

MỤC LỤC

BÀI 1. LÀM QUEN VỚI NGÔN NGỮ LẬP TRÌNH C#	2
1.1. Cấu trúc chương trình C#.....	2
1.2. Các kiểu dữ liệu cơ sở.....	2
1.3. Khai báo biến	3
1.4. Chuyển đổi dữ liệu số sang kiểu chuỗi	3
1.5. Chuyển đổi kiểu dữ liệu chuỗi sang kiểu int, long....	3
1.6. Khai báo hằng số.....	3
1.7. Nhập xuất dữ liệu với chế độ console.....	4
1.8. Các cấu trúc điều khiển chương trình	4
1.9. Các phép toán.....	7
1.10. Các hàm toán học thường dùng	7
1.11. Mảng một chiều	7
1.12. Mảng hai chiều	8
1.13. Tạo project dạng Console	10
1.14. Tạo project dạng Windows Form	10
1.15. Cấu trúc một chương trình Windows Forms	11
1.16. Lập trình giao diện đơn giản.....	12
1.17. Bài tập thực hành	13

Trường ĐH CNTT TP.HCM Khoa: Công nghệ thông tin Bộ môn: Hệ thống thông tin Môn: TH Mã hóa và ứng dụng	BÀI 1. LÀM QUEN VỚI NGÔN NGỮ LẬP TRÌNH C#	
--	--	--

A. MỤC TIÊU

Sau khi học xong bài này người học có khả năng:

- Làm quen với ngôn ngữ lập trình C# và IDE Microsoft Visual Studio.
- Cài đặt và thực thi được các chương trình đơn giản ở chế độ Console và Winform.
- Cài đặt được chương trình sử dụng các kiểu dữ liệu cơ bản và cấu trúc điều khiển trong C#.

B. NỘI DUNG

1.1. Cấu trúc chương trình C#

Chương trình C# gồm một hoặc nhiều lớp (class) trong đó phải có một lớp chứa hàm Main. Hàm Main là một phương thức tĩnh của một lớp trong chương trình và là điểm bắt đầu của chương trình.

```

1  using System;
2      using System.Collections.Generic;
3      using System.Linq;
4      using System.Text;
5      using System.Threading.Tasks;
6
7  namespace Test1
8  {
9      0 references
10     class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             Console.WriteLine("Hello world!");
16         }
17     }

```

1.2. Các kiểu dữ liệu cơ sở

Kiểu	Số byte	Mô tả
byte	1	Số nguyên dương không dấu từ 0-255
char	2	Ký tự Unicode
bool	1	Giá trị logic true/ false

Kiểu	Số byte	Mô tả
sbyte	1	Số nguyên có dấu (từ -128 đến 127)
short	2	Số nguyên có dấu giá trị từ -32768 đến 32767.
ushort	2	Số nguyên không dấu 0 – 65.535
Int	4	Số nguyên có dấu –2.147.483.647 và 2.147.483.647
uint	4	Số nguyên không dấu 0 – 4.294.967.295
float	4	Kiểu dấu chấm động, giá trị xấp xỉ từ 3,4E-38 đến 3,4E+38.
double	8	Kiểu dấu chấm động có độ chính xác gấp đôi, giá trị xấp xỉ từ 1,7E-308 đến 1,7E+308.
decimal	8	Có độ chính xác đến 28 con số và giá trị thập phân, được dùng trong tính toán tài chính, kiểu này đòi hỏi phải có hậu tố “m” hay “M” theo sau giá trị .
Long	8	Kiểu số nguyên có dấu có giá trị trong khoảng : - 9.223.370.036.854.775.808 đến 9.223.372.036.854.775.807
ulong	8	Số nguyên không dấu từ 0 đến 0xffffffffffffffff

1.3. Khai báo biến

```
KieuDuLieu TenBien;
KieuDuLieu TenBien = GiaTri;
```

Ví dụ:

```
int a;
int b = 1;
```

1.4. Chuyển đổi dữ liệu số sang kiểu chuỗi

```
String str= TenBien.ToString();
```

Ví dụ:

```
int a = 1;
String str = a.ToString();
```

1.5. Chuyển đổi kiểu dữ liệu chuỗi sang kiểu int, long....

```
int int.Parse(string str)
long long.Parse(string str)
float float.Parse(string str)
double double.Parse(string str)
char[] array =str.ToCharArray();
```

