// Actual program running outside BlueJ.
/**
This program creates a bank account object and computes the balance at
the end of 10 years period, when you deposit the same quantity each
month.

It requires entering the interest rate in % as whole number, and the
amount deposited each month
*/

import java.util.*;
import java.io.*;

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
        {
            // Add code to handle invalid interest rate
        }
    }
}
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        // Checking that this amount can be withdrawn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()
{
m_dBalance -= 5.0; // $5.00 monthly maintenance fee
}

private void monthlyInterest()
{
    m_dBalance += m_dBalance*(m_dCurrentInterestRate/1200.0);

    // Try to figure out why 1200.0
}

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

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

public static void main(String args[])
{
    // first argument is the deposit per month
    double dInterest = (double) Integer.parseInt(args[0]);
    // second argument is the amount deposited by month
    double dPerMonth = (double) Integer.parseInt(args[1]);
    // Create my bank account
    BankAccount myBankAccount = new BankAccount(dInterest);
    // Make a deposit each month and show the growing of the balance each year
    System.out.println("Watch the balance of the account grow yearly");
    System.out.println();

    for (int whichYear = 0; whichYear < 10; whichYear++)
    {
        System.out.println("After year " + (whichYear+1));
        for (int nMonth = 0; nMonth < 12; nMonth++)
        {
            //make monthly deposit
    }
myBankAccount.newDeposit(dPerMonth);

// perform the monthly changes
myBankAccount.monthlyChanges();
}
System.out.println("$ " +myBankAccount.getBalance() );
}  // End the class definition