

Санкт-Петербургский Национальный Исследовательский
Университет Информационных Технологий, Механики и Оптики

Факультет инфокоммуникационных технологий

Лабораторная работа №2
Создание и использование размерных типов данных

Выполнил
Стафеев И.А.

Группа
К3221

Проверил
Иванов С.Е.

Санкт-Петербург,
2024

СОДЕРЖАНИЕ

	Стр.
ВВЕДЕНИЕ	3
1 Упражнение 1.....	4
2 Упражнение 2.....	5
3 Упражнение 3.....	9
ЗАКЛЮЧЕНИЕ	12

ВВЕДЕНИЕ

Цель работы: изучение размерных типов, данных и приобретение навыков работы со структурными типами.

Для достижения цели необходимо выполнить следующие упражнения:

1. Создание перечисления
2. Создание и использование структуры
3. Реализация структуры Distance.

1 Упражнение 1

Задача: В этом упражнении вы создадите перечисление для представления различных типов банковских счетов. Затем вы используете данное перечисление для создания двух переменных, которым вы присвоите значения **Checking** и **Deposit**. Далее вы выведете на экран значения этих переменных, используя функцию **System.Console.WriteLine**.

В соответствии с задачей был написан код программы (рисунок 1).

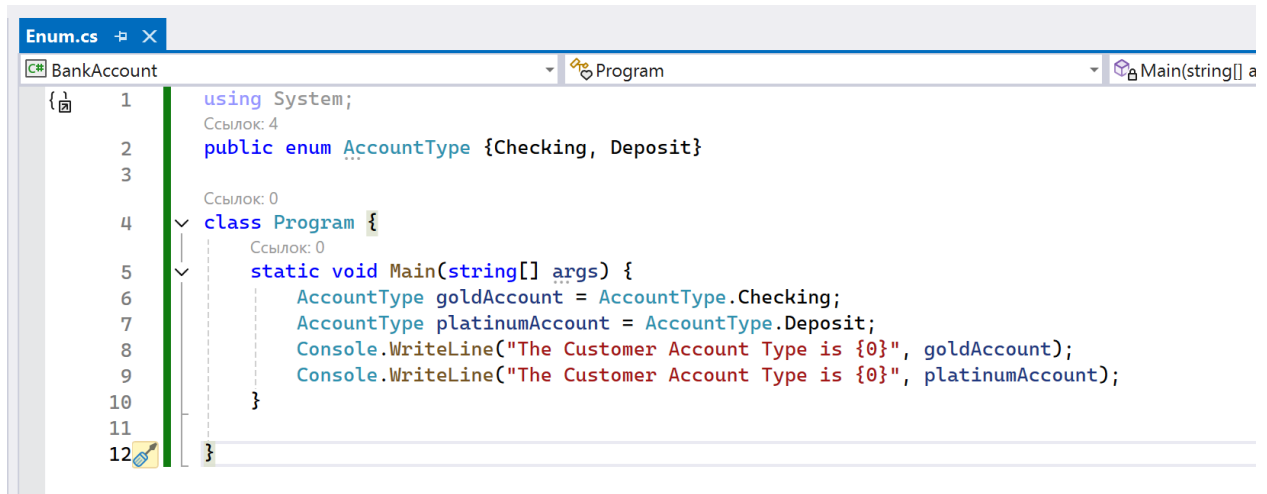


Рисунок 1 — Код программы

Результат выполнения этой программы показан на рисунке 2. Как можно заметить, переменным **goldAccount** и **platinumAccount** были присвоены значения **Checking** и **Deposit**. Это константы, заданные в перечислении enum.

