



1

Lab

Lập trình C# & Windows Forms cơ bản

Introduction to C# & Winform Application

Thực hành Lập trình mạng căn bản

GVHD: Phan Trung Phát

Học kỳ II – Năm học 2022-2023

Lưu hành nội bộ

A. TỔNG QUAN

1. Mục tiêu

- Lập trình với ngôn ngữ C# ở mức cơ bản:
 - Đọc hiểu chương trình C# ở mức cơ bản.
 - Nắm các kiểu dữ liệu trong C#:
 - Kiểu dữ liệu nguyên thủy: int, long, double, char...
 - Kiểu dữ liệu tham chiếu: Array, List, Class...
 - Nắm được các cấu trúc điều khiển trong lập trình:
 - Cấu trúc điều khiển tuần tự.
 - Cấu trúc điều khiển rẽ nhánh if.
 - Cấu trúc điều khiển rẽ nhánh if... else.
 - Cấu trúc điều khiển vòng lặp.
 - Nắm được các kỹ thuật xử lý trên mảng 1 chiều.
- Lập trình C# với Window Forms
 - Có thể lập trình Giao diện ứng dụng với:
 - Giao diện đồ họa (Kéo thả, thay đổi thuộc tính).
 - Code.
 - Có thể lập trình các Ứng dụng Window Forms cơ bản giải quyết các bài toán nhỏ.

2. Môi trường

- IDE Microsoft Visual Studio 2010 trở lên.

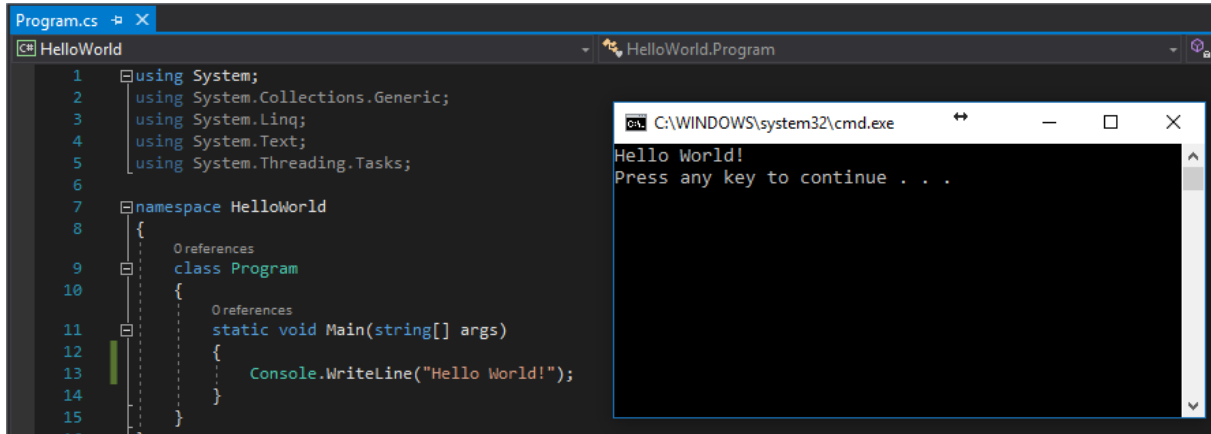
3. Liên quan

- Sinh viên cần nắm được các kiến thức nền tảng về lập trình. Các kiến thức này đã được giới thiệu trong các môn học trước và trong nội dung lý thuyết đã học do đó sẽ không được trình bày lại trong nội dung thực hành này.
- Tham khảo tài liệu (Mục E) để có kiến thức cơ bản về C#, Winforms.

