**Лабораторна робота №9**

**Тема:** Основи роботи в середовищі програмування Visual C# 2010 Express Edition для створення консольних програм.

**Мета:** Засвоїти методику та виробити практичні навички роботи та створення консольних програм за допомогою Visual C# 2010.

**Теоретичні відомості.**

C# (вимовляється Сі-шарп) — об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Синтаксис C# близький до С++ і Java.

C# є дуже близьким родичем мови програмування Java. Мова Java була створена компанією Sun Microsystems, коли глобальний розвиток інтернету поставив задачу роззосереджених обчислень. Взявши за основу популярну мову C++, Java виключила з неї потенційно небезпечені речі (типу вказівників без контролю виходу за межі). Для роззосереджених обчислень була створена концепція віртуальної машини та машинно-незалежного байт-коду, свого роду посередника між вихідним текстом програм і апаратними інструкціями комп'ютера чи іншого інтелектульного пристрою.

Java набула чималої популярності, і була ліцензована також і компанією Microsoft. Але з плином часу Sun почала винуватити Microsoft, що та при створенні свого клону Java робить її сумісною виключно з платформою Windows, чим супепречить самій концепції машинно-незалежного середовища виконання і порушує ліцензійну угоду. Microsoft відмовилася піти назустріч вимогам Sun, і тому з'ясування стосунків набуло статусу судового процесу. Суд визнав позицію Sun справедливою, і зобов'язав Microsoft відмовитися від позаліцензійного використання Java.

У цій ситуації в Microsoft вирішили, користуючись своєю вагою на ринку, створити свій власний аналог Java, мови в якій корпорація стане повновласним господарем. Ця створена мова отримала назву C#. Вона успадкувала від Java концепції віртуальної машини (середовище .NET), байт-коду (MSIL) і більшої безпеки вихідного коду програм, плюс врахувала досвід використання програм на Java.

Нововведенням C# стала можливість легшої взаємодії, порівняно з мовами-попередниками, з кодом програм, написаних на інших мовах, що є важливим при створенні великих проектів. Якщо програми на різних мовах виконуються на платформі .NET, .NET бере на себе клопіт по сумісності програм (тобто типів даних, за кінцевим рахунком).

Станом на сьогодні C# визначено флагманською мовою корпорації Microsoft, бо вона найповніше використовує нові можливості .NET. Решта мов програмування, хоч і підтримуються, але визнані такими, що мають спадкові прогалини щодо використання .NET.

У C# підтримуються всі типи даних, а також цілі числа без знака і новий 128-розрядний тип чисел з плаваючою комою високої точності.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Коротке імя** | **Класс .NET** | **Тип** | **Розмір** | **Діапазон (біт)** |
| *byte* | Byte | Ціле число без знаку | 8 | Від 0 до 255 |
| *sbyte* | SByte | Ціле число зі знаком | 8 | Від -127 до 127 |
| *int* | Int32 | Ціле число зі знаком | 32 | Від -2 147 483 648 до 2 147 483 647 |
| *uint* | UInt32 | Ціле число без знаку | 32 | Від 0 до 4 294 967 259 |
| *short* | Int16 | Ціле число зі знаком | 16 | Від -32 768 до 32 767 |
| *ushort* | UInt16 | Ціле число без знаку | 16 | Від 0 до 65 535 |
| *long* | Int64 | Ціле число зі знаком | 64 | Від -922 337 203 685 477 508  до 922 337 203 685 477 507 |
| *ulong* | UInt64 | Ціле число без знаку | 64 | Від 0 до 18 446 744 073 709 551 615 |
| *float* | Single | Число з плаваючою комою | 32 | Від -3,402 823е38 до 3,402 823е38 |
| *double* | Double | Число з плаваючою комою | 64 | від -1,797 693 134 862 32e308 до 1,797 693 134 862 32e308 |
| *char* | Char | Одиночний знак Юнікода | 16 | Знаки Юнікода в тексті |
| *bool* | Boolean | Логічний тип | 8 | true або false |
| *object* | Object | Базовий тип для всіх інших типів |  |  |
| *string* | String | Послідовність знаків |  |  |

