactionId` (String)
2. `date` (String)
3. `amount` (double)
4. `description` (String)

2. Create two subclasses `Deposit` and `Withdrawal` that inherit from `Transaction`.

1. `Deposit` should have a constructor that sets the type as "Deposit" and initializes the properties.
2. `Withdrawal` should have a constructor that sets the type as "Withdrawal" and initializes the properties.

3. Create a class `BankAccount` with the following properties:

1. `accountNumber` (String)
2. `accountHolder` (String)
3. `balance` (double)
4. `transactions` (ArrayList of `Transaction` objects)

4. The `BankAccount` class should have the following methods:

1. `deposit(double amount)`: Creates a `Deposit` transaction, adds it to the `transactions` list, and updates the balance.
2. `withdraw(double amount)`: Creates a `Withdrawal` transaction, adds it to the `transactions` list if there are sufficient funds, and updates the balance; otherwise, throw an `InsufficientFundsException`.
3. `displayBalance()`: Displays the current balance.
4. `displayTransactionHistory()`: Displays all the transactions made on the account, showing the type (Deposit or Withdrawal), date, and amount.

5. Create a `main` method where you:

1. Instantiate a `BankAccount` object.
2. Perform multiple deposits and withdrawals.
3. Display the final balance and the transaction history.

Expected Output Example:

Account Number: 12345678
Account Holder: John Doe
Depositing 500.0 on 2024-08-27...
Withdrawing 200.0 on 2024-08-28...
Withdrawing 2000.0 on 2024-08-29... (Insufficient funds)
Current Balance: 300.0
Transaction History:
- [Deposit] Date: 2024-08-27, Amount: 500.0
- [Withdrawal] Date: 2024-08-28, Amount: 200.0
- [Failed Withdrawal] Date: 2024-08-29, Amount: 2000.0

Problem 2: Student Grade Management

Description:
Create a program that manages and analyzes student grades with additional functionality.

1. Create a base class `Person` with the following properties:

a. `name` (String)
b. `id` (String)

2. Create a subclass `Student` that inherits from `Person` with the following additional properties:

a. `grades` (ArrayList of integers)

3. The `Student` class should have the following methods:

a. `addGrade(int grade)`: Adds a grade to the grades list.
b. `calculateAverage()`: Returns the average of the grades.
c. `displayGrades()`: Displays the student's name, ID, and grades.
d. `findHighestGrade()`: Returns the highest grade.
e. `findLowestGrade()`: Returns the lowest grade.
f. `hasPassed()`: Returns `true` if the average grade is above a certain threshold (e.g., 60), otherwise returns `false`.

4. Create a class `Course` with the following properties:

a. `courseName` (String)
b. `students` (ArrayList of `Student` objects)

5. The `Course` class should have the following methods:

a. `addStudent(Student student)`: Adds a student to the course.
b. `removeStudent(String studentId)`: Removes a student from the course based on their ID.
c. `displayAllStudents()`: Displays the details of all students in the course.
d. `findTopStudent()`: Finds and returns the student with the highest average grade.

6. In the `main` method:

a. Create a `Course` object.
b. Add 3 `Student` objects to the course by taking input from the user for names, IDs, and grades.
c. Display all students in the course with their grades, average grade, highest grade, lowest grade, and whether they passed the course.
d. Display the details of the top student in the course.

Expected Output Example:

Course: Introduction to Programming
Name: Alice, ID: S123
Grades: 85, 90, 78, 92, 88
Average Grade: 86.6
Highest Grade: 92
Lowest Grade: 78
Passed: Yes

Name: Bob, ID: S124
Grades: 70, 60, 75, 80, 68
Average Grade: 70.6
Highest Grade: 80
Lowest Grade: 60
Passed: Yes

Top Student: Alice (S123) with an average grade of 86.6