

**Санкт-Петербургский национальный исследовательский
университет информационных технологий, механики и оптики**

ЛАБОРАТОРНАЯ РАБОТА №2

Создание и использование структуры

Студент:
Швалов Даниил Андреевич
Факультет ИКТ
Группа: K32211
Преподаватель:
Иванов Сергей Евгеньевич

Упражнение 1. Создание перечисления

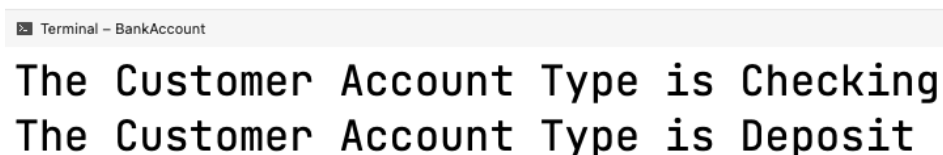
Проделав все шаги, перечисленные в задании, получим следующий исходный код:

```
namespace BankAccount;

public enum AccountType { Checking, Deposit }

class Program
{
    static void Main(string[] args)
    {
        AccountType goldAccount;
        AccountType platinumAccount;
        goldAccount = AccountType.Checking;
        platinumAccount = AccountType.Deposit;
        Console.WriteLine("The Customer Account Type is {0}", goldAccount);
        Console.WriteLine("The Customer Account Type is {0}", platinumAccount);
    }
}
```

Результат работы программы представлен на рис. 1.1.



```
Terminal - BankAccount
The Customer Account Type is Checking
The Customer Account Type is Deposit
```

Рис. 1.1: Вывод программы BankAccount

Упражнение 2. Создание и использование структуры

Выполнив все шаги из задания, получился следующий исходный код:

```
namespace StructType;

public enum AccountType { Checking, Deposit }

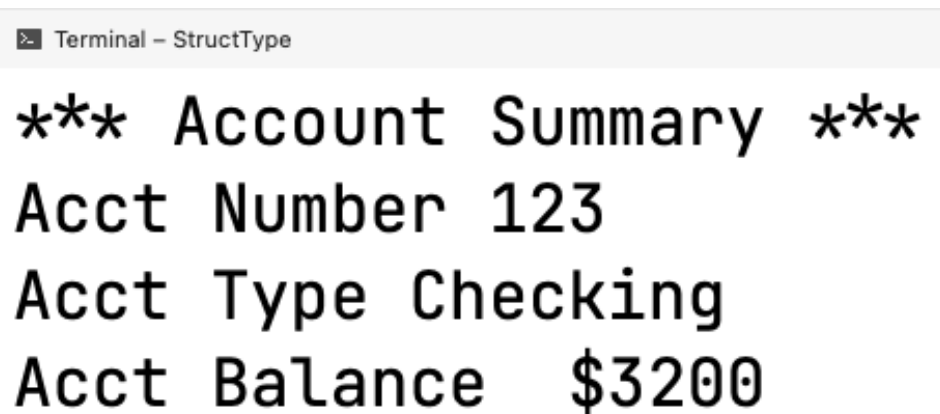
public struct BankAccount
{
    public long accNo;
    public decimal accBal;
    public AccountType accType;
}
```

```

class Program
{
    static void Main(string[] args)
    {
        BankAccount goldAccount;
        goldAccount.accType = AccountType.Checking;
        goldAccount.accBal = (decimal)3200.00;
        goldAccount.accNo = 123;
        Console.WriteLine("*** Account Summary ***");
        Console.WriteLine("Acct Number {0}", goldAccount.accNo);
        Console.WriteLine("Acct Type {0}", goldAccount.accType);
        Console.WriteLine("Acct Balance ${0}", goldAccount.accBal);
    }
}

```

Запустив его, получим вывод, изображенный на рис. 2.1.



```

Terminal - StructType

*** Account Summary ***
Acct Number 123
Acct Type Checking
Acct Balance $3200

```

Рис. 2.1: Вывод программы Struct

Доработав программу и добавив ввод пользовательских данных, получим следующую программу:

```

namespace StructType;

public enum AccountType { Checking, Deposit }

public struct BankAccount
{
    public long accNo;
    public decimal accBal;
    public AccountType accType;
}

class Program
{
    static void Main(string[] args)