1.6. Khai báo hằng số

```
const KieuDuLieu TENHANGSO = GiaTri;
```

1.7. Nhập xuất dữ liệu với chế độ console

1.7.1. Nhập liệu với màn hình console

```
string str = Console.ReadLine();
```

1.7.2. Xuất ra màn hình console

```
Sử dụng Console.Write (...) hay Console.WriteLine (...)  
Console.Write("Hello World");  
String s = "Hello World";  
Console.WriteLine(s);
```

Định dạng chuỗi xuất với cấu trúc:

```
String.Format("chuỗi định dạng", tham số 1, ..., tham  
số n);
```

Ví dụ:

```
String s = "Hello World";  
int n = 100;  
double m = 20.8;  
string str = string.Format("n={0}, m={1} , s={2}",  
n, m, s);  
Console.WriteLine(str);
```

1.8. Các cấu trúc điều khiển chương trình

1.8.1. Cấu trúc rẽ nhánh với if...else

Cú pháp:

```
if (biểu thức điều kiện)  
{  
    ...  
}  
else  
{  
    ...  
}
```

Ví dụ:

```
Console.Write("d=");  
String str = Console.ReadLine();  
double d = double.Parse(str);  
if (d > 0)  
{  
    Console.WriteLine("{0}> 0", d);  
}  
else  
{  
    Console.WriteLine("{0} <= 0", d);  
}
```

1.8.2. Cấu trúc rẽ nhánh với switch...case

Cú pháp:

```
switch (giá trị){  
    case giá trị 1:  
        ...  
        break;  
        ...  
    case giá trị N:  
        ...  
        break;  
    default:  
        ...  
        break;  
}
```

Ví dụ:

```
char grade = 'B';  
switch (grade)  
{  
    case 'A':  
        Console.WriteLine("Excellent!");  
        break;  
    case 'B':  
    case 'C':  
        Console.WriteLine("Well done");  
        break;  
    case 'D':  
        Console.WriteLine("You passed");  
        break;  
    case 'F':  
        Console.WriteLine("Better try again");  
        break;  
    default:  
        Console.WriteLine("Invalid grade");  
        break;  
}
```

1.8.3. Cấu trúc lặp với while

Cú pháp:

```
while (điều kiện lặp) {  
    ...  
}
```

Ví dụ:

```

Console.Write("n=");
String str = Console.ReadLine();
int n = int.Parse(str);
int s = 0;
int i = 1;
while (i <= n)
{
    s = s + i;
    i++;
}
Console.WriteLine("S=" + s.ToString());

```

1.8.4. Cấu trúc lặp với do...while**Cú pháp:**

```

do{
...
}while (điều kiện lặp);

```

Ví dụ:

```

int i = 0;
Console.Write("Cac so nho hon 5: ");
do
{
    Console.Write("{0} ", i);
    i++;
} while ( i < 5 ) ;
// Kết quả: Cac so nho hon 5 : 0 1 2 3 4

```

1.8.5. Cấu trúc lặp với for**Cú pháp:**

```

for (khởi tạo biến; điều kiện lặp; hành động sau mỗi lần lặp)
{
    ...
}

```

Ví dụ:

```

Console.Write("Cac so chan nho hon 5: ");
for (int i = 0; i <= 5 ; i+=2)
{
    Console.WriteLine(i.ToString() + " ");
}
// Kết quả: Cac so chan nho hon 5 : 0 2 4

```

1.8.6. Lệnh break và continue

- Được sử dụng trong for, while, do-while với mục đích thoát khỏi vòng lặp
- Được sử dụng trong for, while, do-while với mục đích bỏ qua bước lặp hiện tại, quay trở lại vòng lặp để thực hiện tiếp.

Ví dụ:

```
for (int i = 0; i < 5; i++)
{
    if (i == 3)
        break;
    Console.Write("{0} ", i);
}
// Kết quả: 0 1 2
for (int i = 0; i < 5; i++)
{
    if (i == 3)
        continue;
    Console.Write("{0} ", i);
}
// Kết quả: 0 1 2 4
```

1.9. Các phép toán

- Các phép toán số học: +, -, *, /, %, =, +=, -=, *=, /=, %=
- Các phép toán so sánh: ==, !=, >, >=, <, <=
- Các phép toán luận lý: &&, ||, !