B. VÍ DỤ XÂY DỰNG ỨNG DỤNG C# WINFORM

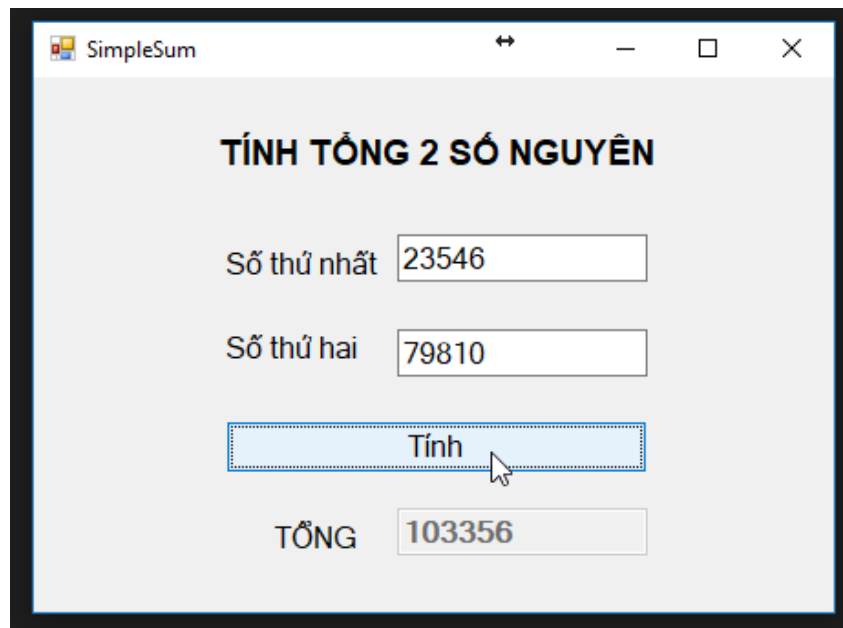
1. Chương trình Hello World (Console App)

Chọn *File > New > Project > Visual C# > Console App* để tạo Project **Console App**.



2. Windows Forms Application

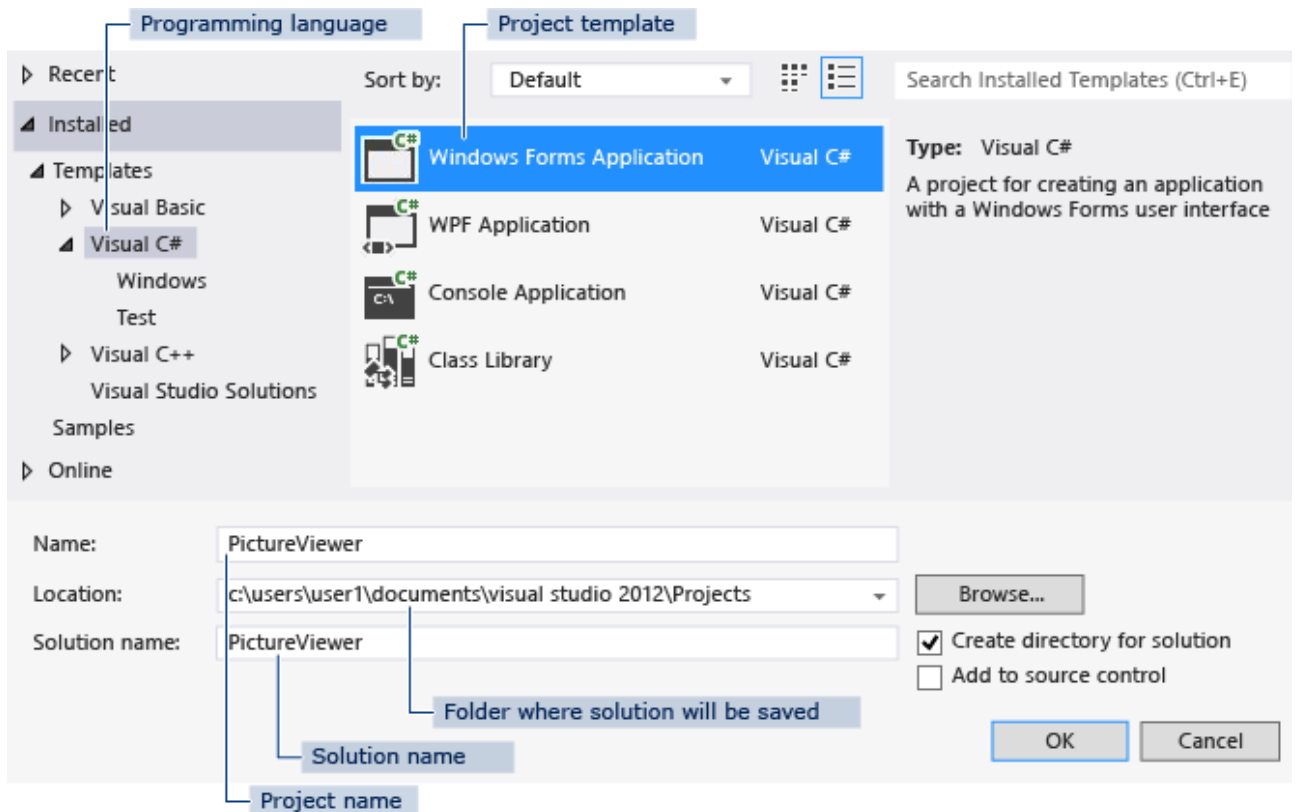
Yêu cầu: Viết chương trình có giao diện đồ họa cho phép nhập 2 số nguyên và tính tổng 2 số đó với giao diện như hình bên dưới.



