



1

Lab

Lập trình C# & Windows Forms cơ bản

Introduction to C# & Winform Application

Thực hành Lập trình mạng căn bản – NT106

Giảng viên biên soạn: ThS. Đỗ Thị Hương Lan

Tháng 03/2025

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... else
 - Cấu trúc điều khiển switch ... case
 - Cấu trúc điều khiển vòng lặp
 - Nắm được các kỹ thuật xử lý với string, mảng và các kiểu dữ liệu khác
- Lập trình ứng dụng có giao diện đồ họa GUI với C# và Windows Forms
 - Có thể lập trình Giao diện ứng dụng với:
 - Giao diện đồ họa (Kéo thả, thay đổi thuộc tính)
 - Lập trình mã với các file .designer.cs
 - Có thể lập trình chức năng, sự kiện cơ bản giải quyết các bài toán nhỏ

2. Môi trường

- IDE Microsoft Visual Studio 2010 – 2019
- Máy tính sử dụng hệ điều hành Windows/Linux/macOS

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

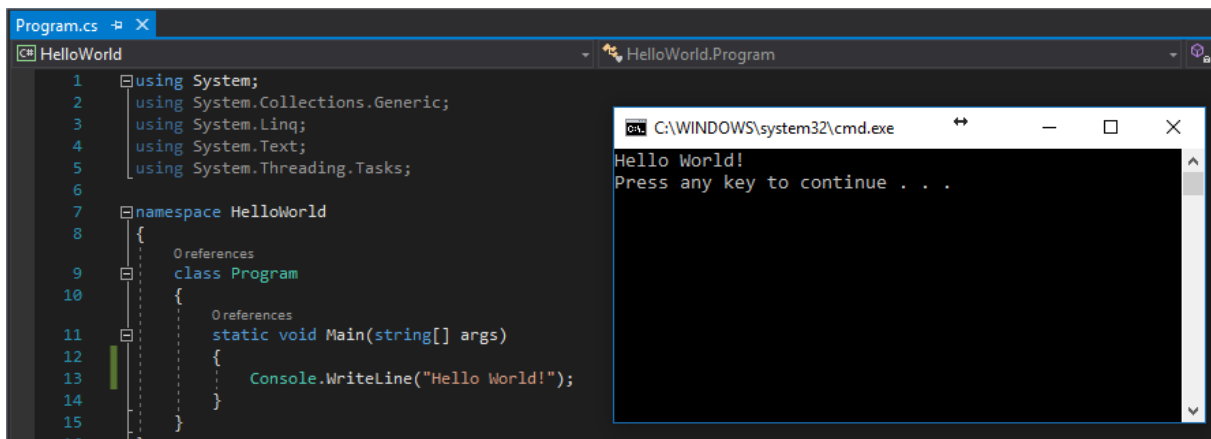
B. VÍ DỤ XÂY DỰNG ỨNG DỤNG C# CONSOLE VÀ C# WINFORM

1. Chương trình Hello World (Console App)

Yêu cầu: Viết chương trình console in ra màn hình dòng thông tin "Hello World!" và hiện dấu nhắc đợi thông tin nhập từ bàn phím

Hướng dẫn:➤ **Tạo Project**

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

➤ **Lập trình**

- Chọn File Program.cs và thêm mã nguồn vào hàm Main()
- Sử dụng Console.WriteLine("Hello World!") để in ra màn hình "Hello World"
- Sử dụng Console.ReadKey() để đợi nhập từ bàn phím