1.10. Các hàm toán học thường dùng

```
double Math.Abs (double d)
double Math.Sqrt (double d)
double Math.Min (double a, double b)
double Math.Max (double a, double b)
double Math.Exp (double e)
double Math.Round (double r)
double Math.Pow (double a, double b)
double Math.PI
double Math.E
```

1.11. Mảng một chiều

a) Khai báo

```
KieuDuLieu[ ] tenBien;
```

Ví dụ:

```
int[] arr1; //Khai báo mảng kiểu int
long[] arr2; //Khai báo mảng kiểu long
float[] arr3; //Khai báo mảng kiểu float
double[] arr4; //Khai báo mảng kiểu double
bool[] arr5; //Khai báo mảng kiểu boolean
String[] arr6; //Khai báo mảng kiểu string
```

b) Cấp phát vùng nhớ

```
KieuDuLieu[] tenBien = new KieuDuLieu [n] ;
```

Hoặc

```
KieuDuLieu[] tenBien;  
tenBien = new KieuDuLieu [n];
```

Ví dụ:

```
int[] arr1 = new int[5]; // Khai báo và cấp phát mảng kiểu  
int , a.Length = 5  
long[] arr2 = new long[5]; // Khai báo và cấp phát mảng  
kiểu long , a.Length = 5  
float[] arr3 = new float[7]; // Khai báo và cấp phát mảng  
kiểu float, a.Length = 7  
String[] arr6 = new String[6]; // Khai báo và cấp phát mảng  
kiểu string, a.length = 6
```

c) Nhập mảng

```
Console.Write("Số phần tử của mảng là ");  
int n = int.Parse(Console.ReadLine());  
int[] a = new int[n]; //a.Length = n  
for (int i = 0; i < a.Length; i++)  
{  
    Console.Write("a[" + i.ToString() + "]=");  
    a[i] = int.Parse(Console.ReadLine());  
}
```

d) Xuất mảng

```
Console.WriteLine("Số phần tử của mảng {0}", a.Length);  
//Xuất mảng dùng for  
for (int i = 0; i < a.Length; i++) {  
    Console.WriteLine(a[i]);  
}
```

1.12. Mảng hai chiều

a) Khai báo

```
KieuDuLieu[,] tenBien;
```

Ví dụ:

```
//Khai báo mảng hai chiều kiểu int  
int[,] arr1;  
//Khai báo mảng hai chiều kiểu long  
double[,] arr4;  
//Khai báo mảng hai chiều kiểu boolean  
bool[,] arr5;
```


b) Cấp phát vùng nhớ

```
KieuDuLieu[,] tenBien = new KieuDuLieu [n,m];
```

Hoặc

```
KieuDuLieu[,] tenBien;  
tenBien = new KieuDuLieu [n,m];
```

Ví dụ:

```
//Khai báo và cấp phát mảng hai chiều kiểu int  
int[,] arr1 = new int[3, 5];  
//soDong = 3 ; soCot = 5  
//Khai báo và cấp phát mảng kiểu long  
long[,] arr2 = new long[5, 6];  
//soDong = 5 ; soCot = 6  
//Khai báo và cấp phát mảng hai chiều kiểu float  
float[,] arr3 = new float[7, 9];  
//soDong = 7 ;soCot = 9
```

c) Nhập mảng

```
int dong, cot;  
Console.Write("So dong:");  
dong = int.Parse(Console.ReadLine());  
Console.Write("So cot:");  
cot = int.Parse(Console.ReadLine());  
int[,] arr = new int[dong, cot];  
//Nhap gia tri cho mang  
for (int i = 0; i < dong; i++)  
{  
    for (int j = 0; j < cot; j++)  
    {  
        Console.Write("arr[{0}][{1}]:", i , j );  
        arr[i, j] = int.Parse(Console.ReadLine());  
    }  
}
```