Hình 1. Chương trình tính tổng 2 số nguyên đơn giản

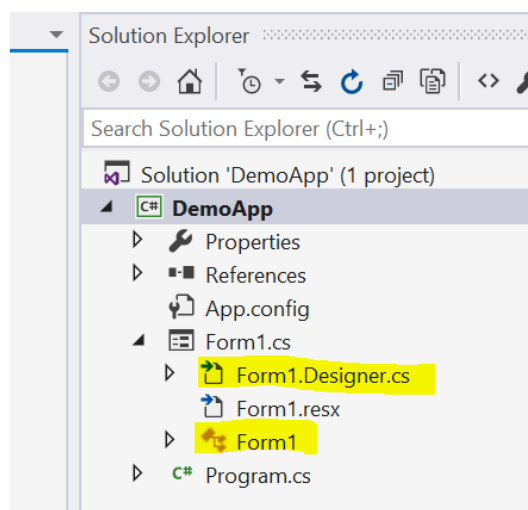
Hướng dẫn:➤ **Tạo Project**

- Chọn *File > New > Project > Visual C# > Windows Forms Application* để tạo Project **Windows Forms App**. Đặt tên Project, Solution, chọn nơi lưu trữ và nhấn OK.

➤ **Thiết kế giao diện (Design)**

- Với Visual Studio, để lập trình ứng dụng Winform sẽ có 3 cửa sổ cơ bản: *Cửa sổ chính*, *Solution Explorer* và *Cửa sổ Properties*. Lưu ý: nếu khuyết mất cửa sổ nào, có thể hiển thị các cửa sổ này bằng cách chọn trên menu bar hoặc dùng tổ hợp phím: **View > Properties Window (F4)**, **View > Solution Explorer (Ctrl + Alt +L)**.
- **Cửa sổ chính**: cửa sổ làm việc chính, cho phép người dùng xem, thay đổi giao diện, mã nguồn của Form.

- **Solution Explorer:** liệt kê và điều hướng đến tất cả các thành phần có trong Solution.
- **Properties:** bằng cách lựa chọn 1 đối tượng (click chọn trên Cửa sổ chính hoặc click chọn trong Solution Explorer), cửa sổ Properties sẽ hiển thị các thuộc tính của đối tượng được chọn. Có thể sử dụng để thay đổi các giá trị thuộc tính của đối tượng cần quan tâm (Form, Control v.v...).
- Có 2 đối tượng chính cần quan tâm trong 1 ứng dụng Window Form: **Form** và **Controls**.
- **Form:** là một giao diện để hiển thị thông tin dữ liệu cho người dùng, thường biểu diễn dưới dạng 1 cửa sổ window.
- **Controls:** là các đối tượng nằm trong 1 Form. Mỗi loại control sẽ mang một tập những thuộc tính đặc trưng, phương thức và sự kiện riêng phù hợp với một mục đích cụ thể. Ví dụ: **button** là 1 loại control có thiết kế dưới dạng nút, cho phép người dùng kích hoạt bằng cách click chọn và có sự kiện đặc trưng riêng là *click*.
- Trong Visual Studio, người dùng có thể dễ dàng thêm 1 control vào Form bằng các cách sau đây: **kéo – thả** hoặc **thêm mã nguồn** vào file thiết kế của Form.
- 1 Form (Giao diện hiển thị) có các file sau:
 - File **.Designer.cs** chứa mã nguồn thiết kế (design) của form.



- File **.cs** chứa mã nguồn các sự kiện, chức năng của form.
- Click chọn Form1.cs[Design] để xem được giao diện thiết kế của Form.

➤ **Thiết kế giao diện (Design) theo yêu cầu của đề bài**

▪ **Thiết kế Form1**

- **Thay đổi tên Form:** dùng chuột click chọn vào cửa sổ của Form1, mở cửa sổ Properties, tại Trường Text gõ tên Form muốn đổi. Ví dụ: *Simple Sum*

- **Tại cửa sổ Properties** có thể thử thay đổi 1 số trường để thấy sự thay đổi về giao diện. Ví dụ: *Size* (Kích thước Form), *BackColor* (Màu nền), *Opacity* (Độ trong suốt) v.v...