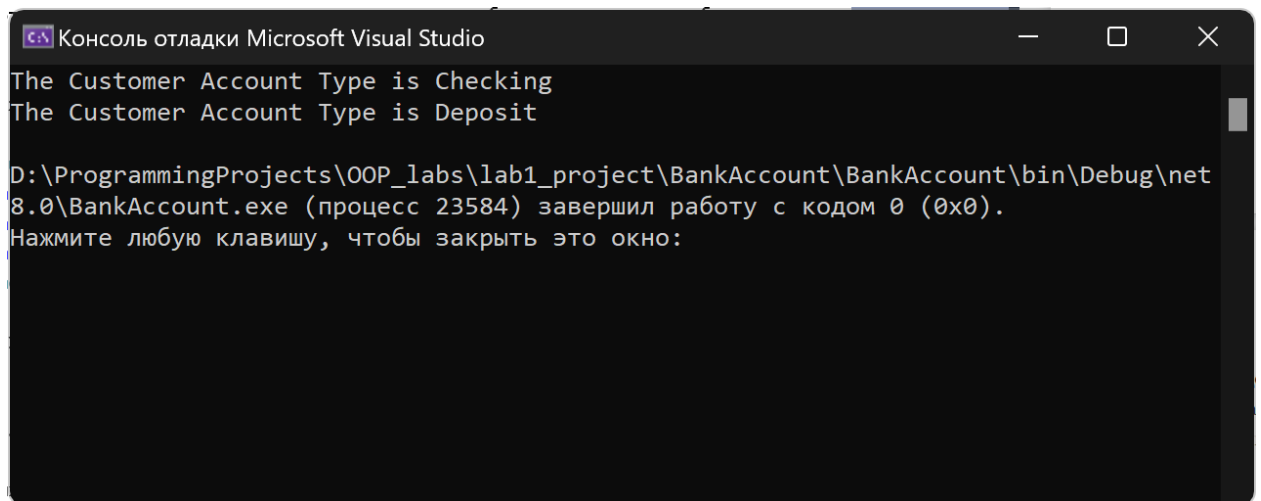
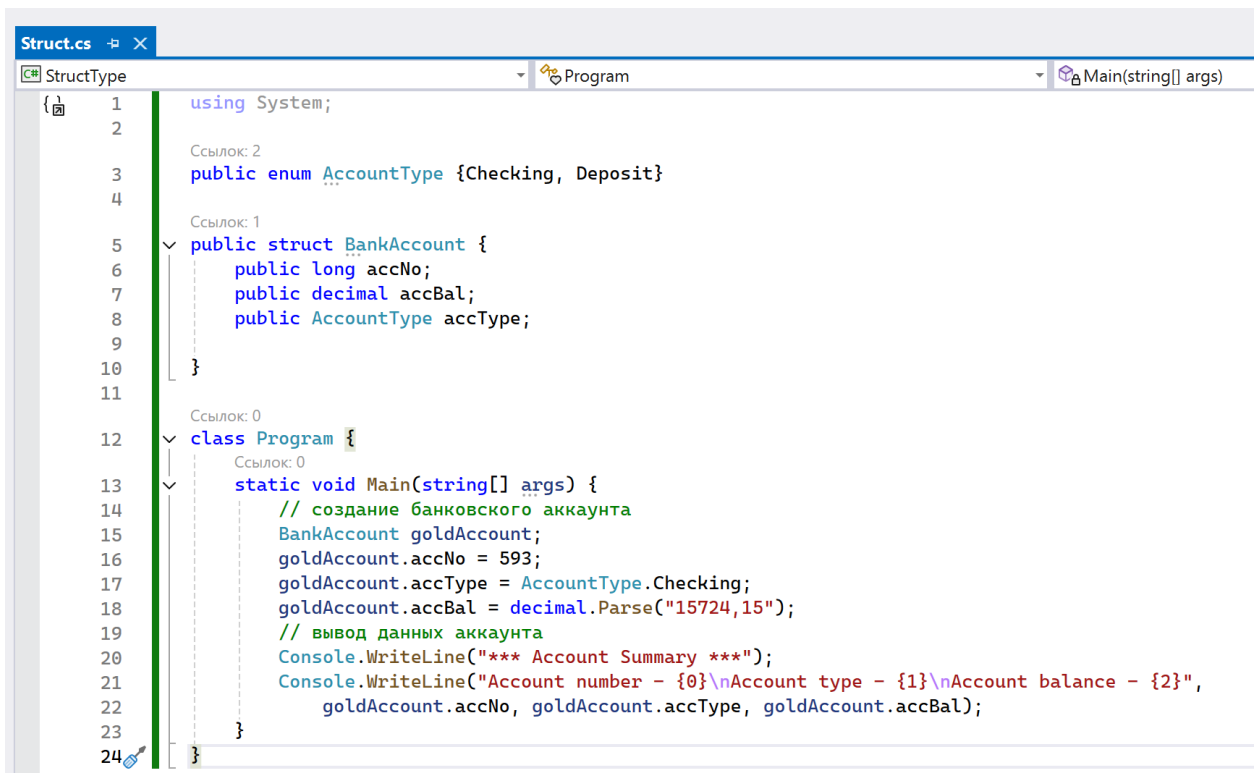


