/**
 * BankAccount – The class BankAccount represents a simplification of a bank account like one you would see at any financial institution.
 */

public class BankAccount
{
    // Fields of this class

    // Interest rate
    private double m_dCurrentInterestRate;

    // The current account balance
    private double m_dBalance;

    // Constructors

    public BankAccount(double interestRate, double deposit)
    {
        m_dCurrentInterestRate = interestRate;
        m_dBalance = deposit;
    }

    public BankAccount(double interestRate)
    {
        m_dCurrentInterestRate = interestRate;
        m_dBalance = 0;
    }

    // Methods

    // Mutator Methods

    // Setting a new rate for the account:

    public void setRate(double newRate)
    {
        // Check for a good interest rate
        if (newRate > 0.0 && newRate < 20.0)
        {
            m_dCurrentInterestRate = newRate;
        }
    }
}
else
{
    System.out.println(newRate + " is not a valid interest rate");
}

// Depositing on the account

public void newDeposit(double amount)
{
    // Checking first for a good deposit
    if ( amount > 0)
    {
        m_dBalance += amount;
    }
    else
    {
        System.out.println(" Your deposit should be positive");
    }
}

// Withdrawing from an account

public void newWithdrawal(double withdraw)
{
    // Checking first for a good withdraw
    if ( withdraw > 0)
    {
        // Checking that this amount can be withdrawn
        if (m_dBalance >= withdraw)
        {
            m_dBalance -= withdraw;
        }
        else
        {
            System.out.println(" You don’t have enough money");
        }
    }
    else
    {
        System.out.println(" Your withdraw should be positive");
    }
}
public void monthlyChanges()
{
    monthlyFee();
    monthlyInterest();
}

// Private methods to implement the monthlyChanges
private void monthlyFee()
{
    _dBalance += 5.0; // $5.00 monthly maintenance fee
}

private void monthlyInterest()
{
    _dBalance += _dBalance*( _dCurrentInterestRate/1200.0);  
    // Try to figure out why 1200.0
}

// Accessor Methods
public double getRate()
{
    return _dCurrentInterestRate;
}

public double getBalance()
{
    int nCents = (int) ( _dBalance*100 + 0.5);
    return nCents/100.0;
}

}  // End the class definition