▪ **Thêm Control (đối tượng) hiển thị tiêu đề của ứng dụng:**

- *Kích hoạt cửa sổ Toolbox:* chọn **View >**

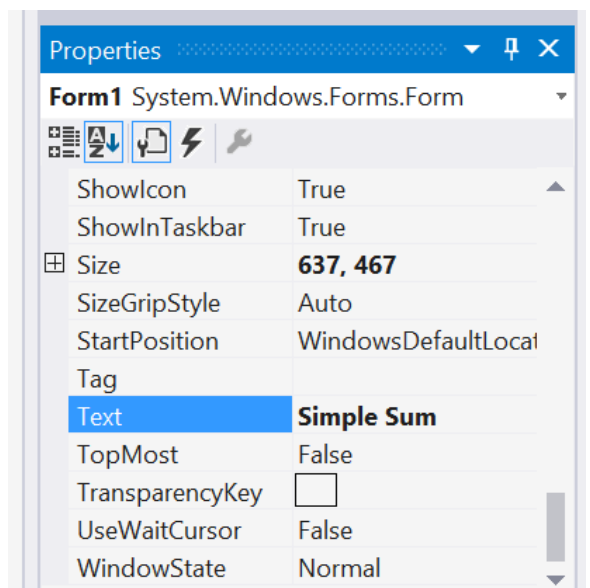
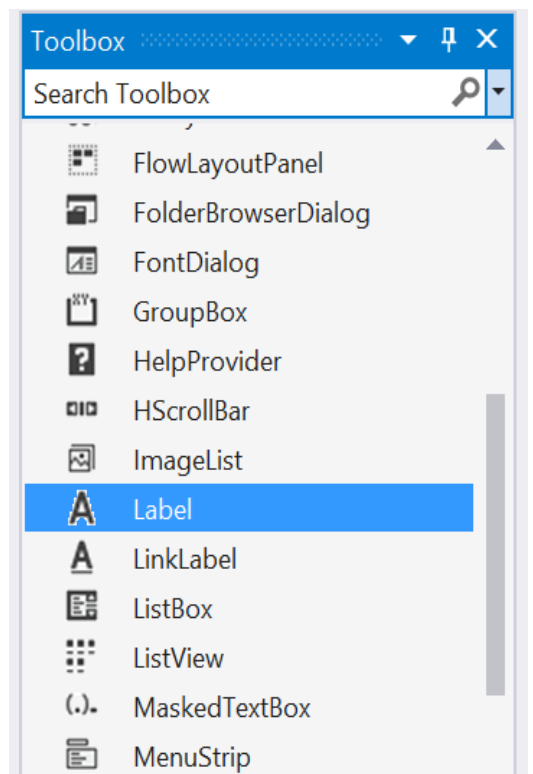
Toolbox hoặc dùng tổ hợp phím **Ctrl + Alt + X**


- *Thêm 1 nhãn "TÍNH TỔNG 2 SỐ NGUYÊN":* Từ Toolbox, kéo 1 Label và thả vào giao diện Form. Trong cửa sổ Properties, thay đổi trường Text thành "TÍNH TỔNG 2 SỐ NGUYÊN".

▪ **Tương tự như trên, kéo thả các Control phù hợp với yêu cầu của đề.**

- "Số thứ nhất", "Số thứ hai", "TỔNG" sử dụng Control **Label**

- Các ô để nhập số thứ 1 và thứ 2 sử dụng Control **Textbox**



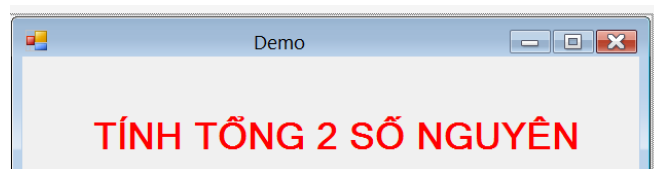
- Ô để hiển thị Kết quả cũng sử dụng Control **Textbox**, tuy nhiên Trường ReadOnly (Properties) được gán giá trị true.
- Nút "Tính" sử dụng Control Button  Button
- Bên cạnh việc sử dụng kéo thả và đổi các trường trong cửa sổ Properties, có thể thêm và chỉnh sửa các đối tượng bằng cách sử dụng mã nguồn (Code).
- Mở File mã nguồn thiết kế của Form (.Designer.cs).

```
Form1.Designer.cs  Form1.cs [Design]  Form1.resx
DemoApp.Form1  InitializeComponent()

/// </summary>
private void InitializeComponent()
{
    this.label1 = new System.Windows.Forms.Label();
    this.label2 = new System.Windows.Forms.Label();
    this.label3 = new System.Windows.Forms.Label();
    this.textBox1 = new System.Windows.Forms.TextBox();
    this.textBox2 = new System.Windows.Forms.TextBox();
    this.button1 = new System.Windows.Forms.Button();
    this.button2 = new System.Windows.Forms.Button();
    this.label4 = new System.Windows.Forms.Label();
    this.textBox3 = new System.Windows.Forms.TextBox();
    this.SuspendLayout();
    //
    // label1
    //
    this.label1.AutoSize = true;
    this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 18F, System.Drawing.FontStyle.I
    this.label1.Location = new System.Drawing.Point(62, 57);
    this.label1.Name = "label1";
    this.label1.Size = new System.Drawing.Size(490, 40);
    this.label1.TabIndex = 0;
    this.label1.Text = "TÍNH TỔNG 2 SỐ NGUYÊN";
    ..
}
```