Рисунок 2 — Результат выполнения программы

2 Упражнение 2

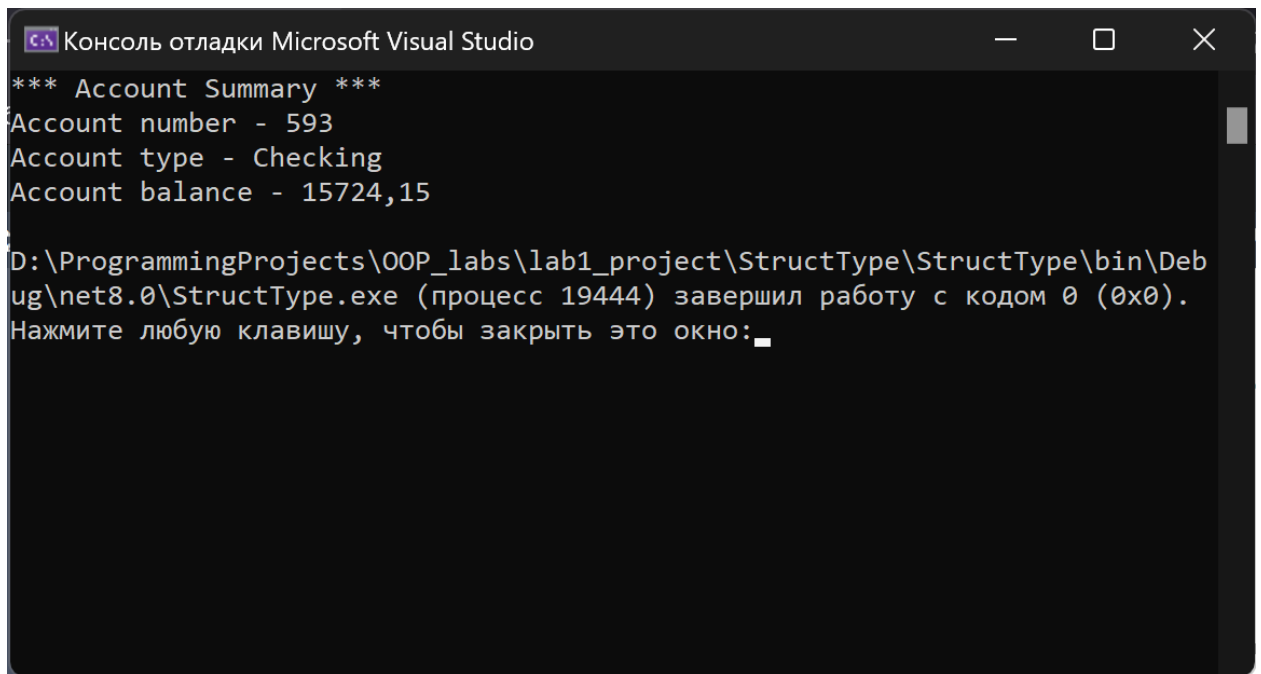
Задача: В этом упражнении вы создадите структуру, которую можно использовать для представления банковских счетов. Для хранения номеров счетов (тип данных **long**), балансов счетов (тип данных **decimal**) и типов счетов (перечисление, созданное в упражнении 1) вы будете использовать переменные. Затем создадите переменную типа структуры, заполните ее данными и выведете результаты на консоль.

Код программы показан на рисунке 3. В соответствии с заданием было создано перечисления для типов банковских аккаунтов и структура для банковского аккаунта. Затем в Main создана переменная типа структура и внесены данные в ее поля, после чего поля структуры выведены на экран 4.



```
1  using System;
2
3  Ссылка: 2
4  public enum AccountType {Checking, Deposit}
5
6  Ссылка: 1
7  public struct BankAccount {
8      public long accNo;
9      public decimal accBal;
10     public AccountType accType;
11 }
12
13 Ссылка: 0
14 class Program {
15     Ссылка: 0
16     static void Main(string[] args) {
17         // создание банковского аккаунта
18         BankAccount goldAccount;
19         goldAccount.accNo = 593;
20         goldAccount.accType = AccountType.Checking;
21         goldAccount.accBal = decimal.Parse("15724,15");
22         // вывод данных аккаунта
23         Console.WriteLine("*** Account Summary ***");
24         Console.WriteLine("Account number - {0}\nAccount type - {1}\nAccount balance - {2}",
25                             goldAccount.accNo, goldAccount.accType, goldAccount.accBal);
26     }
27 }
```