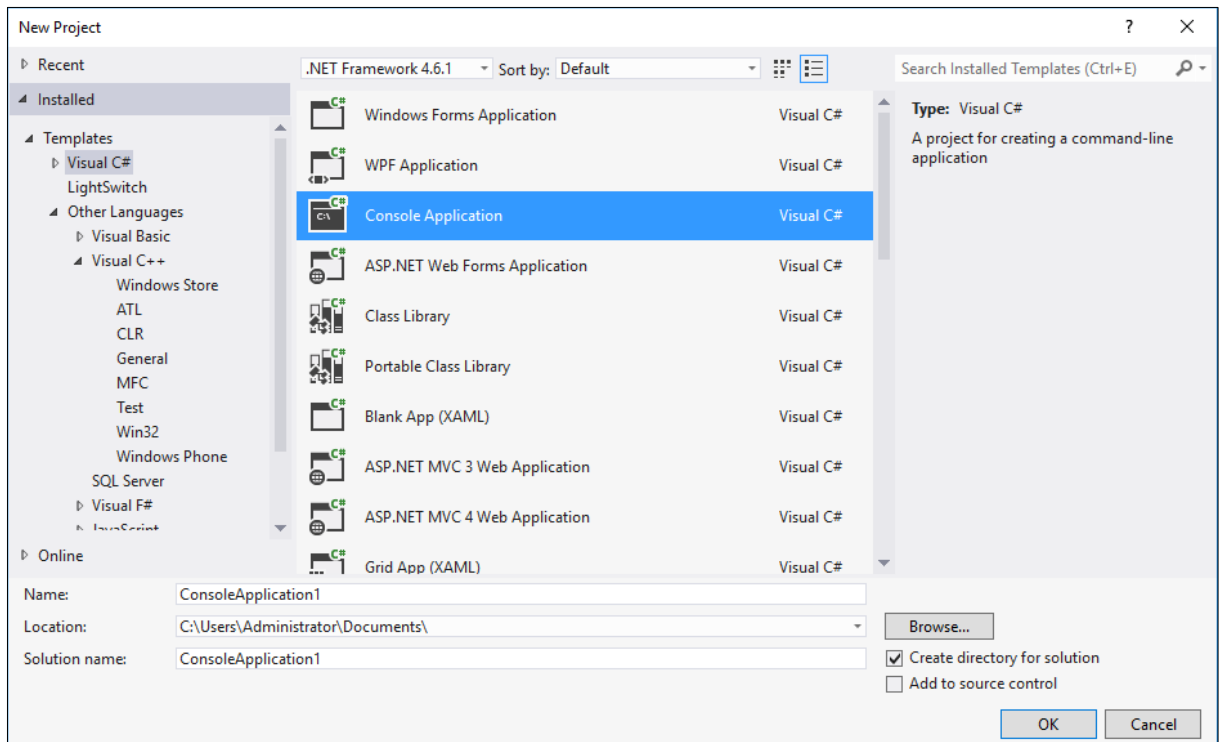
d) Xuất mảng

```
//Xuat mang da nhap  
for (int i = 0; i < dong; i++)  
{  
    for (int j = 0; j < cot; j++)  
    {  
        Console.WriteLine("arr[{0}][{1}]={2}", i,  
j,arr[i,j]);  
    }  
}
```