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

TÍNH TỔNG 2 SỐ NGUYÊN

Số thứ nhất: 23546

Số thứ hai: 79810

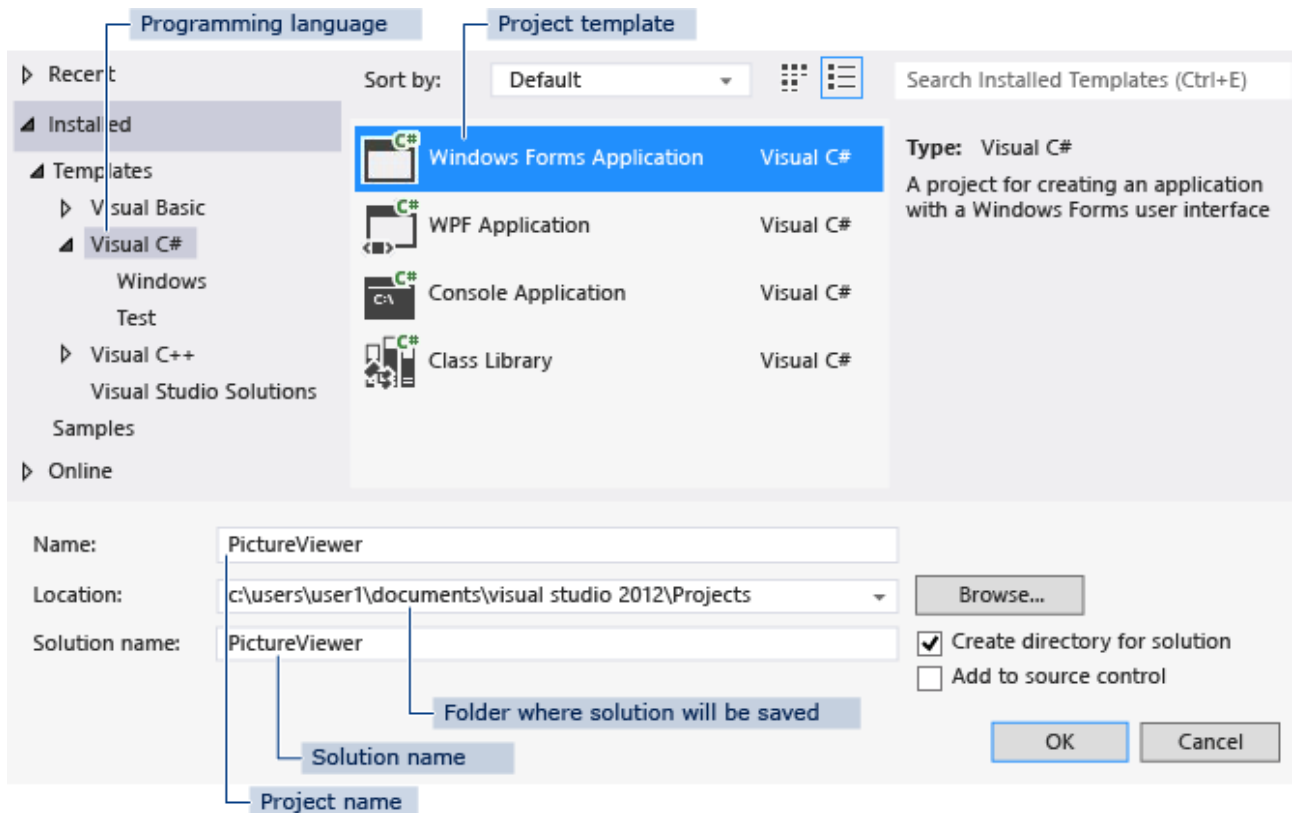
Tính

TỔNG 103356

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

Hướng dẫn:➤ **Tạo Project**

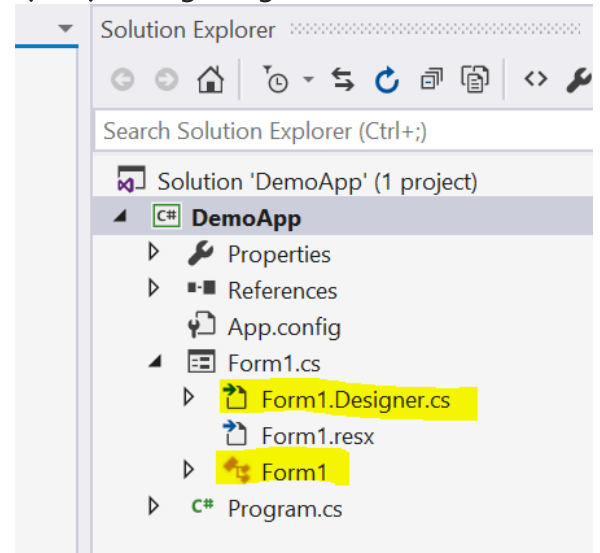
- Chọn *File > New > Project > Visual C# > Windows Forms Application* để tạo Project **Windows Forms App**. Đặt tên Project, Solution, chọn nơi lưu trữ và nhấn OK.