В мові програмування C# вивід даних відбувається за допомогою фіункцій Write і WriteLine:

*int MyInt = 100;*

*String MyString = MyInt.ToString("C");*

*Console.WriteLine(MyString);*

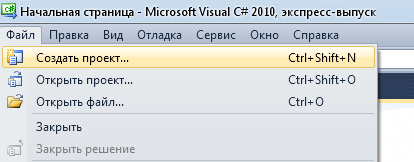
Ввід даних відбувається за допомогою функції ReadLine:

*Console.Write("Введіть імя і натисніть Enter: ");*

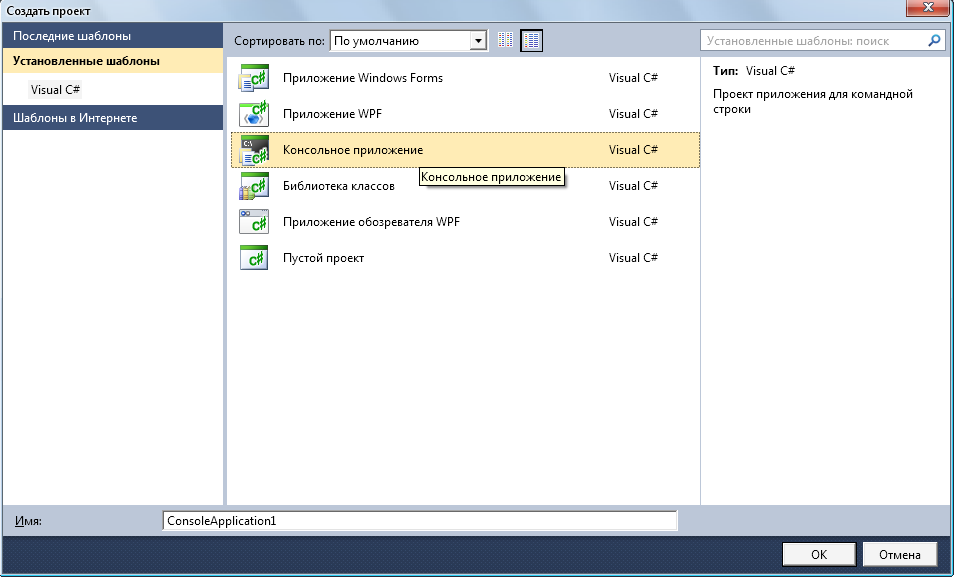
*string name = Console.ReadLine();*

*Console.WriteLine("Привіт, " + name + "!");*

Пишемо першу программу на C#. Запускайте Visual Studio.NET. Для створення нового пустого проекту C # натискаємо на кнопку на New Project (Новой проект), або натискаємо комбінацію клавіш Ctrl + Shift + N, або просто заходимо в меню File (Файл) і далі вибираємо New Project (Создать проект):



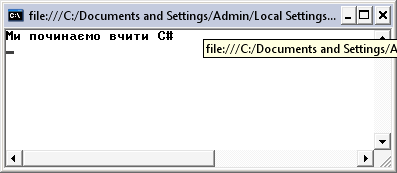
У вікні New Project (Создать проект) зліва вибираємо Visual C#, а справа тип програми - Console Application (Консольное приложение):



Як ім'я проекту друкуэмо FirstApp або щось в цьому роді. Нажати на кпопку Ok для закриття данного діалогового вікна. Тепер приступаємо до коду. Наша перша програма просто виведе деяке фіксоване слово в консольне віконце:

|  |
| --- |
| *using System;*  *using System.Collections.Generic;*  *using System.Linq;*  *using System.Text;*  *namespace FirstApp*  *{*  *class Program*  *{*  *static void Main(string[] args)*  *{*  *Console.WriteLine("Ми починаємо вчити C#");*  *Console.ReadKey();*  *}*  *}*  *}* |