```

```

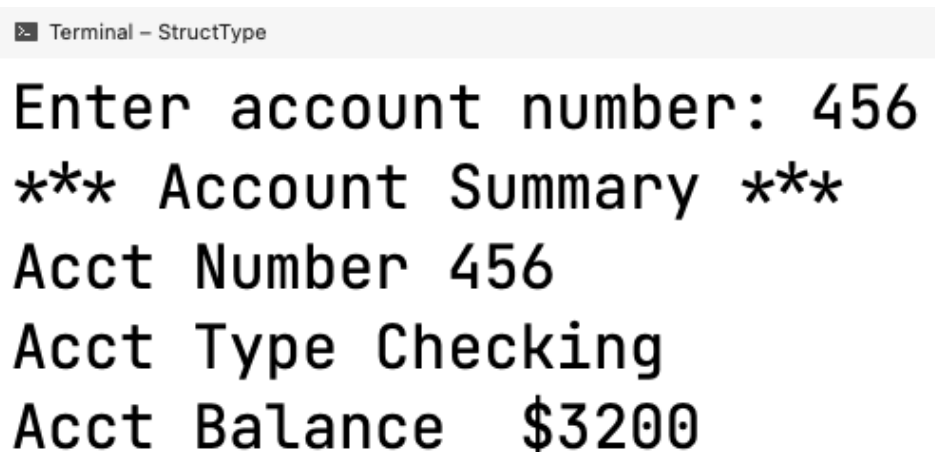
{
    BankAccount goldAccount;
    goldAccount.accType = AccountType.Checking;
    goldAccount.accBal = (decimal)3200.00;

    Console.Write("Enter account number: ");
    goldAccount.accNo = long.Parse(Console.ReadLine());

    Console.WriteLine("*** Account Summary ***");
    Console.WriteLine("Acct Number {0}", goldAccount.accNo);
    Console.WriteLine("Acct Type {0}", goldAccount.accType);
    Console.WriteLine("Acct Balance ${0}", goldAccount.accBal);
}
}

```

Запустив доработанную программу, получим результат, показанный на рис. 2.2.



```

Terminal - StructType
Enter account number: 456
*** Account Summary ***
Acct Number 456
Acct Type Checking
Acct Balance $3200

```

Рис. 2.2: Вывод программы доработанной Struct

Упражнение 3. Реализация структуры Distance

Согласно заданию считываем значения длин в дюймах с стандартного ввода. После этого записываем значение дюймов в каждое из двух расстояний, а в третье расстояние записываем их сумму также в дюймах. Затем для каждой структуры пересчитываем дюймы в футы. В конце концов, выводим значение расстояний на экран в заданном формате (см. рис. 3.1).

```

using System;

namespace Distance;

public struct Distance
{

```

```

    public int feet;
    public int inch;
}

class Program
{
    static void Main(string[] args)
    {
        Distance distance1, distance2, distance3;

        Console.WriteLine("Enter the first distance in inches:");
        distance1.inch = int.Parse(Console.ReadLine());
        Console.WriteLine("Enter the second distance in inches:");
        distance2.inch = int.Parse(Console.ReadLine());

        distance3.inch = distance1.inch + distance2.inch;

        const int inches_in_feet = 12;

        distance1.feet = (int)(distance1.inch / inches_in_feet);
        distance1.inch = distance1.inch % inches_in_feet;
        distance2.feet = (int)(distance2.inch / inches_in_feet);
        distance2.inch = distance2.inch % inches_in_feet;
        distance3.feet = (int)(distance3.inch / inches_in_feet);
        distance3.inch = distance3.inch % inches_in_feet;

        Console.WriteLine("Distance 1: {0}'-{1}\"", distance1.feet, distance1.inch);
        Console.WriteLine("Distance 2: {0}'-{1}\"", distance2.feet, distance2.inch);
        Console.WriteLine("Distance 3: {0}'-{1}\"", distance3.feet, distance3.inch);
    }
}

```

Terminal – Distance

```

Enter the first distance in inches:
15
Enter the second distance in inches:
13
Distance 1: 1'-3"
Distance 2: 1'-1"
Distance 3: 2'-4"

```

Рис. 3.1: Вывод программы Distance