➤ **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**

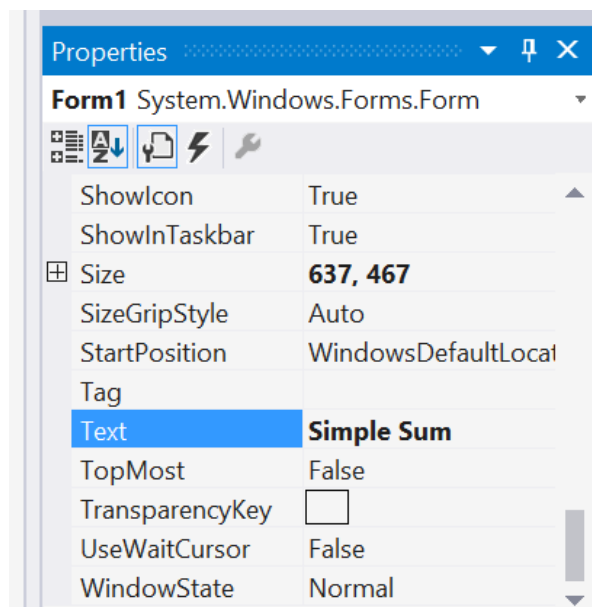
Lab 1: Lập trình C# & Windows Forms cơ bản

- **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.






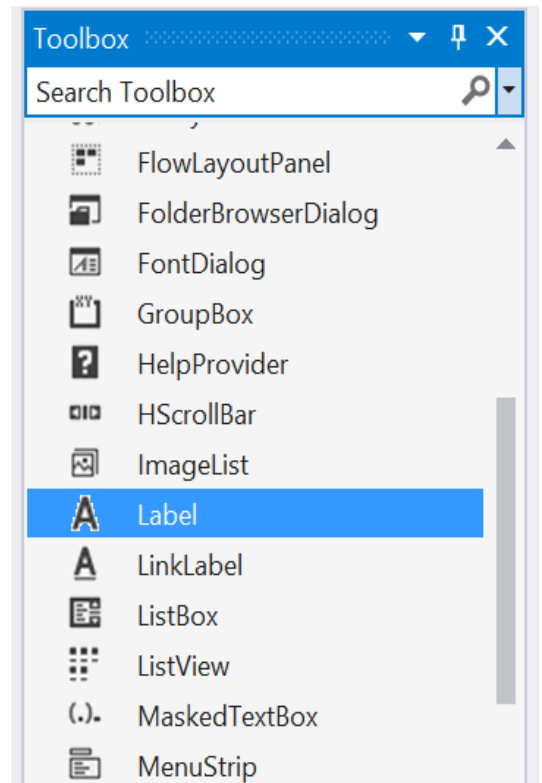
➤ Thiết kế giao diện (Design) theo yêu cầu của đề bài

- **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 ...



Lab 1: Lập trình C# & Windows Forms cơ bản

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

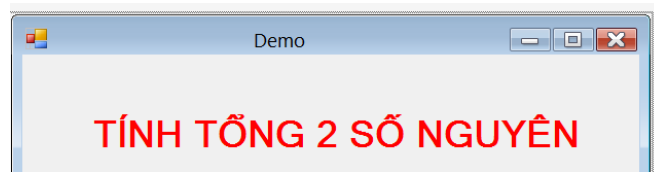
/// </summary>
private void InitializeComponent()
{
    this.label1 = new System.Windows.Forms.Label();
    this.label2 = new System.Windows.Forms.Label();
    this.label3 = new System.Windows.Forms.Label();
    this.textBox1 = new System.Windows.Forms.TextBox();
    this.textBox2 = new System.Windows.Forms.TextBox();
    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";
    ..

```

Lab 1: Lập trình C# & Windows Forms cơ bản

- 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();
}
```