Рисунок 3 — Код программы



```
Консоль отладки Microsoft Visual Studio

*** Account Summary ***
Account number - 593
Account type - Checking
Account balance - 15724,15

D:\ProgrammingProjects\OOP_labs\lab1_project\StructType\StructType\bin\Debug\net8.0\StructType.exe (процесс 19444) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно: _
```

Рисунок 4 — Результат выполнения программы

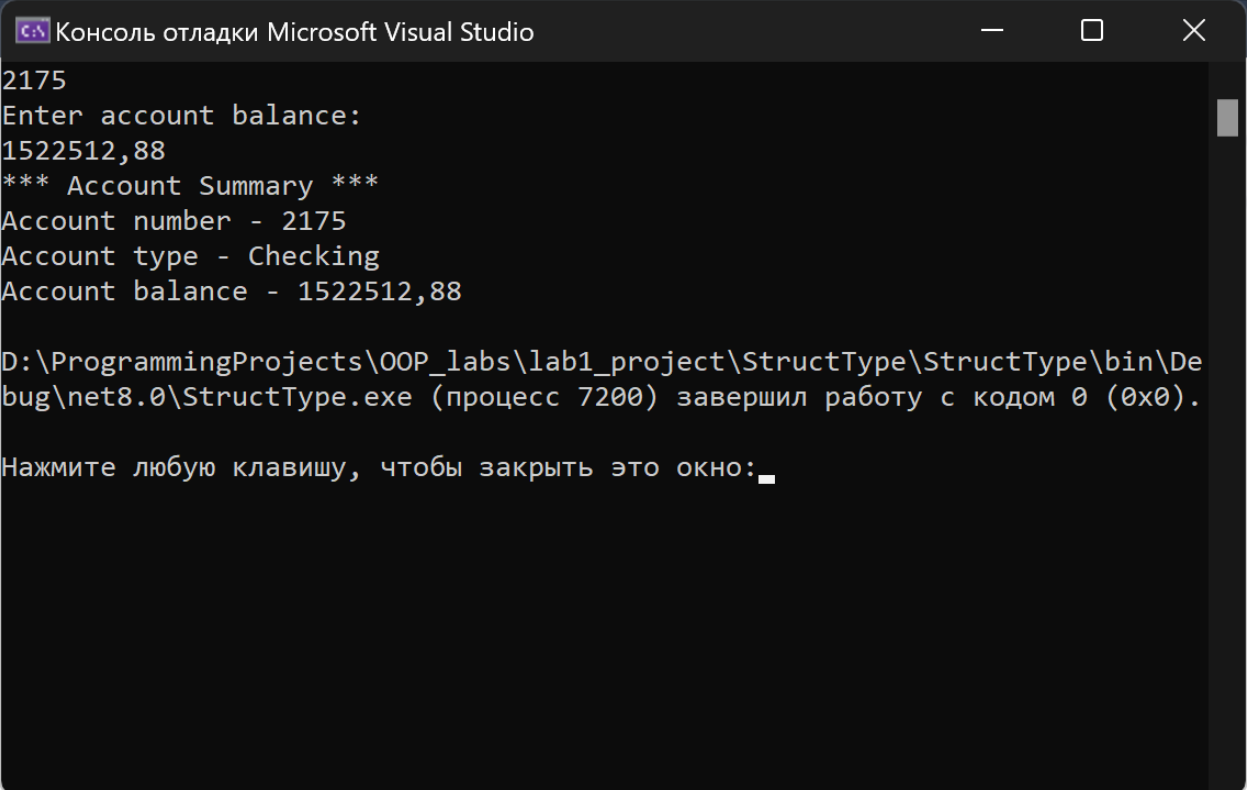
При добавлении пользовательского ввода необходимо учитывать некорректный формат входных данных, поэтому была добавлена обработка исключений (рисунок 5).



```
class Program {
    static void Main(string[] args) {
        // создание банковского аккаунта
        try
        {
            BankAccount goldAccount;
            Console.WriteLine("Enter account number:");
            goldAccount.accNo = long.Parse(Console.ReadLine());
            goldAccount.accType = AccountType.Checking;
            Console.WriteLine("Enter account balance:");
            goldAccount.accBal = decimal.Parse(Console.ReadLine());
            // вывод данных аккаунта
            Console.WriteLine("*** Account Summary ***");
            Console.WriteLine("Account number - {0}\nAccount type - {1}\nAccount balance - {2}",
                goldAccount.accNo, goldAccount.accType, goldAccount.accBal);
        }
        catch (FormatException e) {
            Console.WriteLine("A format exception has occurred: {0}", e.Message);
        }
        catch (Exception e)
        {
            Console.WriteLine("An unexpected error has occurred: {0}", e.Message);
        }
    }
}
```

