MÔN : LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
Bài thực hành số 10.1 : Xây dựng và debug class cụ thể để chuẩn bị chuyển thành class tổng quát hóa

I. Mục tiêu :

- Giúp SV làm quen với việc chuẩn bị xây dựng class tổng quát hóa.

II. Nội dung :

- Viết code miêu tả class "Stack các số nguyên" cung cấp 2 tác vụ push() và pop() để cất/lấy lại từng số nguyên trên đỉnh stack.
- Viết chương trình nhỏ thử dùng đối tượng "Stack các số nguyên", thử cất 10 số nguyên vào stack rồi lấy lại xem Stack có hoạt động tốt không.
- Debug phần mềm và class Stack để đảm bảo chúng chạy đúng chức năng.

III. Chuẩn đầu ra :

- Sinh viên nắm vững việc đặc tả class cụ thể, nhận thức đây là bước đầu tiên nên làm để xây dựng class tổng quát hóa dễ dàng, ít rủi ro.

IV. Qui trình :

1. Chạy VS .Net, chọn menu File.New.Project để hiển thị cửa sổ New Project.
2. Mở rộng mục Visual C# trong TreeView "Project Types", chọn mục Windows, chọn icon "Console Application" trong listbox "Templates" bên phải, thiết lập thư mục chứa Project trong listbox "Location", nhập tên Project vào textbox "Name:" (td. AnyStackpp), click button OK để tạo Project theo các thông số đã khai báo.
3. Ngay sau Project vừa được tạo ra, cửa sổ soạn code cho chương trình được hiển thị. Viết code cho hàm Main như sau :

```
namespace AnyStackApp {
   class Program  {
      static void Main(string[] args) {
         int i;
         //tạo đối tượng IntStack để dùng
         IntStack si = new IntStack();
         //push lần lượt 11 giá trị từ -5 tới 5
         for (i = -5; i <= 5; i++)  {
            if (!si.push(i)) {
               Console.WriteLine("Khong push duoc nua!!!");
               return;
            }
         }
         //pop các giá trị trong Stack ra và hiển thị để kiểm tra
         try {
            while (true) {
               int ci = si.pop();
               Console.WriteLine("Tri vua pop ra la : " + ci);
            }
         }  catch (Exception e) {
            Console.Write("Hết stack. Ấn Enter để đóng cửa sổ");
            Console.Read();
         }
      } //hết hàm Main
   } //hết class Program
} //hết namespace
```

4. Ấn phải chuột vào phần tử gốc của cây Project trong cửa sổ Solution Explorer, chọn option Add.Class, đặt tên là IntStack.cs để tạo ra file đặc tả class IntStack. Khi cửa sổ hiển thị mã nguồn của class IntStack hiển thị, đặc tả class IntStack như đoạn code dưới đây :

```
namespace AnyStackApp {
public class IntStack {
private int[] data; //danh sách các phần tử trong stack
private int top;      // chỉ số phần tử đỉnh stack
private int max;     // số lượng max hiện hành stack

// khai báo hằng miêu tả số lượng phần tử cần thêm mỗi lần thiếu stack
private int GROWBY = 4;

//hàm constrcutor
public IntStack() {
     top = 0;
     max =GROWBY;
     data = (int[])new int[max];
}

//hàm push phần tử vào đỉnh
public bool push(int newVal) {
int[] newdata;
  if (top==max) { //nếu đầy stack
       //xin cấp phát lại vùng nhớ lớn hơn GROWBY phần tử sơ với stack hiện hành
       try {
            newdata = (int[])new int[GROWBY+max];
        } catch (Exception e){
          //System.out.println("He thong het cho roi!!!");
          return false;
       }
       //di chuyển stack hiện hành về stack mới
       for (int i = 0; i<max; i++)
          newdata[i] =data[i];
       //cập nhật lại stack mới, để hệ thống xóa stack cũ tự động
       data = newdata;
       max += GROWBY;
   }
   //chứa giá trị mới vào đỉnh stack
   data[top++] =newVal;
   return true;
}

//hàm pop 1 phần tử từ đỉnh stack
public int pop() {
    if (top == 0) //nếu cạn stack  thì tạo Exception
       throw new Exception ("Cạn stack");
    else //trả về trị ở đỉnh stack
       return data[--top];
}
} //hết class IntStack
} //hết namespace AnyStackApp
```

5. Chọn menu Debug.Start Debugging để dịch và chạy ứng dụng. Hãy xem kết quả hiển thị và đánh giá chức năng của đối tượng IntStack.