

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра САПР

ОТЧЕТ
по лабораторной работе №6
по дисциплине «Объектно-ориентированное
программирование»
Тема: «Создание и использование классов»

Студенты гр. 1302

_____ Романова О.В.

_____ Новиков Г.В.

Преподаватель:

_____ Васильев А.А.

Санкт-Петербург

2023

1. Цель работы

Изучение классов и инкапсуляции в языке C# с помощью программного продукта компании Microsoft VS 2022.

2. Анализ задачи

Необходимо:

- 1) Написать программу, которая преобразовывает структуру, описывающую банковский счет в класс;
- 2) Написать программу, которая вносит изменения в прошлую программу, добавляет автоматическое назначение номера аккаунта;
- 3) Написать программу, которая вносит изменения в прошлую программу, добавляет два метода, которые добавляет и снимает деньги.

3. Ход выполнения работы

3.1 Упражнение 1

В ходе выполнения данного упражнения написана программа, которая преобразует структуру в класс, описывающий банковский счет с параметрами: номер, счет и тип аккаунта.

3.1.1 Пошаговое описание алгоритма

- 1) Изучить данные предоставленных файлов.
- 2) Изменить в BankAccount.cs структуру на класс.
- 3) Добавить метод Populate для назначения.
- 4) Добавить три метода, которые возвращают значения трех полей.
- 5) Изменить метод Write.
- 6) Ввод номера, счета.
- 7) Вывести на экран номер, счет и тип аккаунтов.

3.1.2 Используемые классы и методы

В программе, написанной в данном упражнении, используются следующие методы:

- `System.Console.WriteLine()` – служит для отображения на экране строк и значений переменных, переданных в метод в качестве параметров, с переходом на новую строку;

- `System.Console.ReadKey()` – ожидает следующего нажатия клавиши пользователем;

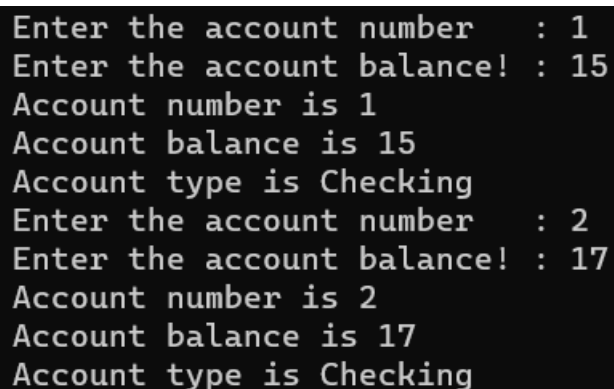
- `Main()` – служит для запуска программы.

- `BankAccount` – класс, который описывает банковский аккаунт с помощью методов: `Populate` (создание), `Number` (возвращает номер аккаунта), `Balance` (возвращает текущий баланс), `Type` (возвращает тип аккаунта).

- `CreateAccount` – класс, который содержит методы: `NewBankAccount` (создание аккаунта), `Write` (вывод информации об аккаунте).

3.1.3 Контрольный пример

На рис.3.1 представлены результаты выполнения программы.



```
Enter the account number : 1
Enter the account balance! : 15
Account number is 1
Account balance is 15
Account type is Checking
Enter the account number : 2
Enter the account balance! : 17
Account number is 2
Account balance is 17
Account type is Checking
```

Рис.3.1 Контрольный пример для программы

Как видно из рисунка, пользователем вводятся номера и баланс аккаунтов и на экран выводятся результаты.

3.2 Упражнение 2

В ходе выполнения данного упражнения, написана программа, которая вносит изменения в прошлую программу, добавляет автоматическое назначение номера аккаунта.

3.2.1 Пошаговое описание алгоритма

- 1) Ввод счета.
- 2) Использование метода `Populate`.
- 3) Вывод на экран номера, счета и типов аккаунтов.

3.2.2 Используемые классы и методы

В программе, написанной в данном упражнении, используются следующие методы:

- `System.Console.WriteLine()` – служит для отображения на экране строк и значений переменных, переданных в метод в качестве параметров, с переходом на новую строку;

- `System.Console.ReadKey()` – ожидает следующего нажатия клавиши пользователем;

- `Main()` – служит для запуска программы.

- `BankAccount` – класс, который описывает банковский аккаунт с помощью методов: `Populate` (создание), `Number` (возвращает номер аккаунта), `Balance` (возвращает текущий баланс), `Type` (возвращает тип аккаунта), `NextNumber` (считает номер аккаунта при создании).

- `CreateAccount` – класс, который содержит методы: `NewBankAccount` (создание аккаунта), `Write` (вывод информации об аккаунте).

3.2.3 Контрольный пример

На рис.3.2 представлены результаты выполнения программы.

```
Enter the account balance! : 5
Account number is 123
Account balance is 5
Account type is Checking
Enter the account balance! : 16
Account number is 124
Account balance is 16
Account type is Checking
```

Рис.3.2 Контрольный пример для программы

Как видно из рисунка, пользователем вводятся балансы аккаунтов и на экран выводятся результаты.

3.3 Упражнение 3

В ходе выполнения данного упражнения, написана программа, которая вносит изменения в прошлую программу, добавляет два метода, один вносит деньги на счет, другой снимает деньги со счета.