- Giả sử thêm đoạn code sau để thấy được sự thay đổi.

```
this.label1.ForeColor = System.Drawing.Color.Red;
```



➤ Lập trình sự kiện, chức năng

Với yêu cầu của đề bài là tính tổng 2 số nguyên và xuất kết quả tại ô Tổng. Ta phân tích như sau:

- Để thực hiện hàm tính tổng, người dùng phải ấn nút "Tính", vì thế ta sẽ bắt sự kiện Click cho button "Tính" này.
- Nhấp đúp chọn button "Tính", chương trình sẽ khởi tạo sẵn 1 hàm đặc trưng tương ứng với sự kiện xảy ra khi click chuột vào button "Tính". Ví dụ button_Click.
- Thêm mã nguồn để tính toán và hiển thị kết quả như sau:

```
private void button1_Click(object sender, EventArgs e)
{
    int num1, num2;
    long sum = 0;
    num1 = Int32.Parse(textBox1.Text.Trim());
    num2 = Int32.Parse(textBox2.Text.Trim());
    sum = num1 + num2;
    textBox3.Text = sum.ToString();
}
```

- **Chạy ứng dụng:** nhấn F5 hoặc chọn Debug để chạy chương trình.

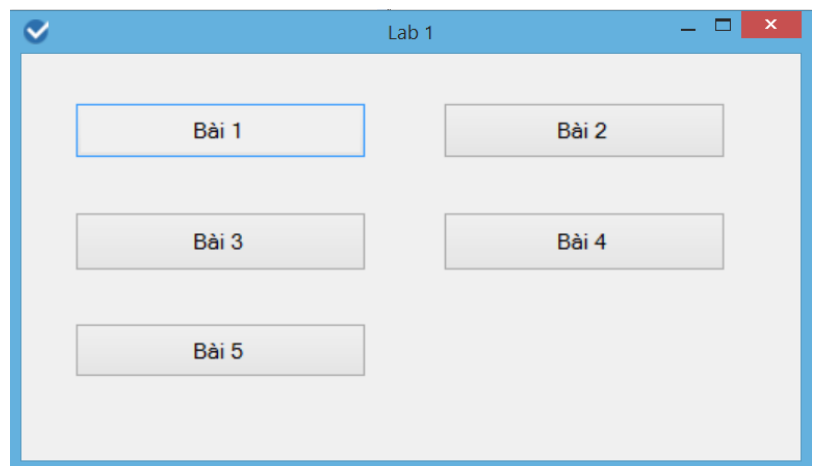
Từ Ví dụ trên, Sinh viên hãy tham khảo thêm tài liệu và sử dụng Visual Studio để làm các bài tập ở Mục C.

C. BÀI TẬP

Các bài thực hành dưới đây yêu cầu viết chương trình dưới dạng *Windows Forms App*.

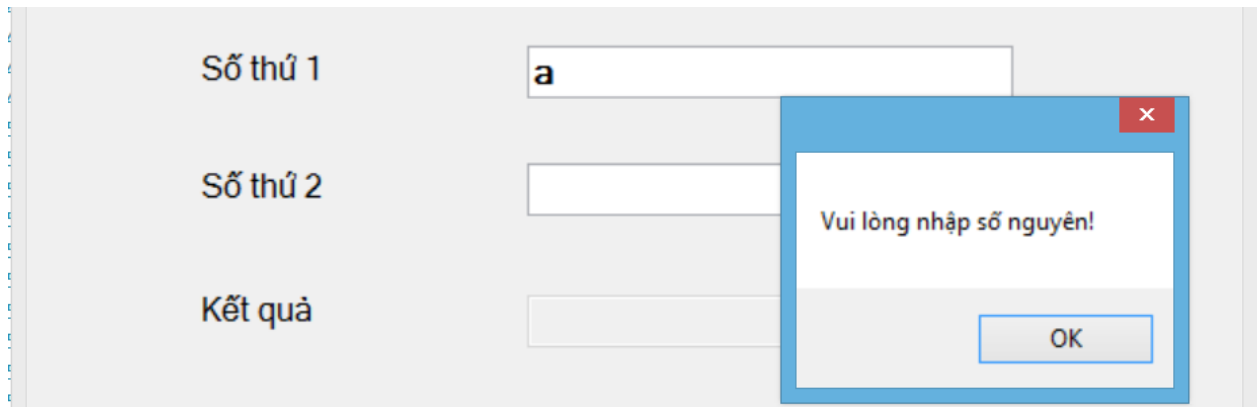
Sinh viên có thể tùy biến cách sắp xếp giao diện khác sao cho hợp lý.