1.13. Tạo project dạng Console

–**Bước 1:** Khởi động Ms Visual Studio

–**Bước 2:** Vào menu *File > New > Project*

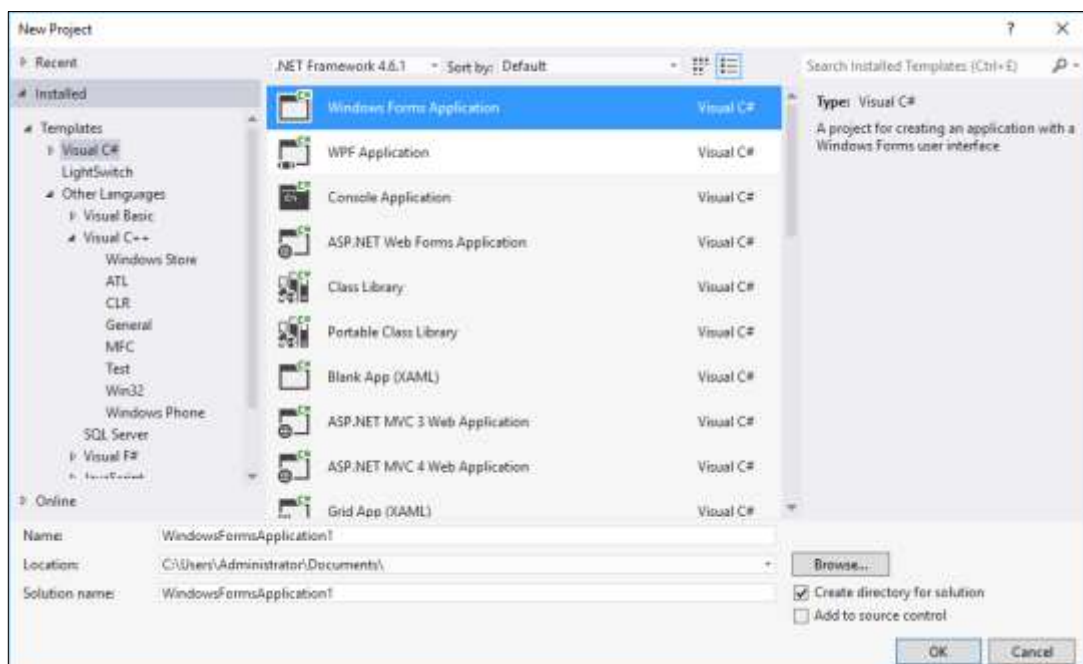


–**Bước 3:** Chọn ngôn ngữ lập trình C#, chọn loại ứng dụng *Console Application* (ứng dụng với giao diện console), thiết lập project name, solution name, location → OK.

1.14. Tạo project dạng Windows Form

–**Bước 1:** Khởi động Ms Visual Studio

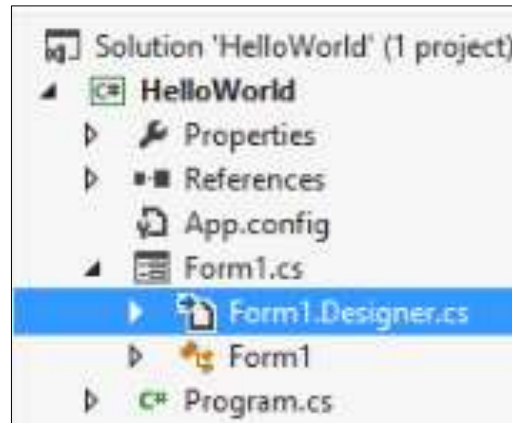
–**Bước 2:** Vào menu *File > New > Project*



–Bước 3: Chọn ngôn ngữ lập trình C#, chọn loại ứng *Windows Forms Application* (ứng dụng với giao diện đồ họa), thiết lập project name, solution name, location → OK.

1.15. Cấu trúc một chương trình Windows Forms

Khi vừa tạo xong, một chương trình Windows Forms sẽ bao gồm các thành phần quan trọng sau:



- Tập tin Program.cs: đây là mã nguồn chứa hàm main của chương trình

```
static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Application.Run(new Form1());
}
```

Mặc định, mã nguồn hàm main sẽ khởi tạo chương trình với giao diện Form1. Các tập tin xử lý Form1 bao gồm:

- **Form1.cs**: chứa mã nguồn xử lý do người dùng định nghĩa thêm. Mặc định, ban đầu Form1.cs

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }
}
```

- **Form1.designer.cs**: chứa mã nguồn về giao diện. Tại đây, từ khoá partial cho phép định nghĩa lớp Form1 được đặt trong cả 2 tập tin Form1.cs và Form1.designer.cs. Lúc này, mã nguồn giao diện chưa có phần nào thêm. Phần mã nguồn trong mục “Windows Forms Designer generated code” thể hiện phần mã nguồn giao diện Form1 do Visual Studio tự phát sinh.

```

partial class Form1
{
    private System.ComponentModel.IContainer components = null;

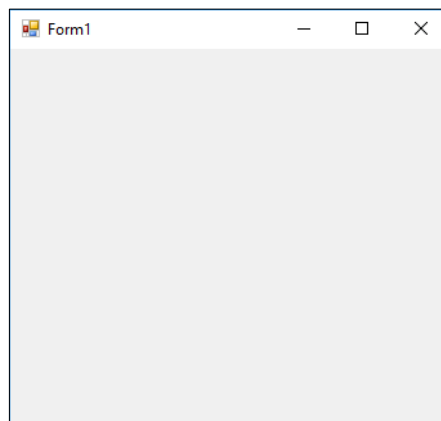
    protected override void Dispose(bool disposing)
    {
        if (disposing && (components != null))
        {
            components.Dispose();
        }
        base.Dispose(disposing);
    }

