



TRƯỜNG ĐẠI HỌC THỦY LỢI  
Khoa CNTT – Bộ môn CNPM

# LẬP TRÌNH NÂNG CAO



Giảng viên: Nguyễn Thị Phương Dung

Email: [dungntp@tlu.edu.vn](mailto:dungntp@tlu.edu.vn)

SĐT: [0946 079 903](tel:0946079903)



# XỬ LÝ NGOẠI LỆ



# Ngoại lệ là gì?

- Một ngoại lệ (Exception) là một vấn đề xuất hiện trong khi thực thi một chương trình.
- Ví dụ:
  - Lỗi chuyển đổi sai kiểu (chuyển ký tự thành số)
  - Lỗi chia cho số 0.
  - Lỗi truy cập tới phần tử của mảng với chỉ số không đúng
  - Lỗi truy cập vào đối tượng null
  - ....





# Xử lý ngoại lệ

- Xử lý ngoại lệ là cách đưa ra những thông báo khi chương trình gặp phải ngoại lệ trong quá trình chạy, giúp chương trình vẫn có thể chạy đến khi kết thúc một cách mượt mà theo chủ ý, không bị dừng đột ngột.





# Thực hiện bắt ngoại lệ

- Sử dụng khối lệnh try...catch()

- Cú pháp:

```
try
{
    //các lệnh có thể gây ra ngoại lệ
}
catch(/*khai báo biến bắt ngoại lệ*/)
{
    //các lệnh xử lý ngoại lệ
}
```





# Ví dụ

Lệnh có khả năng bị lỗi

```
int a;  
Console.Write("a = ");  
try  
{  
    a = Convert.ToInt32(Console.ReadLine());  
}  
catch (FormatException e)  
{  
    Console.WriteLine("Lỗi " + e.ToString());  
}
```

Lỗi bắt được từ lệnh trên





# Bắt nhiều lỗi cùng lúc

```
try
{
    //đoạn mã có khả năng bị lỗi
}
catch(/*loại lỗi 1*/)
{
    //đoạn mã xử lý
}
catch (/*loại lỗi 2*/)
{
    //đoạn mã xử lý
}
```





# Ví dụ

```
try
{
    Console.Write("a = ");
    a = Convert.ToInt32(Console.ReadLine());
    Console.Write("b = ");
    b = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("Ket qua: {0}/{1}={2}", a, b, a / b);
}
catch(FormatException e)
{
    Console.WriteLine("Lỗi nhập sai dữ liệu");
}
catch(DivideByZeroException e)
{
    Console.WriteLine("Lỗi chia cho 0 ");
}
```