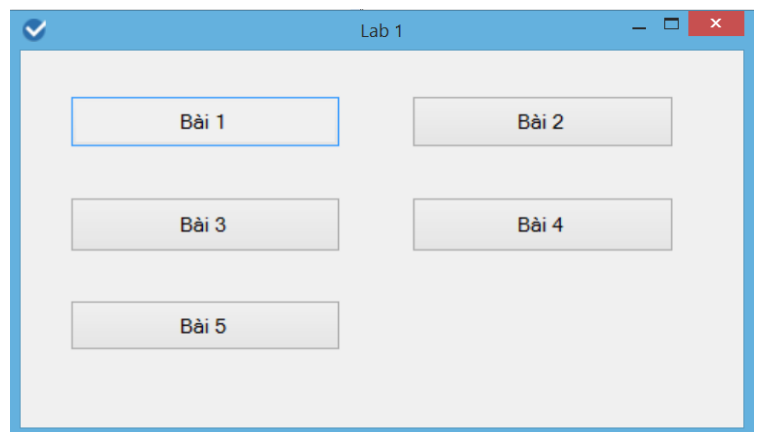
➤ Chạy ứng dụng: nhấn F5 hoặc chọn Debug để chạy chương trình

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 tập 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.

Khuyến nghị: Tất cả các bài thực hành đặt chung trong 1 Project duy nhất, mỗi bài tập là 1 Form đặt tên theo dạng **Lab1-BaiX** (với X là số thứ tự bài tập) và có Form điều hướng để mở các bài tập

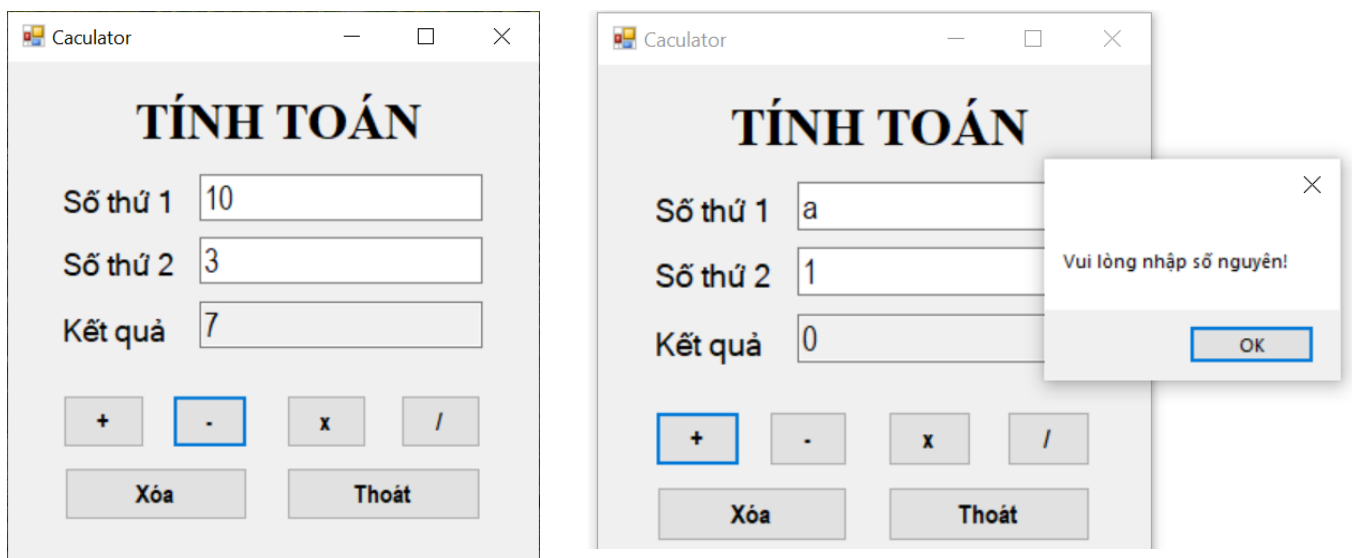


1. Bài 01 – Chương trình tính toán đơn giản (Có điều kiện)**Yêu cầu:**

Yêu cầu: Viết chương trình cho phép nhập 2 số nguyên và thực hiện các phép tính toán (Cộng, trừ, nhân, chia) với giao diện bên dưới

Yêu cầu chi tiết:

- Thiết kế giao diện phù hợp, dễ sử dụng
- Có tính năng kiểm tra tính hợp lệ của dữ liệu
- Nhấn các nút **+** **-** **x** **/**: cho biết giá trị phép toán tương ứng của 2 số đã nhập
- Nhấn nút **"Xóa"**: để làm rỗng/làm mới toàn bộ các ô nhập
- Nhấn nút **"Thoát"**: để thoát/tắt ứng dụng

Giao diện minh họa:**Gợi ý:**

- Tương tự Ví dụ **Tính tổng 2 số nguyên**
- Sử dụng **MessageBox** để hiển thị thông điệp cảnh báo
- Sử dụng phương thức **TryParse()** hoặc cấu trúc **try catch**

Tham khảo thêm:

- Microsoft, **Int32.TryParse Method**, (Online)
<https://docs.microsoft.com/en-us/dotnet/api/system.int32.parse?view=net-6.0>
- Microsoft, **try-catch (C# Reference)**, (Online)
<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/try-catch>

2. Bài 02 – Số lớn nhất, nhỏ nhất

Yêu cầu:

Viết chương trình cho phép nhập vào 3 số bất kỳ, tìm số lớn nhất và nhỏ nhất

Yêu cầu chi tiết:

- Thiết kế giao diện phù hợp, dễ sử dụng
- Có tính năng kiểm tra tính hợp lệ của dữ liệu
- Nhấn nút “**Tìm**”: cho biết số lớn nhất và số nhỏ nhất trong 3 số
- Nhấn nút “**Xóa**”: để làm rỗng/làm mới toàn bộ các ô nhập
- Nhấn nút “**Thoát**”: để thoát/tắt ứng dụng

Giao diện minh họa:

Min Max

TÌM SỐ NHỎ NHẤT, LỚN NHẤT

Các giá trị

Số thứ 1 Số thứ 2 Số thứ 3

Kết quả

Số bé nhất Số lớn nhất

Gợi ý:

- Chú ý **kiểu dữ liệu** cho phép nhập
- Sử dụng **MessageBox** để hiển thị thông điệp cảnh báo
- Sử dụng **TryParse()** hoặc cấu trúc **try catch** để kiểm tra tính hợp lệ của dữ liệu
- Sử dụng cấu trúc **If – else** hoặc các phương thức/hàm hỗ trợ sẵn của C#

3. Bài 03 – Đọc số

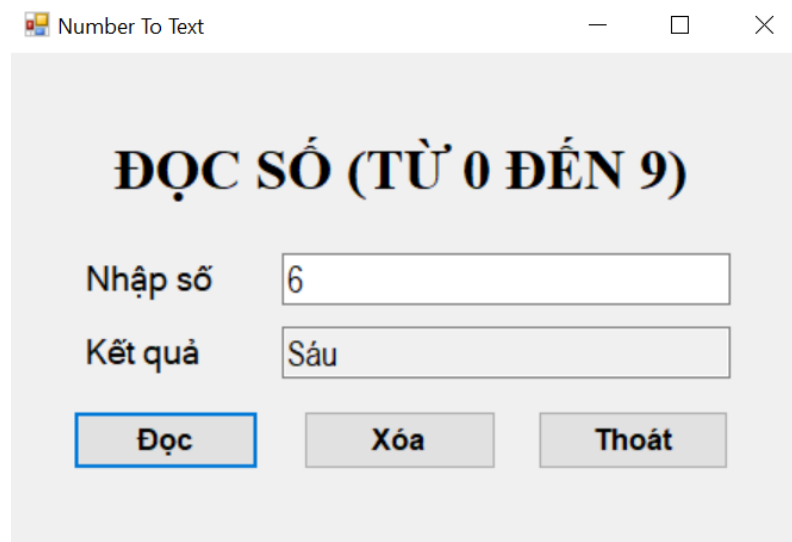
Yêu cầu:

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”

Yêu cầu chi tiết:

- Thiết kế giao diện phù hợp, dễ sử dụng
- Có tính năng kiểm tra tính hợp lệ của dữ liệu
- Nhấn nút “**Đọc**”: hiển thị cách đọc thành chữ của số đã nhập
- Nhấn nút “**Xóa**”: để làm rỗng/làm mới ô nhập
- Nhấn nút “**Thoát**”: để thoát/tắt ứng dụng

Giao diện minh họa:**Gợi ý:**

- Sử dụng **MessageBox** để hiển thị thông điệp cảnh báo
- Sử dụng **TryParse()** hoặc cấu trúc **try catch** để kiểm tra tính hợp lệ của dữ liệu
- Sử dụng cấu trúc **Switch – case**

Nâng cao:

- Nhập 101, Kết quả: Một trăm lẻ một
- Nhập 2090, Kết quả: Hai ngàn không trăm chín mươi

4. Bài 04 – Đổi hệ cơ số

Yêu cầu:

Viết chương trình cho phép nhập vào một số và chuyển đổi giữa các hệ cơ số (Nhị phân, Thập phân, Thập lục phân)

Yêu cầu chi tiết:

- Thiết kế giao diện phù hợp, dễ sử dụng
- Có tính năng kiểm tra tính hợp lệ của dữ liệu
- Nhấn nút **“Thực hiện”**: để thực hiện chuyển đổi giữa các hệ cơ số
- Nhấn nút **“Xóa”**: để làm rỗng/làm mới ô nhập
- Nhấn nút **“Thoát”**: để thoát/tắt ứng dụng

Giao diện minh họa:

Convert HEX, DEC, BIN

Nhập thông tin

Nhập một số: 52366333

Chọn: Decimal Sang Binary

Thực hiện

Kết quả: 11000111110000101111111101

Gợi ý:

- Sử dụng **ComboBox** để thiết kế danh sách chọn
- Sử dụng **MessageBox** để hiển thị thông điệp cảnh báo
- Sử dụng **TryParse()** hoặc cấu trúc **try catch** để kiểm tra tính hợp lệ của dữ liệu
- Sinh viên tự tìm hiểu các chuyển đổi giá trị từ hệ cơ số này sang hệ cơ số khác

5. Bài 05 – Xử lý mảng & Tổng hợp

Yêu cầu:

Thiết kế và viết chương trình quản lý điểm của sinh viên

Yêu cầu chi tiết:

Viết chương trình cho phép nhập vào danh sách điểm của sinh viên dưới dạng một mảng 1 chiều, mỗi phần tử điểm cách nhau bởi một dấu cách (khoảng trắng)

Ví dụ: 7.5 5 8 10 9 10 8.5 9 10 3.5 5.5 2

- Kiểm tra tính hợp lệ và thông báo (Đã nhập đúng/sai format)
- 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:
 - o Giỏi: ĐTB ≥ 8 , không có môn nào < 6.5
 - o Khá: ĐTB ≥ 6.5 , không có môn nào < 5
 - o TB: ĐTB ≥ 5 , không có môn nào < 3.5
 - o Yếu: ĐTB ≥ 3.5 , không có môn nào < 2
 - o Kém: Còn lại

Giao diện minh họa:

The screenshot shows a Windows application window titled "StuScore". The main title is "QUẢN LÝ ĐIỂM". Below the title, there is a text input field labeled "Dữ liệu điểm:" containing the text "6.5 7.5 7 8.5 10 6 8 9 8.5 7.2". Below the input field, there is a red italicized note: "* Lưu ý: Nhập với dạng danh sách. Ví dụ: 5 6.5 8 9 10 7". To the right of the input field is a blue button labeled "Xuất". Below the input field, there is a section titled "Danh sách môn học và điểm" containing a list of subjects and their scores: Môn 1: 6.5, Môn 2: 7.5, Môn 3: 7, Môn 4: 8.5, Môn 5: 10, Môn 6: 6, Môn 7: 8, Môn 8: 9, Môn 9: 8.5, Môn 10: 7.2. At the bottom of the window, there are six labels: "Điểm TB: 7.82", "Số môn đậu: 10", "Số môn không đậu: 0", "Điểm cao nhất: 10", "Điểm thấp nhất: 6", and "Xếp loại: Giỏi".

D. YÊU CẦU & ĐÁNH GIÁ

1. Yêu cầu

- Sinh viên thực hành và nộp bài **cá nhân** theo thời gian quy định.
- Bài nộp: **Source code** (nén) và 1 **File .pdf** báo cáo tóm lược các bài tập liên quan kèm hình ảnh minh họa

Toàn bộ project đặt vào 1 file nén (.rar/.zip)	File .PDF screenshot
LabX-MSSV-HọTênSV	LabX-MSSV-HọTênSV.pdf
Ví dụ: Lab1-16520901-NguyenVanA	Ví dụ: Lab1-16520901-NguyenVanA.pdf

2. Đánh giá kết quả

- Tiêu chí đánh giá:
 - Chương trình chạy được, hoàn thành các yêu cầu cơ bản: **+70%**.
 - Có kiểm tra các điều kiện ràng buộc khi nhập dữ liệu, code “sạch” [2], đặt tên biến rõ ràng: **+30%**.
 - 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* (- **(10 → 100)%**)

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