Рисунок 5 — Измененный код класса Program

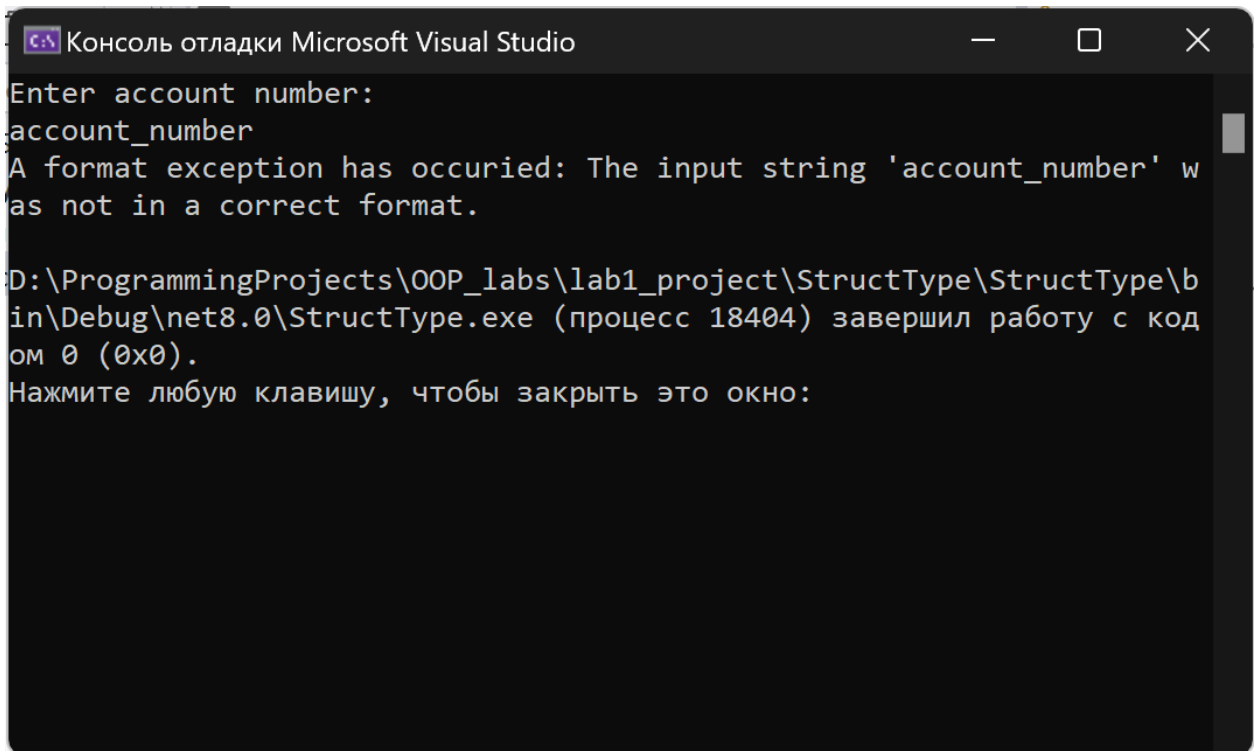
Примеры выполнения программы с пользовательским вводом при корректных и некорректных входных данных показаны на рисунках 6 и 7.

The image shows a screenshot of the 'Консоль отладки Microsoft Visual Studio' (Microsoft Visual Studio Debug Console) window. The window has a dark background and a title bar with standard Windows window controls. The text inside the console is as follows:
2175
Enter account balance:
1522512,88
*** Account Summary ***
Account number - 2175
Account type - Checking
Account balance - 1522512,88

D:\ProgrammingProjects\OOP_labs\lab1_project\StructType\StructType\bin\Debug\net8.0\StructType.exe (процесс 7200) завершил работу с кодом 0 (0x0).

Нажмите любую клавишу, чтобы закрыть это окно:

Рисунок 6 — Пример выполнения с корректными входными данными



Консоль отладки Microsoft Visual Studio

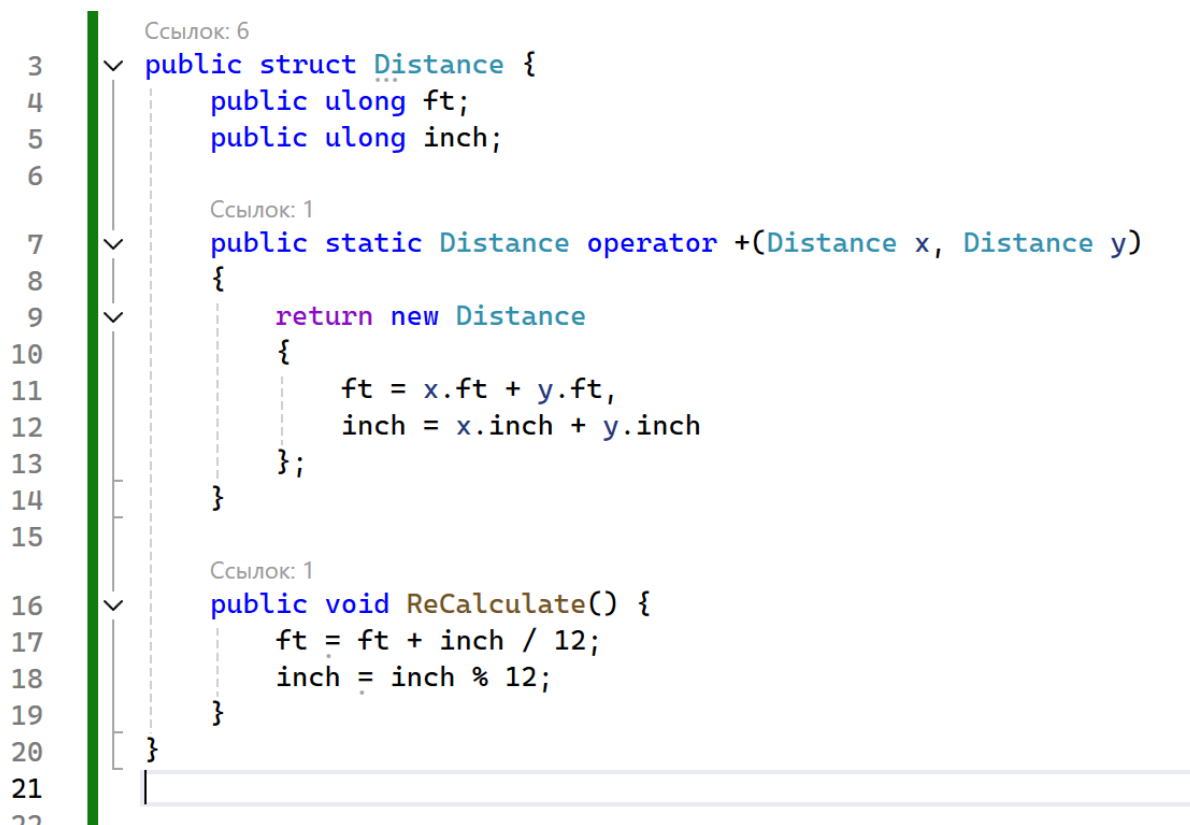
```
Enter account number:  
account_number  
A format exception has occurred: The input string 'account_number' w  
as not in a correct format.  
  
D:\ProgrammingProjects\OOP_labs\lab1_project\StructType\StructType\b  
in\Debug\net8.0\StructType.exe (процесс 18404) завершил работу с код  
ом 0 (0x0).  
Нажмите любую клавишу, чтобы закрыть это окно:
```

Рисунок 7 — Пример выполнения с некорректными входными данными

3 Упражнение 3

Задача: Требуется создать структуру **Distance**, определяющую длину в английской системе мер. В английской системе мер основными единицами измерения длины служат фут и дюйм, причем один фут равен 12 дюймам. Расстояние, равное, например, 15 футам и 8 дюймам, на экран выведете как 15 ' - 8". Дефис в данной записи будет служить для разделения значений футов и дюймов.