Запускаємо програму, натиснувши Ctrl + F5. Результат буде таким:



Часто буває корисно вставляти в програму текст, який є коментарем тільки для читання людиною, що ігнорується компілятором. У С # це можна зробити одним з двох способів. Символи / \* починають коментар, який закінчується символами \*/ Це особливо корисно для багаторядкових коментарів і вилучення частин програми при редагуванні. Символи // починають коментар, що закінчується в кінці рядка, на якому вони з'явилися. Цей спосіб найбільш корисний для коротких коментарів.

Масивом називається впорядкована індексована сукупність однотипних елементів, що мають загальне ім’я. Кожний елемент масиву визначається ім’ям масиву й індексом. Оголошення масиву в С#:

*int[] a, b, c; // оголошення трьох одновимірних масивів*

*double[] x= {5.5, 6.6, 7.7}; //оголошення і ініціалізація масиву з трьома елементами*

*int[] d= new int[5]; //оголошення і ініціалізація пустого масиву з п’ятьма елементами.*

Вивід масиву відбувається за допомогою оператора циклу for:

*int[] testArray = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };*

*for (int a = 0; a <= 10; a++ )*

*{*

*Console.WriteLine(testArray[a]);*

*}*

Синтаксис операторів циклів та умови такі самі як у С++.

Умовний перехід можна реалізувати в програмі за допомогою ключових слів мови: if, else або switch. Такий перехід можливий лише за умови, якщо він є істинним.

if ... else – це оператор розгалуження, робота якого визначається умовою. Умова оператора аналізується інструкцією if. Якщо умова вірна (true), то виконується блок програми, описаних після умови  
if (expression) statement 1  
[else statement2]

Він показує,що робота умовного оператора визначається булевим виразом (вираз, який маєз начення true або false) в круглих дужках. Якщо значення цього виразу істинне, то виконується блок інструкцій statement1.  В іншому випадку виконується блок statement2.

|  |
| --- |
| *int valueOne = 50;*  *int valueTwo = 5;*  *if (valueOne > valueTwo)*  *Console.WriteLine("valueOne: {0} більше чим valueTwo: {1}", valueOne, valueTwo);*  *else*  *Console.WriteLine("valueTwo: {0} більше чи рівне valueOne: {1}", valueTwo, valueOne);*  *valueTwo = 50;*  *if (valueOne > valueTwo)*  *{*  *Console.WriteLine("valueOne: {0} більше чим valueTwo: {1}", valueOne, valueTwo);*  *}*  *valueOne = valueTwo;*  *if (valueOne == valueTwo)*  *{*  *Console.WriteLine("valueOne і valueTwo рівні: {0}=={1}", valueOne, valueTwo);*  *}* |

Оператор if  може застосовувати декілька інструкцій,об'єднаних арифметичними операторами.  Використовуються оператори (&& - І), (||-АБО) і (! - НЕ).

|  |
| --- |
| *int n1 = 5;*  *int n2 = 0;*  *if ((n1 == 5) && (n2 == 5))*  *Console.WriteLine("Інструкція І вірна");*  *else*  *Console.WriteLine("Інструкція І не вірна");*  *if ((n1 == 5) || (n2 == 5))*  *Console.WriteLine("Інструкція АБО вірна");*  *else*  *Console.WriteLine("Інструкція АБО не вірна");*  *Console.Read();* |

Приклад оператора swich:

*switch ( myValue )   
{   
    case 100:   
        Console.WriteLine("Змінна myValue рівна 100");   
    break;   
    case 200:   
        Console.WriteLine("Змінна myValue рівна 200");   
    break;   
        case 300: Console.WriteLine("Змінна myValue рівна 300");   
    break;   
}*

Цикл — різновид керівної конструкції у високорівневих мовах програмування, призначена для організації багаторазового виконання набору [інструкцій](http://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%86%D1%96%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) (команд).