    #region Windows Form Designer generated code

    private void InitializeComponent()
    {
        this.components = new System.ComponentModel.Container();
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.Text = "Form1";
    }

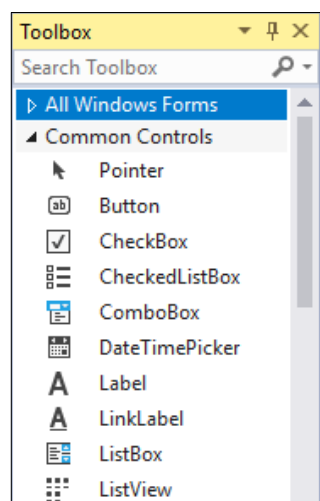
    #endregion
}

```



1.16. Lập trình giao diện đơn giản

- Sử dụng View/Toolbox để hiển thị các controls có trong Windows Forms.



- Lần lượt kéo thả các control vào giao diện Form1, ví dụ như sau:

- Viết hàm xử lý sự kiện cho button Xin chào

```
private void btnXinChao_Click(object sender, EventArgs e)
{
    string hoten = txtHoTen.Text;
    MessageBox.Show("Xin chào: " + hoten);
}
```

1.17. Bài tập thực hành

Viết chương trình thực hiện các yêu cầu sau ở dạng **Console Application**:

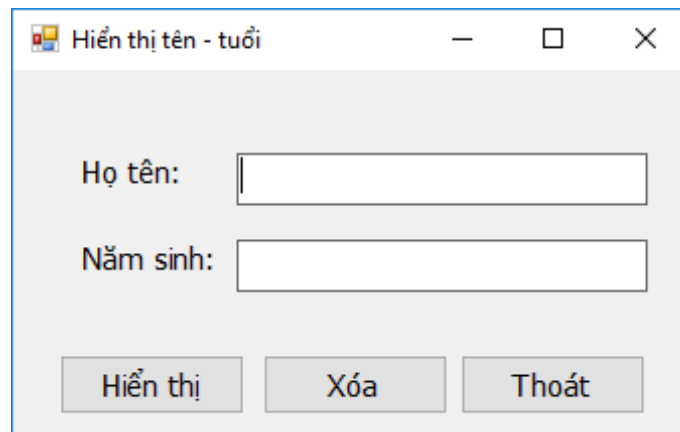
1. Nhập vào một số nguyên là năm, cho biết năm đó có là năm nhuận hay không. Biết rằng: năm nhuận là năm chia hết cho 400 hoặc chia hết cho 4 nhưng không chia hết cho 100.
2. Giải và biện luận phương trình bậc 2 ($ax^2 + bx + c = 0$).
3. Nhập vào một số nguyên là tháng, cho biết tháng đó có bao nhiêu ngày (nếu là tháng 2 thì yêu cầu nhập thêm năm).
4. Nhập vào một số thực là điểm ở thang điểm 10, cho biết giá trị ở thang điểm chữ ABCDF. Biết rằng: 8.5 \rightarrow 10: A, 7.0 \rightarrow 8.4: B, 5.5 \rightarrow 6.9: C, 4.0 \rightarrow 5.4: D, dưới 4: F.
5. Nhập vào chỉ số điện cũ và chỉ số điện mới. Tính tiền điện phải trả trong tháng, dữ liệu nhập phải đảm bảo chỉ số điện cũ \leq chỉ số điện mới. Cách thức tính tiền điện theo quy định sau:
 - Bậc 1: cho các kWh từ 0 đến 50: 1480 VNĐ/kw;
 - Bậc 2: cho các kWh từ 51 đến 100: 1533 VNĐ/kw;
 - Bậc 3: cho các kWh từ 101 đến 200: 1786 VNĐ/kw;
 - Bậc 4: cho các kWh từ 201 đến 300: 2242 VNĐ/kw;
 - Bậc 5: cho các kWh từ 301 trở đi: 2503 VNĐ/kw;
6. Tính tiền thuê phòng khi biết số ngày thuê và loại phòng (một trong 3 loại A, B, hoặc C) theo quy định như sau:
 - Loại A: 500.000 đ/ngày;
 - Loại B: 400.000 đ/ngày;
 - Loại C: 300.000 đ/ngày.

Nếu thuê quá 7 ngày thì được giảm 10% trên tổng số tiền cho phòng loại A, 8% cho phòng loại B hoặc C.