Код структуры Distance приведен на рисунке 8. Для этой структуры был создан оператор сложения, который возвращает новый объект структуры, где значениями полей являются суммы соответствующих значений полей "слагаемых и метод для приведения футов и дюймов к стандартному формату (количество дюймов меньше 12-ти).



```
3  public struct Distance {
4      public ulong ft;
5      public ulong inch;
6
7      Ссылка: 1
8      public static Distance operator +(Distance x, Distance y)
9      {
10         return new Distance
11         {
12             ft = x.ft + y.ft,
13             inch = x.inch + y.inch
14         };
15
16         Ссылка: 1
17         public void ReCalculate() {
18             ft = ft + inch / 12;
19             inch = inch % 12;
20         }
21     }
22 }
```

Рисунок 8 — Структура Distance

Код основного класса программы приведен на рисунке 9. Поскольку данные вводит пользователь, добавлена обработка исключений.

```

22
23 class Program {
24     static void Main(string[] args) {
25         try {
26             Distance dist1, dist2;
27             Console.WriteLine("Enter distance 1 fts:");
28             dist1.ft = ulong.Parse(Console.ReadLine());
29             Console.WriteLine("Enter distance 1 inches:");
30             dist1.inch = ulong.Parse(Console.ReadLine());
31             Console.WriteLine("Enter distance 2 fts:");
32             dist2.ft = ulong.Parse(Console.ReadLine());
33             Console.WriteLine("Enter distance 2 inches:");
34             dist2.inch = ulong.Parse(Console.ReadLine());
35             Distance dist3 = dist1 + dist2;
36             dist3.ReCalculate();
37             Console.WriteLine("Total distance is {0}'-{1}\"", dist3.ft, dist3.inch);
38         }
39         catch (FormatException e)
40         {
41             Console.WriteLine("Input data must be positive integer.");
42         }
43         catch (Exception e)
44         {
45             Console.WriteLine("An unexpected error has occurred: {0}", e.Message);
46         }
47     }
48 }
49

```

Рисунок 9 — Класс Program

Пример выполнения программы с корректными и некорректными данными показан на рисунках 10, 11 и 12.

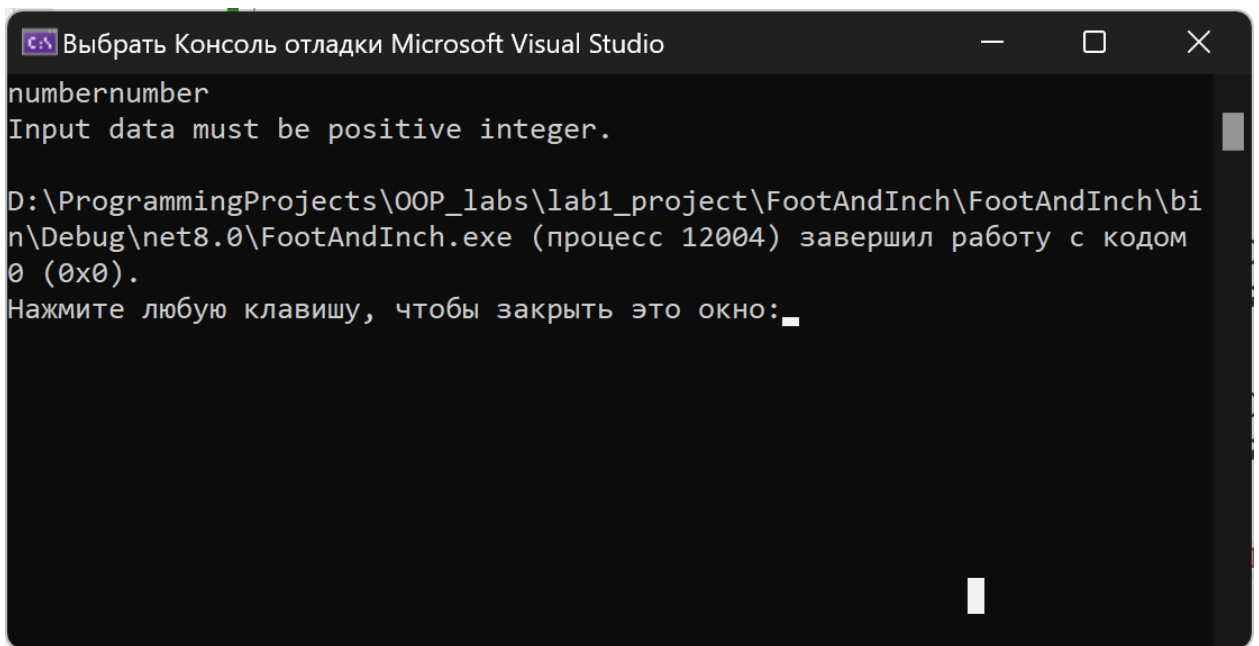
```

Консоль отладки Microsoft Visual Studio
Enter distance 1 fts:
11
Enter distance 1 inches:
16
Enter distance 2 fts:
24
Enter distance 2 inches:
2
Total distance is 36'-6"

D:\ProgrammingProjects\OOP_labs\lab1_project\FootAndInch\FootAndInch\bin\Debug\net8
.0\FootAndInch.exe (процесс 8744) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно:

```

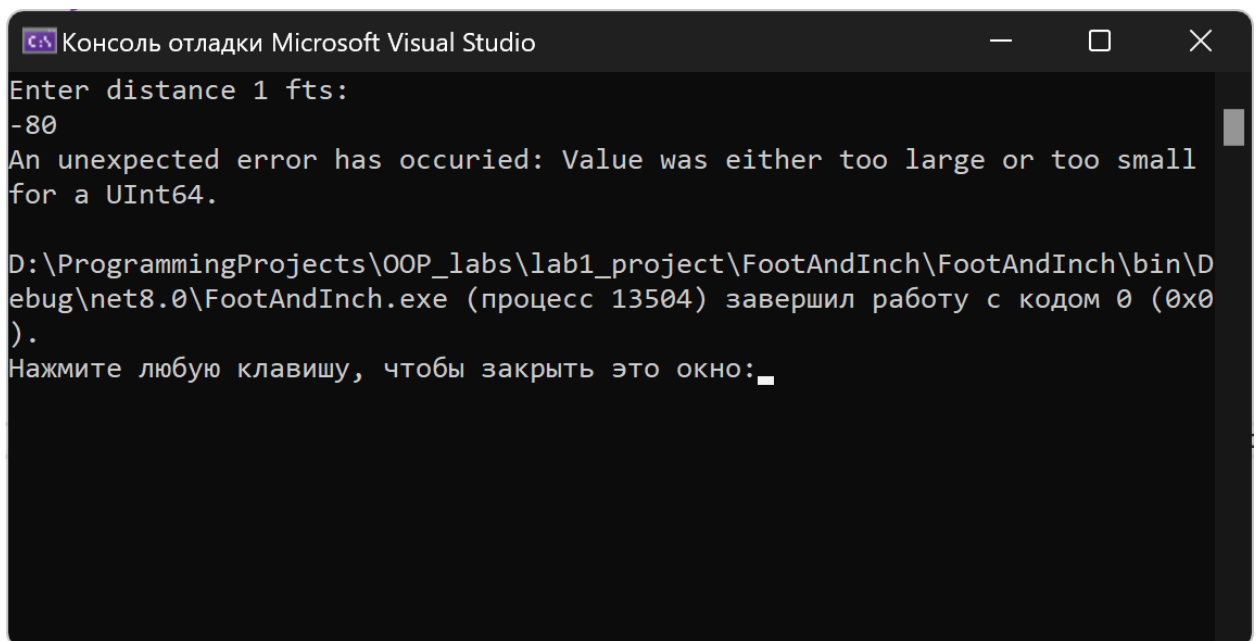
Рисунок 10 — Пример с корректными данными



```
Выбрать Консоль отладки Microsoft Visual Studio
numbernumber
Input data must be positive integer.

D:\ProgrammingProjects\OOP_labs\lab1_project\FootAndInch\FootAndInch\bin\
n\Debug\net8.0\FootAndInch.exe (процесс 12004) завершил работу с кодом
0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно: _
```

Рисунок 11 — Пример 1 с некорректными данными



```
Консоль отладки Microsoft Visual Studio
Enter distance 1 fts:
-80
An unexpected error has occurred: Value was either too large or too small
for a UInt64.

D:\ProgrammingProjects\OOP_labs\lab1_project\FootAndInch\FootAndInch\bin\D
ebug\net8.0\FootAndInch.exe (процесс 13504) завершил работу с кодом 0 (0x0
).
Нажмите любую клавишу, чтобы закрыть это окно: _
```

Рисунок 12 — Пример 2 с некорректными данными

ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы были выполнены все требуемые упражнения. Цель работы достигнута. Получены знания о размерных типах и приобретены навыки работы со структурными типами.