Цикл з передумовою

*using System;   
public class WhileCycle   
{   
    public static int Main()   
    {   
        int i = 0;   
        while (i < 10)   
        {   
            Console.WriteLine("i: {0}", i);   
            ++i;   
        }   
        return 0;   
    }   
}*

Цикл з після умовою

*using System;   
public class DoWhile   
{   
    public static int Main()   
    {   
        int i = 0;   
        do   
        {   
            Console.WriteLine("i : {0}", i);   
            ++i;   
        }   
        while (i < 10);   
        return 0;   
    }   
}*

Цикл з параметром

using System;   
  
public class ForCycle   
{   
    public static int Main()   
    {   
        for (int i = 0; i < 10; i++)   
        {   
            Console.WriteLine("i: {0}", i);   
        }   
        return 0;   
    }   
}

Клас Convert служить для конвертіціі різних типів. Ось так, наприклад, можна сконвертувати строкове представлення числа або Булевського значення в саме число або навпаки:

*string s = "1234";*

*int a = Convert.ToInt32 (s);*

*bool b = Convert.ToBoolean ("true");*

У змінні a і b запишуться число 1234 і значення true відповідно.

Клас Math з простору імен System містить статичні методи для роботи з тригонометричними, логарифмічними та іншими математичними функціями. Практично всі функції перевантажені і приймають різні типи параметрів, наприклад:double, float, int і тд  
Розглянь найбільш часто використовувані:

Math.Abs(<>); Модуль

Math.Ceiling (<>); Округлення до більшого цілого

Math.Floor(<>); Округлення до меншого цілого

Math.Cos (<>); Косинус

Math.E Число е

Math.Exp (<>); Експонента

Math.Log(<>); Натуральний логарифм

Math.Log10(<>); Десятковий логарифм

Math.Max(<>,<>); Максимум із 2 змінних

Math.Min (<>,<>); Мінімум із 2 змінних

Math.Pi Число пі

Math.Pow(<>,<>); Степінь

Math.Round(<>); Просте заокруглення

Math.Sing(<>); Знак числа

Math.Sin(<>); Синус

Math.Sqrt(<>); Квадратний корінь

Math.Tan(<>); тангенс

Приклад використання класу Math:

*namespace D\_Math*

*{*

*class Class1*

*{*

*[STAThread]*

*static void Main(string[] args)*

*{*

*// Оголошення змінних*

*int num1 = 15;*

*int num2 = 10;*

*// Математичнв дії*

*int addAnswer = num1 + num2;*

*int subAnswer = num1 - num2;*

*int mulAnswer = num1 \* num2;*

*int divAnswer = num1 / num2;*

*int modAnswer = num1 % num2;*

*int bigAnswer = (num1 - num2) \* num2 + num1;*

*// Вивід результату*

*Console.WriteLine("number1 == {0}, и number2 == {1}...", num1, num2);*

*Console.WriteLine("Результат додавання " + addAnswer);*

*Console.WriteLine("Результат віднімання: " + subAnswer);*

*Console.WriteLine("Результат множення: " + mulAnswer);*

*Console.WriteLine("Результат ділення: " + divAnswer);*

*Console.WriteLine("Залишок від ділення: " + modAnswer);*

*Console.WriteLine("Результат \"(num1 - num2) \* num2 + num1\": " + bigAnswer);*

*// Порівняння чисел*

*Console.WriteLine("Найбільше число: " + Math.Max(num1, num2));*

*Console.WriteLine("Найменше число: " + Math.Min(num1, num2));*

*// Заокруглення*

*double num3 = 94.537;*

*Console.WriteLine(Environment.NewLine + "Оригінальне число:\t" + num3);*

*Console.WriteLine("Заокруглення до...");*

*Console.WriteLine("2 числа після коми:\t" + Math.Round(num3, 2));*

*Console.WriteLine("1 числа після коми:\t" + Math.Round(num3, 1));*

*Console.WriteLine("0 числа після коми:\t" + Math.Round(num3, 0));*

*// Пі*

*Console.WriteLine(Environment.NewLine + "Пі = " + Math.PI);*

*// Степінь*

*Console.WriteLine(Environment.NewLine + "2 в степені 1 = " + Math.Pow(2, 1));*

*Console.WriteLine("2 в степені 2 = " + Math.Pow(2, 2));*

*Console.WriteLine("2 в степені 3 = " + Math.Pow(2, 3));*

*Console.WriteLine("2 в степені 4 = " + Math.Pow(2, 4));*

*Console.ReadLine();*

*}*

*}*

*}*

**Приклад виконання лабораторної роботи.**

**Задача.**

Дано двовимірний масив , , . Обчислити суму кожного рядка масиву.

**Код програми:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int i,j,s;

int N = 4;

int M = 4;

int[,] a= new int[N,M];

s = 0;

for (i=0;i<N;i++)

for (j = 0; j < M; j++)

{

Console.Write("Enter a["+i.ToString()+","+j.ToString()+"]=");

a[i,j]=Convert.ToInt16(Console.ReadLine());

}

for (i = 0; i < N; i++)

{

for (j = 0; j < N; j++)

{

s = s + a[i, j];

}

Console.WriteLine("s"+i.ToString()+"="+s);

s = 0;

}

Console.ReadKey();

}

}

}

**Результат виконання програми:**

****

**Варіанти завдань до лабораторної роботи**

Утворити двовимірний масив і вивести його на друк вигляді матриці. Виконати завдання вказане у варіанті .

1. . Обчислити суму елементів парних стовпців.

2. . Обчислити добуток елементів парних рядків.

3. . Другий рядок матриці замінити нулями.

4. . Третій стовпець матриці замінити одиницями.

5. . Побудувати вектор з найбільших значень кожного рядка матриці.

6.. Побудувати вектор з найменших значень кожного стовпця матриці.

7. . Обчислити кількість додатних елементів у кожному стовпці матриці.

8. . В матриці поміняти місцями другий і третій рядки.

9. . В матриці поміняти місцями третій і четвертий стовпці.

10. . Знайти суму кожного рядка і всі додатні елементи рядка замінити відповідною сумою.

11. . Четвертий рядок матриці замінити першим стовпцем.

12.. Другий рядок матриці замінити найбільшим елементом першого стовпця.

13. . Третій стовпець матриці замінити одиницями.

14. . Обчислити .

15. , . Побудувати матрицю В, перші два рядки якої складаються з перших двох рядків матриці Х, наступні два рядки – з двох останіх рядків матриці У.

16. , . Обчислити .

17. , . Помножити матрицю Хна третій стовпець матриці У.

18. . З третього стоваця матриці Х утворити вектор У. Знайти ХУ.

19. , . Другий рядок матриці Х замінити четвертим стовпцем матриці У.

20. . Третій стовпець матриці Х збільшити вдвоє.

21. , . Обчислити ХУ.

22. . Утворити вектор з сум елементів кожного стовпця матриці Х.

23. , . Утворити матрицю В, перші два стовпці якої складаються з перших двох стовпців матриці Х. Наступні два стовпці - з третього і четвертого стовпців матриці У.

24. . Утворити два вектори: перший вектор складається з другого рядка матриці Х, другий вектор - з четвертого стовпця матриці Х.

25. . Обчислити добуток матриці Х на третій стовпець цієї ж матриці.

26. . Обчислити .

27. . Поміняти місцями третій і четвертий стовпці матриці Х.

28. . Поміняти місцями перший і третій рядки матриці Х.

29. . Утворити матрицю В. Всі парні рядки матриці В відповідають парним рядкам матриці Х. Всі непарні рядки відповідають непарним рядкам матриці У.

30. . Утворити матрицю В з парних рядків матриці Х.