7. Kiểm tra xem ngày (gồm 3 thành phần là ngày, tháng, năm) cho trước có phải là một ngày hợp lệ. Trong đó, một ngày được gọi là hợp lệ khi:
- Năm: số nguyên lớn hơn 0;
 - Tháng: từ 1 đến 12;
 - Ngày từ 1 đến số ngày tối đa của tháng – năm tương ứng.
8. Nhập vào ngày, tháng, năm. Cho biết ngày kế tiếp và ngày kế trước của ngày đó.
9. Nhập vào số nguyên dương n. Cho biết n có phải là số nguyên tố hay không?
10. Nhập 2 số nguyên dương a, b. Tìm ước số chung lớn nhất của a, b.
11. Nhập 2 số nguyên dương a, b. Tìm bội số chung nhỏ nhất của a, b.
12. Cho mảng 1 chiều chứa số nguyên. Viết các hàm thực hiện các yêu cầu sau:
- a) Nhập mảng a gồm n phần tử.
 - b) Xuất mảng a ra màn hình
 - c) Cho biết số phần tử có giá trị chẵn có trong mảng
 - d) Cho biết giá trị nhỏ nhất, lớn nhất trong mảng
 - e) Liệt kê các phần tử có giá trị là số nguyên tố trong mảng
 - f) Tính tổng các phần tử có giá trị dương trong mảng.

Viết chương trình thực hiện các yêu cầu sau ở chế độ **Windows Forms Application**:

13. Hiển thị Họ tên – Tuổi

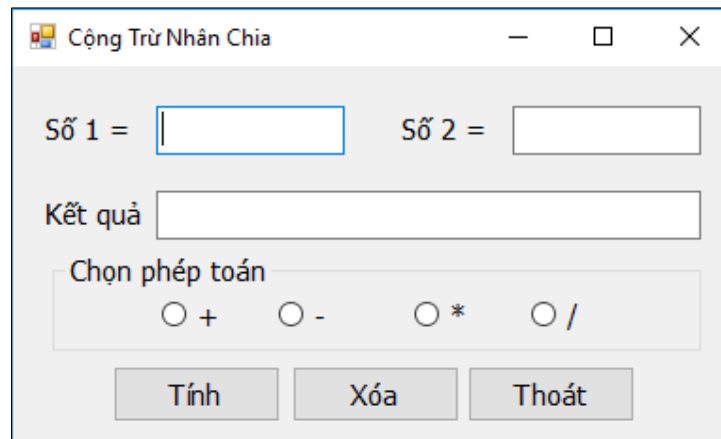


The screenshot shows a standard Windows Forms application window. The title bar reads 'Hiển thị tên - tuổi'. The main area has a light gray background. There are two text boxes stacked vertically. The first is labeled 'Họ tên:' and the second is labeled 'Năm sinh:'. Below these text boxes, there are three buttons arranged horizontally: 'Hiển thị', 'Xóa', and 'Thoát'.

Yêu cầu:

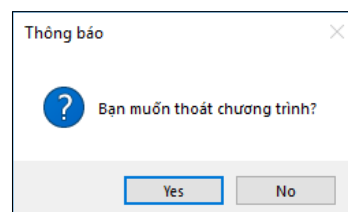
- Nút **Hiển thị**: hiện thông báo tên và tuổi (năm hiện tại – năm sinh).
- Nút **Xóa**: xóa thông tin họ tên và tuổi trong các TextBox, chuyển focus về textbox họ tên.
- Nút **Thoát**: thoát chương trình

14. Cộng trừ nhân chia 2 số



Yêu cầu:

- Nút **Tính**: hiển thị giá trị trong textbox kết quả tùy vào phép toán được chọn.
- Nút **Xóa**: xóa thông tin trên các textbox, chuyển focus về textbox số 1.
- Nút **Thoát**: hiển thị hộp thoại có 2 nút Yes/No, nếu người dùng click Yes thì thoát chương trình.



15. Thiết kế giao diện và viết code xử lý bài toán giải và biện luận phương trình bậc 2 ($ax^2 + bx + c = 0$).

16. Thiết kế giao diện như sau:

