Отчёт  
по самостоятельной работе №1

по дисциплине: «ОП.04 Основы алгоритмизации и программирования»

Тема: «Создание библиотек подпрограмм»

|  |  |  |
| --- | --- | --- |
| Студент гр. 13 ИТ |  | Липинский К.С. |
| Преподаватель |  | Счастливцев А.К. |

### Цель работы.

Ознакомиться с созданием библиотек подпрограмм, получить начальные сведения о связи модулей.

### Теоретические сведения.

Определение специальных пространств имен явным образом играет важную роль при совместной разработке, поскольку другие разработчики должны иметь возможность импортировать эти специальные пространства имен, чтобы использовать содержащиеся в них типы.

***Пространство имен*** определяет область объявлений, в которой допускается хранить одно множество имен отдельно от другого.

По существу, имена, объявленные в одном пространстве имен, не будут вступать в конфликт с аналогичными именами, объявленными в другой области.

Так, в библиотеке классов для среды .NET Framework, которая одновременно является библиотекой классов C#, используется пространство имен System.

Такой механизм предусматривает, что исходный текст каждой процедуры или функции хранится в отдельном файле и при необходимости с помощью специальной директивы компилятора включается в текст создаваемой программы.

### Выполнение работы.

Рассмотрим приведённый код. Каждая строка была закомментирована её описанием:

using System; // Подключение стандартного пространства имён System для использования основных функций C#

namespace MathFunctions // Пространство имён MathFunctions для определения пользовательских функций

{

class CustomFunc // Класс, содержащий пользовательские функции

{

public int res {get; set;} // Свойство для хранения результата вычислений

public int Num; // Переменная для хранения числа, с которым будут выполняться операции

public CustomFunc(int n) { // Конструктор класса, принимающий целое число в качестве аргумента

Num = n; // Инициализация переменной Num переданным значением

res = 0; // Инициализация res значением 0

}

public void FirstAndLast() // Метод для вычисления суммы первого и последнего разрядов числа

{

string temp = Convert.ToString(Num); // Преобразование числа Num в строку

if (temp == null) // Проверка на случай, если преобразование вернуло null

res = 0; // Установка res в 0, если строка пустая

int strLength = temp.Length; // Определение длины строки

if (strLength > 1) { // Если длина строки больше 1

int f = Convert.ToInt32(temp.Substring(0, 1)); // Извлечение первого разряда и преобразование его в целое число

int l = Convert.ToInt32(temp.Substring(strLength - 1)); // Извлечение последнего разряда и преобразование его в целое число

res = f + l; // Вычисление суммы первого и последнего разрядов

}

else res = Convert.ToInt32(temp); // Если длина строки равна 1, то res равно самому числу

}

public void Swap() // Метод для замены местами первого и последнего разрядов числа

{

string temp = Convert.ToString(Num); // Преобразование числа Num в строку

if (temp == null) // Проверка на случай, если преобразование вернуло null

res = 0; // Установка res в 0, если строка пустая

int strLength = temp.Length; // Определение длины строки

if (strLength > 1) { // Если длина строки больше 1

var subStr = temp.Substring(1, temp.Length-2); // Извлечение подстроки со всеми символами, кроме первого и последнего

res = Convert.ToInt32(temp.Substring(strLength - 1) + subStr + temp.Substring(0, 1)); // Сборка новой строки с переставленными разрядами и преобразование её в целое число

}

else res = Convert.ToInt32(temp); // Если длина строки равна 1, то res равно самому числу

}

}

}

// Подключение библиотеки в файле, содержащем Main():

using System; // Подключение стандартного пространства имён System для использования основных функциональностей C#

using MathFunctions; // Подключение пользовательского пространства имён MathFunctions

namespace CustomLibrary // Пространство имён CustomLibrary для определения основного класса программы

{

class Program // Основной класс программы

{

public static void Main(string[] args) // Главный метод, точка входа в программу

{

CustomFunc m\_obj = new CustomFunc(256); // Создание экземпляра класса CustomFunc с числом 256

m\_obj.FirstAndLast(); // Вызов метода для вычисления суммы первого и последнего разрядов числа

Console.WriteLine("Для числа 256 сумма первого и последнего разрядов равна: {0}", m\_obj.res); // Вывод результата работы метода FirstAndLast в консоль

m\_obj.Swap(); // Вызов метода для замены местами первого и последнего разрядов числа

Console.WriteLine("Если у числа 256 поменять местами первый и последний разряды, то получим число: {0}", m\_obj.res); // Вывод результата работы метода Swap в консоль

}

}

}

### Контрольные вопросы.

1. Для создания библиотеки используется проект типа "Class Library". Этот тип проекта позволяет создавать библиотеки, которые могут быть использованы в других приложениях.
2. Директива, подключающая внешние библиотеки – using [пространство имён].
3. Для создания библиотеки в C# нужно создать файл класса. Способ его создания может варьироваться в зависимости от среды разработки, но обычно нужно сделать следующее (на примере Visual Studio 2022):
4. Нажать правой кнопкой мыши по проекту -> Добавить -> Класс (рис. 1).

A screenshot of a computer

Description automatically generated

Рисунок 1

1. Затем, нужно дать ему имя и нажать кнопку «Добавить» (рис. 2).

A screenshot of a computer

Description automatically generated

Рисунок 2

1. Класс создался. Теперь нужно его наполнить необходимыми функциями и импортировать в нужное место (рис. 3).

A screenshot of a computer

Description automatically generated

Рисунок 3

1. Добавим в библиотеку функцию для нахождения суммы всех разрядов введённого числа:

В конструктор добавим свойство для хранения суммы:

class CustomFunc

{

public int Res { get; private set; }

public int Num { get; }

public int Sum { get; private set; }

public CustomFunc(int num)

{

Num = num;

Res = 0;

Sum = 0;

}

public void SumDig() // Сумма всех цифр числа

{

int absNum = Math.Abs(Num); // На всякий случай извлекаем модуль из числа, чтобы не было ошибок в подсчёте, если оно отрицательное

int sum = 0;

while (absNum > 0)

{

sum += absNum % 10;

absNum /= 10;

}

Sum = sum;

}

В главном файле добавим вызов функции:

class Program

{

public static void Main(string[] args)

{

CustomFunc mObj = new CustomFunc(256); // Используем класс из библиотеки MathFunctions

mObj.FirstAndLast();

Console.WriteLine("Для числа 256 сумма первого и последнего разрядов равна: {0}", mObj.Res);

mObj.Swap();

Console.WriteLine(

"Если у числа 256 поменять местами первый и последний разряды, то получим число: " +

"{0}", mObj.Res);

mObj.SumDig();

Console.WriteLine("Сумма разрядов числа: {0}", mObj.Sum);

}

Тестируем (рис. 4).

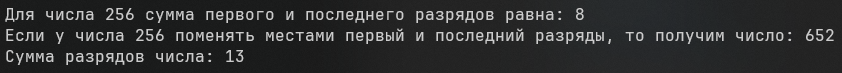


Рисунок 4

### Вывод.

Целью практической работы было ознакомление с созданием библиотек подпрограмм и получение начальных сведений о связи модулей. В ходе выполнения задачи был создан класс библиотеки, реализующий несколько методов по работе с числами. Произведено подключение библиотеки к основному проекту, продемонстрирована работа методов через вызовы из основного кода. Работа позволила усвоить основные принципы создания и взаимодействия модулей в C#.