Yêu cầu: Tất cả các bài thực hành đặt chung trong 1 Project duy nhất, mỗi bài tập là 1 Form đặt tên theo dạng **Lab01-BaiXX** (với XX là số thứ tự bài tập) và có Form điều hướng để mở các bài tập.



1. Bài 01 – Tính tổng 2 số nguyên (Có điều kiện)

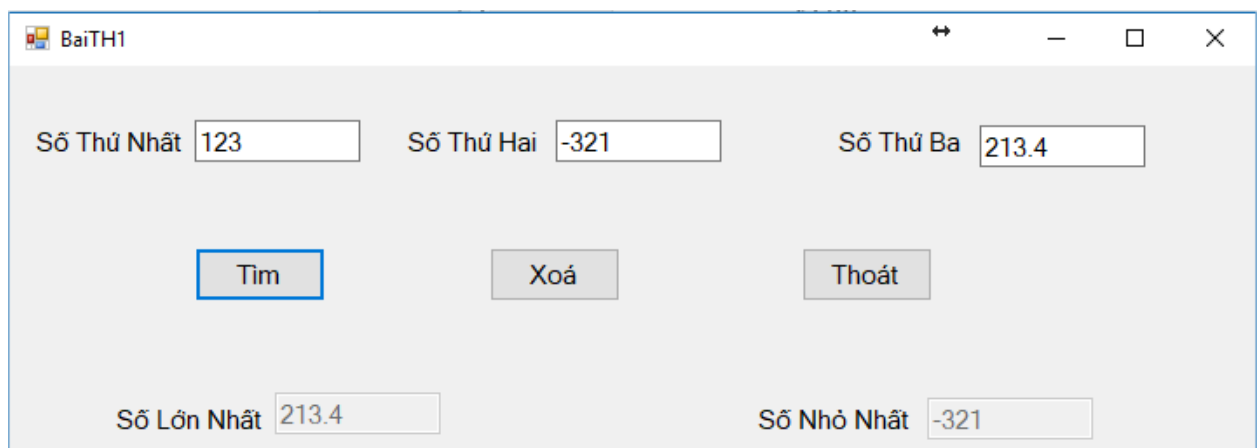
Tương tự Ví dụ Tính tổng 2 số nguyên ở trên, thêm vào tính năng kiểm tra dữ liệu đầu vào hợp lệ hay không. Báo lỗi khi dữ liệu không là số nguyên.



Gợi ý: Sử dụng MessageBox để hiển thị thông điệp cảnh báo.

2. Bài 02 – Số lớn nhất, nhỏ nhất

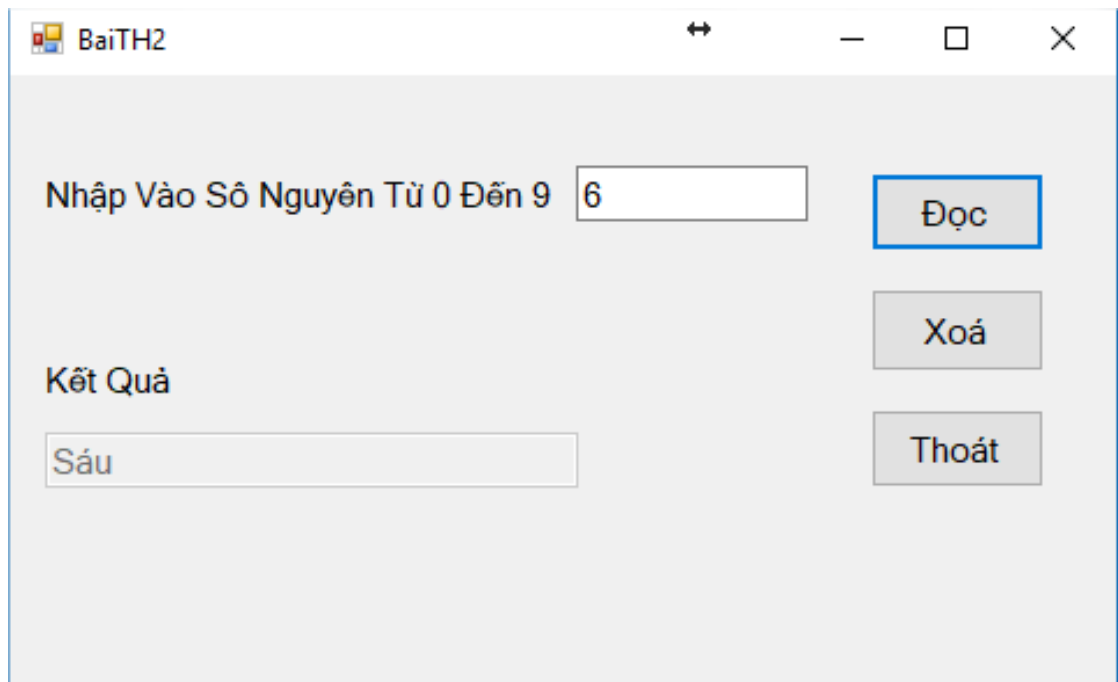
Viết chương trình nhập vào 3 số bất kỳ. Cho biết số lớn nhất và số nhỏ nhất trong 3 số với giao diện ví dụ như sau:



Gợi ý: Sử dụng cấu trúc If – else.

3. Bài 03 – Đọc số

Viết chương trình nhập vào một số nguyên từ 0 đến 9, hiển thị các số trên bằng chữ. Ví dụ: Nhập 1: "Một"; 2: "Hai"; ...; 0: "Không".



Gợi ý: Sử dụng cấu trúc Switch – case.

4. Bài 3.1 – Đọc số nâng cao

Trên cơ sở của bài 03, viết chương trình nhập vào một số có 12 chữ số, hiển thị cách đọc của số đó bằng chữ. Ví dụ: Nhập 123456: "Một trăm hai mươi ba ngàn, bốn trăm năm mươi sáu".

5. Bài 04 – Tính tiền xăng

Dựa vào các thông tin được cung cấp, viết chương trình cho phép người dùng chọn loại xe, loại xăng dầu, tính số km đi được và số tiền cần trả của xe đó khi đổ đầy bình xăng/dầu.

Thông tin mức độ tiêu tốn nhiên liệu của các loại xe:

- Wave Alpha: 1,6 lít/100km, sức chứa 3,7 lít, sử dụng xăng RON 95-III và 92-II.
- Sirius: 1,99 lít/100km, sức chứa 3,8 lít, sử dụng xăng RON 95-III và 92-II.
- Vision: 1,87 lít/100km, sức chứa 5,2 lít, sử dụng xăng RON 95-III và 92-II.
- Lead: 2.02 lít/100km, sức chứa 6 lít, chỉ sử dụng xăng RON 95-III.
- Winner: 1,7 lít/100km, sức chứa 4,5 lít, chỉ sử dụng xăng RON 95-III.
- AirBlade 150: 2,17 lít/100km, sức chứa 4,4 lít, chỉ sử dụng xăng RON 95-III.
- Xe tải 9 tấn: 13 lít/100km, sức chứa 70 lít, chỉ sử dụng dầu DO.

Thông tin tỉ giá xăng dầu:

- Xăng RON 95-III: 26,830 đồng/lít
- Xăng E5 RON 92-II: 26.070 đồng/lít
- Dầu DO 0,05S-II: 21,310 đồng/lít

Gợi ý: Sử dụng Control ComboBox.

6. Bài 05

Viết chương trình cho phép nhập vào 2 số nguyên A và B, tính và xuất kết quả:

- A! và B!
- Tổng **S1** = $1+2+3+4+\dots+A$ và **S2** = $1+2+3+4+\dots+B$
- Tổng **S3** = $A^1 + A^2 + A^3 + A^4 + \dots + A^B$

The screenshot shows a Windows application window with a light gray background. At the top, there are two text input fields labeled "Nhập A" and "Nhập B". The "Nhập A" field contains the number "9" and the "Nhập B" field contains the number "3". Below these fields are three buttons: "Tính các giá trị" (Calculate values), "Xoá" (Delete), and "Thoát" (Exit). The "Thoát" button is highlighted with a blue border. Below the buttons is a section titled "KẾT QUẢ" (Results) in bold. This section contains four lines of text, all in blue: "A! = 362,880", "B! = 6", "S1 = 1 + 2 + 3 + 4 + ... + A = 45", "S2 = 1 + 2 + 3 + 4 + ... + B = 6", and "S3 = A^1 + A^2 + A^3 + A^4 + ... + A^B = 819".

Gợi ý: Sử dụng vòng lặp **for**.

7. Bài 06

Viết chương trình nhập vào ngày tháng năm sinh, xuất ra ngày đó là ngày thứ mấy trong tuần và cung hoàng đạo của bạn là gì?

Cho biết:

- 21/03 – 20/04: cung Bạch Dương
- 21/04 – 21/05: cung Kim Ngưu
- 22/05 – 21/06: cung Song Tử

- 22/06 – 22/07: cung Cự Giải
- 23/07 – 22/08: cung Sư Tử
- 23/08 – 23/09: cung Xử Nữ
- 24/09 – 23/10: cung Thiên Bình
- 24/10 – 22/11: cung Thần Nông
- 23/11 – 21/12: cung Nhân Mã
- 22/12 – 20/01: cung Ma Kết
- 21/01 – 19/02: cung Bảo Bình
- 20/02 – 20/03: cung Song Ngư

8. Bài 07 – Bài tập tổng hợp

Viết chương trình cho phép nhập vào một địa chỉ mạng, kèm theo subnet mask (ví dụ: 192.168.1.0/24), số mạng con cần chia. Thực hiện việc chia địa chỉ mạng đó thành các mạng con với các mô tả tương tự như bảng bên dưới.

STT	Địa chỉ mạng	Địa chỉ đầu	Địa chỉ cuối	Địa chỉ Broadcast
1				

Lưu ý:

- Có thể sử dụng animation hợp lý để thể hiện cách chia mạng con.
- Có kiểm tra điều kiện ràng buộc trước khi tính toán để chia mạng con.

9. Bài 08 – Xử lý mảng & Tổng hợp

Viết chương trình cho phép nhập vào danh sách điểm của sinh viên dưới dạng 1 mảng 1 chiều, mỗi phần tử điểm cách nhau dấu cách phẩy ",", đầu mảng là họ và tên sinh viên.

Ví dụ: Nguyễn Thị A, 7.5, 5, 8, 10, 9,10, 8.5, 9, 10, 3.5 ,5.5, 2

Yêu cầu:

- Kiểm tra tính hợp lệ và thông báo (Đã nhập đúng/sai format).
- Xuất ra tên sinh viên với định dạng:

Họ và tên: Nguyễn Thị A

- Xuất ra danh sách điểm kèm tiêu đề Môn với định dạng:

Môn 1: 7.5 Môn 2: 5 Môn 3: 8 Môn 4: 10

- Thực hiện phép tính tính **Điểm Trung bình**, hiện kết quả.
- Tìm môn điểm **cao nhất, thấp nhất** của sinh viên.
- Tìm **số môn đậu, không đậu**
- Xếp loại sinh viên dựa theo Điểm TB và các ràng buộc:
 - Giỏi: ĐTB ≥ 8 , không có môn nào < 6.5 ○ Yếu: ĐTB ≥ 3.5 , không có môn nào < 2
 - Khá: ĐTB ≥ 6.5 , không có môn nào < 5 ○ Kém: Còn lại
 - TB: ĐTB ≥ 5 , không có môn nào < 3.5

D. YÊU CẦU, NỘI BÀI

1. Yêu cầu

- Các giao diện ở trên chỉ mạng tính chất minh họa, sinh viên tiến hành thiết kế giao của riêng mình đảm bảo các tiêu chí: dễ nhìn, thể hiện hết được các yêu cầu cần thực hiện, đẹp.
- Có kiểm tra các điều kiện ràng buộc khi nhập dữ liệu.
- Có kiểm tra các điều kiện ràng buộc khi nhập dữ liệu, code "sạch" [2], đặt tên biến rõ ràng.
- Nộp bài không đầy đủ; lỗi, không chạy được; nộp trễ; sao chép code bạn khác, nguồn có sẵn: *xử lý tùy theo mức độ*.

2. Nộp bài

- Sinh viên thực hành và nộp bài **cá nhân** tại website môn học theo thời gian quy định.
- Source-code của các bài tập liên quan được thực hiện tại GitHub, báo cáo theo yêu cầu được thống nhất trong Bài thực hành 1.

Nếu có nhiều files, đặt vào 1 file nén (.zip) với tên theo quy tắc sau:

NhomY-LabX_MSSV1_MSSV2

Ví dụ: *Nhom1-Lab1_20520001_20520002*

E. THAM KHẢO

[1] Microsoft (2018). C# Guide. [Online] Available at: <https://docs.microsoft.com/en-us/dotnet/csharp/>

[2] Martin, R. C. (2009). *Clean code: a handbook of agile software craftsmanship*. Pearson Education.

HẾT