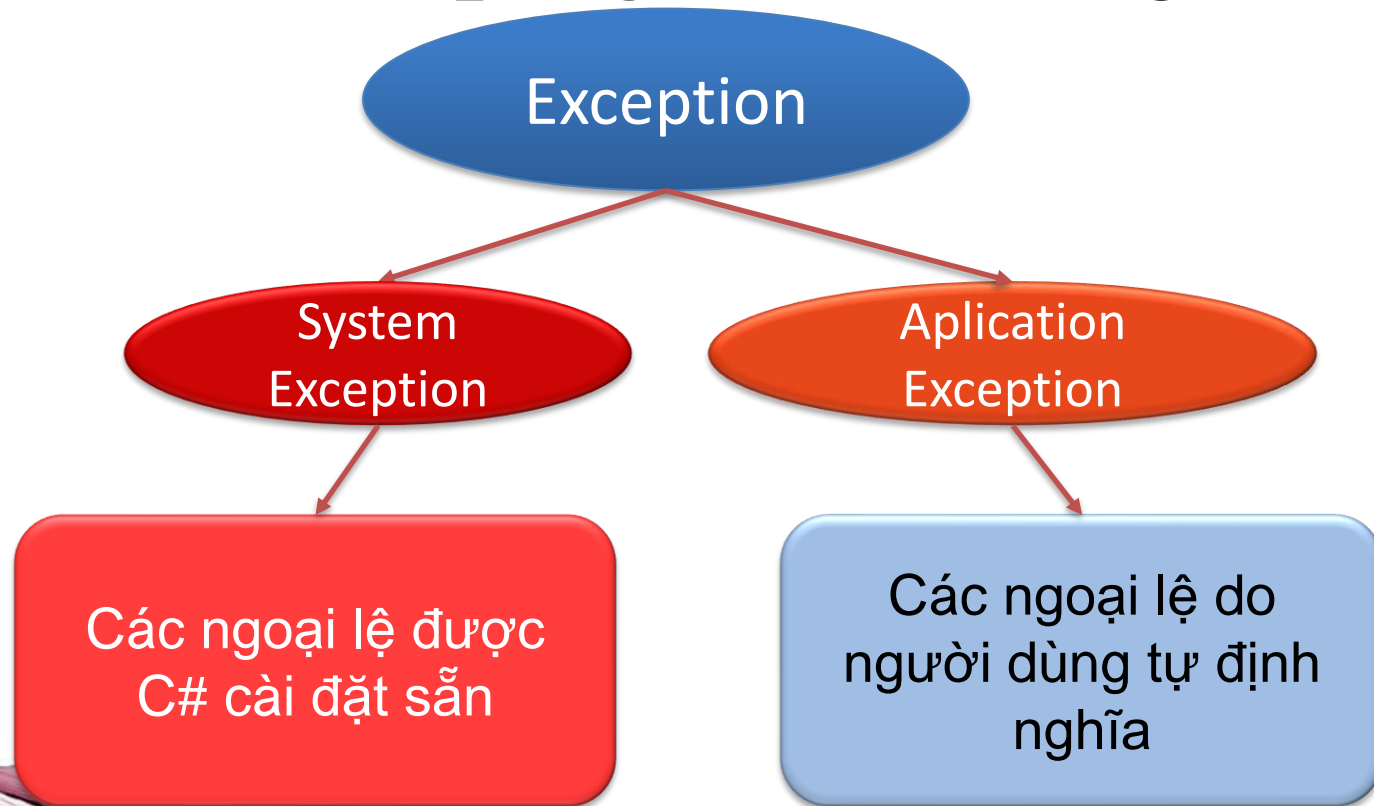


# Bắt nhiều lỗi cùng lúc

```
try
{
    //đoạn mã có khả năng bị lỗi
}
catch(*loại lỗi 1*/)
{
    //đoạn mã xử lý
}
catch (*loại lỗi 2*/)
{
    //đoạn mã xử lý
}
....
finally
{
    //đoạn mã thực hiện mặc định cho dù có lỗi hoặc không lỗi xảy ra
}
```



# Phân cấp ngoại lệ trong C#





# Một số ngoại lệ thông dụng trong C#

Kiểu ngoại lệ	Ý nghĩa
DivideByZeroException	Lỗi chia cho số 0
InvalidCastException	Xử lý lỗi được tạo trong khi ép kiểu
OutOfMemoryException	Xử lý lỗi được tạo từ việc thiếu bộ nhớ rồi
StackOverflowException	Xử lý lỗi được tạo từ việc tràn ngăn xếp (stack)
FormatException	Lỗi chuyển đổi định dạng





# Một số ngoại lệ thông dụng trong C#

Kiểu ngoại lệ	Ý nghĩa
Exception	Lớp ngoại lệ cơ bản
SystemException	Lớp cơ bản của mọi ngoại lệ phát ra tại thời điểm chạy của chương trình.
IndexOutOfRangeException	Được ném ra tại thời điểm chạy khi truy cập vào một phần tử của mảng với chỉ số không đúng.
NullReferenceException	Ném ra tại thời điểm chạy khi một đối tượng null được tham chiếu.
AccessViolationException	Ném ra tại thời điểm chạy khi tham chiếu vào vùng bộ nhớ không hợp lệ.
InvalidOperationException	Ném ra bởi phương thức khi ở trạng thái không hợp lệ.



# Một số ngoại lệ thông dụng trong C#

Kiểu ngoại lệ	Ý nghĩa
ArgumentNullException	Lớp này là con của ArgumentException, nó được ném ra bởi phương thức mà không cho phép thông số null truyền vào.
ArgumentOutOfRangeException	Lớp này là con của ArgumentException, nó được ném ra bởi phương thức khi một đối số không thuộc phạm vi cho phép truyền vào nó.
ExternalException	Lớp cơ bản cho các ngoại lệ xảy ra hoặc đến từ môi trường bên ngoài.
ArgumentException	Lớp cơ bản cho các ngoại lệ liên quan tới đối số (Argument).



# Ví dụ

```
static float phepchia(int a, int b)
{
    float ketqua = 0;
    try
    {
        ketqua = a / b;
    }
    catch(DivideByZeroException)
    {
        Console.WriteLine("Mau so khong hop le");
    }
    return ketqua;
}
```





# Bài tập

- Thiết kế một giao diện cho phép nhập 2 số  $a$  và  $b$ . Thực hiện đưa ra kết quả của các phép tính:
  - $a + b$
  - $a - b$
  - $a * b$
  - $a / b$
- Yêu cầu:
  - Đưa ra thông báo lỗi khi nhập sai dữ liệu
  - Đưa ra thông báo lỗi khi chia cho số 0





# Bài tập 2

- Viết chương trình cho phép nhập vào một số có 7 chữ số
- Thông báo lỗi nếu nhập không đúng







# Ngoại lệ tự định nghĩa

- Kế thừa lớp **ApplicationException**
- Cú pháp khai báo lớp ngoại lệ:

```
class <tên lớp>: ApplicationException
```

```
{
```

```
    public <tên lớp> (string msg) : base(msg){ }
```

```
}
```





# Ngoại lệ tự định nghĩa

- Cú pháp gọi lớp ngoại lệ thực hiện:  
**throw new** <tên lớp ngoại lệ>(<lời thông báo lỗi>);
- Sử dụng: giống như khi sử dụng các lớp ngoại lệ cơ bản





# Ví dụ ngoại lệ tự định nghĩa

```
class myexception:ApplicationException
{
    public myexception (string s):base(s) { }
```





# Ví dụ ngoại lệ tự định nghĩa

```
class canbo
{
    int tuoi;
    public void nhap()
    {
        Console.Write("Nhap tuoi: ");
        try
        {
            tuoi = Convert.ToInt32(Console.ReadLine());
        }
        catch (FormatException ex)
        {
            Console.WriteLine(ex.ToString());
        }
        if (tuoi < 10 || tuoi > 18)
            throw new myexception("Tuoi phai nam trong khoang tu 10 den 18");
    }
}
```



# Ví dụ ngoại lệ tự định nghĩa

```
class Program
{
    static void Main(string[] args)
    {
        canbo cb = new canbo();
        try
        {
            cb.nhap();
        }
        catch (myexception ex)
        {
            Console.WriteLine(ex.ToString());
        }
    }
}
```





# Bài tập

- Định nghĩa lớp ngoại lệ của riêng mình
- Thực hiện tạo một lớp tam giác có:
  - Dữ liệu: 3 cạnh
  - Phương thức: nhập, tính chu vi
  - Yêu cầu đưa ra thông báo lỗi khi nhập sai dữ liệu về cạnh tam giác
- Viết chương trình thực hiện:
  - Khai báo một mảng gồm 3 tam giác.
  - Nhập dữ liệu cho 3 tam giác đó
  - Đưa ra các thông báo lỗi có thể gặp khi chạy chương trình





# Bài tập

- Định nghĩa lớp ngoại lệ của riêng mình để thực hiện bắt các ngoại lệ sau:
  - Lỗi nhập sai dữ liệu
  - Lỗi truy cập ngoài chỉ số mảng
  - Kiểm tra tuổi tuyển dụng lao động
    - Nếu tuổi  $< 18$  thì thông báo là quá trẻ
    - Nếu tuổi  $> 40$  thì thông báo là quá già
    - Nếu tuổi từ 18 đến 40 thì thông báo là đạt yêu cầu

