// Actual program running outside BlueJ.

/**
This program creates a bank account objects for each of three clients. It then
computes the balance at the end of 10 years period.

It requires entering the interest rate in % as whole number, the name of each
client and his/her monthly deposit for the three clients.
*/

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

public class UsingAccounts
{
    public static void main(String args[]) {

        // first argument is the interest rate
        double dInterest = (double) Integer.parseInt(args[0]);

        /*
        Because the Interest rate is a static field, we can set it up before we create
        any AccountAtBank object.
        */
        AccountAtBank.rate(dInterest);

        // Set the number of the next Account to start at 1
        AccountAtBank.putInitialAccountNumber(1);

        // Now we create an array for holding the names of each client
        String[] clientName;
        clientName = new String[3];

        // Now we create an array for holding the monthly deposit of each client
        // Note that we can combine both definition and allocation of space
        double[] clientDeposit = new double[3];

        for(int i = 0; i < 3; i++) {
            // even arguments in the main program are the client’s names
            clientName[i] = new String(args[2*i + 1]);

            // odd arguments are the clients’ monthly deposits.
            clientDeposit[i] = (double) Integer.parseInt(args[2*i + 2]);
        }

        // Create an array of references to clients
        Client[] clientArray;
clientArray = new Client[3];

// Create an array of references to bank accounts
AccountAtBank[] bankAccounts = new AccountAtBank[3];

/**
 * Note that no clients or bank accounts have been created yet
 * Only places to hold their addresses in main memory
 */

// Create 3 clients and put their names and monthly deposits
for(int i = 0; i < 3; i++)
{
    clientArray[i] = new Client(clientName[i], clientDeposit[i] );
}

/** Create the 3 account for the clients and put their names and monthly deposits.

Place the account number in the client
*/
for(int i = 0; i < 3; i++)
{
    bankAccounts[i] = new AccountAtBank(clientName[i], clientDeposit[i] );
    clientArray[i].putAccountNumber(bankAccounts[i].getAccountNumber());
}

// Make a deposit each month and show the growing of the balance each year
System.out.println("Watch the balance of the accounts grow yearly");
for(int nclient = 0; nclient < 3; nclient++)
{
    System.out.print (clientArray[nclient].getOwner() + " deposited $" );
    System.out.println(clientArray[nclient].getMDeposit());
    System.out.println();
    for ( int whichYear = 0; whichYear < 10; whichYear++)
    {
        // System.out.println(whichYear + " ");
        for ( int nMonth = 0; nMonth < 12; nMonth++)
        {
            //make monthly deposit
            bankAccounts[nclient].newDeposit (clientArray[nclient].getMDeposit());
            // perform the monthly changes
            bankAccounts[nclient].monthlyChanges();
        }
        System.out.print("At the end of year "+( whichYear+ 1));
        System.out.println(": $ " + bankAccounts[nclient].getBalance() );
    }
}
System.out.print("The account number of this client is ");
System.out.println(clientArray[nclient].getAccountNumber());
System.out.println(" ***************");
System.out.println();
} // End of main
} // End of UsingAccounts