3.3.1 Пошаговое описание алгоритма

- 1) Ввод счета.
- 2) Использование метода Populate.
- 3) Использование методов TestDeposit и TestWithdraw, которые вносят деньги на счет и снимают их.
- 4) Вывод на экран номера, счета и типов аккаунтов.

3.3.2 Используемые классы и методы

В программе, написанной в данном упражнении, используются следующие методы:

- System.Console.WriteLine() – служит для отображения на экране строк и значений переменных, переданных в метод в качестве параметров, с переходом на новую строку;

- System.Console.ReadKey() – ожидает следующего нажатия клавиши пользователем;

- Main() – служит для запуска программы.

- BankAccount – класс, который описывает банковский аккаунт с помощью методов: Populate (создание), Number (возвращает номер аккаунта), Balance (возвращает текущий баланс), Type (возвращает тип аккаунта), NextNumber (считает номер аккаунта при создании), Withdraw (позволяет произвести снятие денег, только в 3 пункте) и Deposit (позволяет пополнить счет (только в 3 пункте)).

- CreateAccount – класс, который содержит методы: NewBankAccount (создание аккаунта), Write (вывод информации об аккаунте), TestWithdraw (попытка снятия денег), TestDeposit (попытка внесения денег).

3.3.3 Контрольный пример

На рис.3.3 представлены результаты выполнения программы.

```
Enter the account balance! : 14
Account number is 123
Account balance is 14
Account type is Checking
Enter the account balance! : 15
Account number is 124
Account balance is 15
Account type is Checking
Enter amount to deposit: 7
Account number is 123
Account balance is 21
Account type is Checking
Enter amount to deposit: 8
Account number is 124
Account balance is 23
Account type is Checking
Enter amount to withdraw: 2
Account number is 123
Account balance is 19
Account type is Checking
Enter amount to withdraw: 9
Account number is 124
Account balance is 14
Account type is Checking
```

Рис.3.3 Контрольный пример для программы

Как видно из рисунка, пользователем вводятся данные для работы программы и на экран выведены результаты.

4. Листинг программы

4.1 Листинг AccountType:

```
enum AccountType
{
    Checking,
    Deposit
}
```

4.2 Листинг BankAccount:

```
class BankAccount
{
    public void Populate(decimal balance)
    {
        accNo = NextNumber();
        accBal = balance;
        accType = AccountType.Checking;
    }

    private long accNo;
    private decimal accBal;
    private AccountType accType;

    public long Number()
    {
        return accNo;
    }
    public decimal Balance()
    {
        return accBal;
    }
    public string Type()
    {
        return accType.ToString();
    }

    //second
    private static long nextAccNo = 123;

    private static long NextNumber()
    {
        return nextAccNo++;
    }

    //third
    public decimal Deposit(decimal amount)
    {
        accBal = accBal + amount;
        return accBal;
    }

    public bool Withdraw(decimal amount)
    {
        bool sufficientFunds = accBal >= amount;
        if (sufficientFunds)
        {
            accBal -= amount;
        }
    }
}
```

```

    }
    return sufficientFunds;
}
}

```

4.2 Листинг CreateAccount:

```

using System;

class CreateAccount
{
    static void Main()
    {
        BankAccount berts = NewBankAccount();
        Write(berts);

        BankAccount freda = NewBankAccount();
        Write(freda);

        //third
        TestDeposit(berts);
        Write(berts);
        TestDeposit(freda);
        Write(freda);

        TestWithdraw(berts);
        Write(berts);
        TestWithdraw(freda);
        Write(freda);
    }

    static BankAccount NewBankAccount()
    {
        BankAccount created = new BankAccount();

        //for first:
        //Console.WriteLine("Enter the account number : ");
        //long number = long.Parse(Console.ReadLine());

        //for second:
        //long number = BankAccount.NextNumber();

        Console.WriteLine("Enter the account balance! : ");
        decimal balance = decimal.Parse(Console.ReadLine());

        //created.accNo = number;
        //created.accBal = balance;
        //created.accType = AccountType.Checking;

        created.Populate(balance);

        return created;
    }

    //third
    public static void TestDeposit(BankAccount acc)
    {
        Console.WriteLine("Enter amount to deposit: ");
        decimal amount = decimal.Parse(Console.ReadLine());
        acc.Deposit(amount);
    }

    public static void TestWithdraw(BankAccount acc)
    {

```



```

        Console.Write("Enter amount to withdraw: ");
        decimal amount = decimal.Parse(Console.ReadLine());
        if (!acc.Withdraw(amount))
        {
            Console.WriteLine("Insufficient funds");
        }
    }

    static void Write(BankAccount toWrite)
    {
        Console.WriteLine("Account number is {0}", toWrite.Number());
        Console.WriteLine("Account balance is {0}", toWrite.Balance());
        Console.WriteLine("Account type is {0}", toWrite.Type());
    }
}

```

5. Полученные результаты

В ходе выполнения данной лабораторной работы нами были получены следующие результаты:

- в ходе работы программы были созданы методы и классы, с помощью которых создается два банковских аккаунта, в третьем упражнении с счета можно снимать деньги и класть их, также на экран выводится вся информация об аккаунтах.

6. Выводы

В ходе выполнения данной лабораторной работы:

- были изучены классы в языке C#;

7. Список использованной литературы

1. MSDN – сеть разработчиков Microsoft. URL:

<https://learn.microsoft.com/ru-ru/dotnet/csharp/fundamentals/types/classes> (дата обращения: 01.04.2023)