MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG443 – Worksheet for Lecture 3

Purpose:
In this work sheet, you will get hands-on experience with creating classes and objects in Java, and in particular experimenting with abstraction and encapsulation principles.

Tasks:

1. Write a method that checks whether two words are anagrams. Two words are anagrams if they contain the same letters in any order. For example, "silent" and "listen" are anagrams. The header method is as follows:

   ```
   public static boolean isAnagram(String s1, String s2)
   ```

   Write a program that prompts the user to enter two strings and, if they are anagrams, displays "anagram", otherwise displays "not anagram".
2. Design a class named Stock that contains the following:
   - A string data field named symbol for the stock's symbol.
   - A string data field named name for the stock's name.
   - A double data field named previousClosingPrice that stores the stock price for the previous day.
   - A double data field named currentPrice that stores the stock price for the current time.
   - A constructor that creates a stock with the specified symbol and name.
   - A method named getChangePercent() that returns the percentage changed from previousClosingPrice to currentPrice .

   Implement this class. Write a test program with the main method that creates a Stock object with the stock symbol ORCL, the name Oracle Corporation and the previous closing price of 34.5 . Set a new current price to 34.35 and display the price-change percentage.
3. Design a class named QuadraticEquation for a quadratic equation:

   $$ax^2 + bx + c = 0$$

   The class contains:
   ■ Private data fields a, b, and c that represent three coefficients.
   ■ A constructor for the arguments for a, b, and c.
   ■ Three getter methods for a, b, and c.
   ■ A method named getDiscriminant() that returns the discriminant which is $b^2 - 4ac$.
   ■ The methods named getRoot1() and getRoot2() for returning two roots of the equation.

   $$r_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad and \quad r_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

   These methods are useful only if the discriminant is non-negative. Let these methods return 0 if the discriminant is negative.

   Implement this class. Write a test program that prompts the user to enter values for a, b, and c and displays the result based on the discriminant. If the discriminant is positive, display the two roots. If the discriminant is 0, display the one root. Otherwise, display "The equation has no roots".
4. Design a class named Account that contains the following:
   - A private int data field named id for the account (default 0).
   - A private double data field named balance for the account (default 0).

- A private double data field named annualInterestRate that stores the current interest rate (default 0). Assume all accounts have the same interest rate.
- A private Date data field named dateCreated that stores the date when the account was created.
- A no-arg constructor that creates a default account.
- A constructor that creates an account with the specified id and initial balance.
- The accessor and mutator methods for id, balance, and annualInterestRate.
- The accessor method for dateCreated.
- A method named getMonthlyInterestRate() that returns the monthly interest rate.
- A method named getMonthlyInterest() that returns the monthly interest.
- A method named withdraw that withdraws a specified amount from the account.
- A method named deposit that deposits a specified amount to the account.

Write a test program that creates an Account object with an account ID of 1122, a balance of $20,000, and an annual interest rate of 4.5%. Use the withdraw method to withdraw $2,500, use the deposit method to deposit $3,000, and print the balance, the monthly interest, and the date when this account was created

5. Use the account class created in the previous exercise to simulate an ATM machine. Create 10 accounts in an array with an id 0,1,2…,9 and initial balance 50 dollars. The system prompts the user to enter an id. If the id is entered incorrectly, ask the user to enter a correct id. Once an id is accepted, display a menu as follows.

Sample run:

```
Enter an account id: 3

Main menu:
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 1
The balance is 50!

Main menu:
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 2
Enter an amount to withdraw: 25
```

Etc.