**Lab 11**

**Kế thừa**

# Lập trình hướng đối tượng

|  |  |
| --- | --- |
| **Mục tiêu** | 1. Biết tạo ra kế thừa cơ bản 2. Hiểu được thứ tự hàm tạo và hàm hủy trong kế thừa |

**Bộ môn Công nghệ phần mềm – Khoa Công nghệ thông tin**



# Hướng dẫn khởi đầu 1 – Kế thừa

**Mô tả bài tập**

Cài đặt lớp **MixedFraction** để hỗ trợ hỗn số. Sau đó cài đặt hàm **ToString** để biểu diễn một hỗn số, ví dụ với phân số “12/9” sẽ hiển thị “1 1/3”

**Hướng dẫn cài đặt**

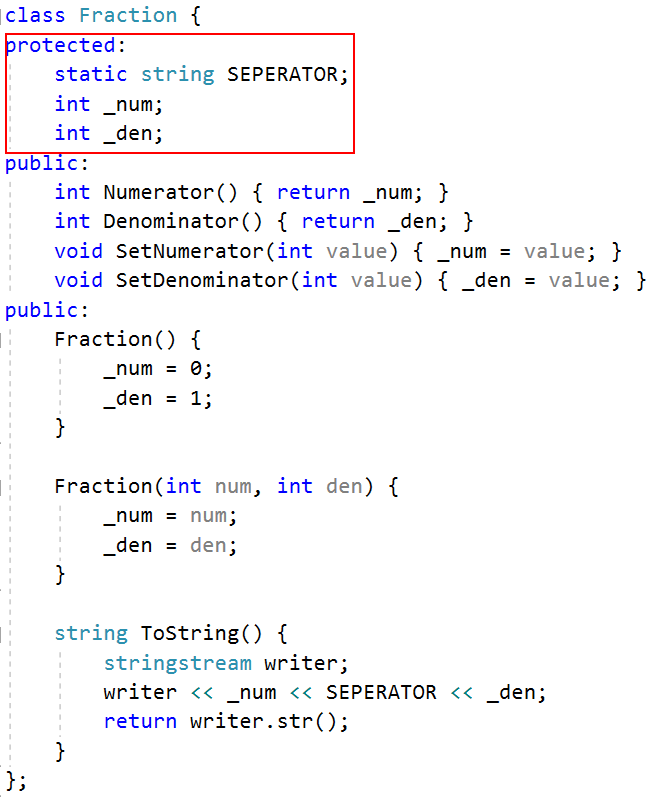
**Bước 1: Tạo mới dự án**

- Chọn loại dự án là **C++** / **Console Application**.

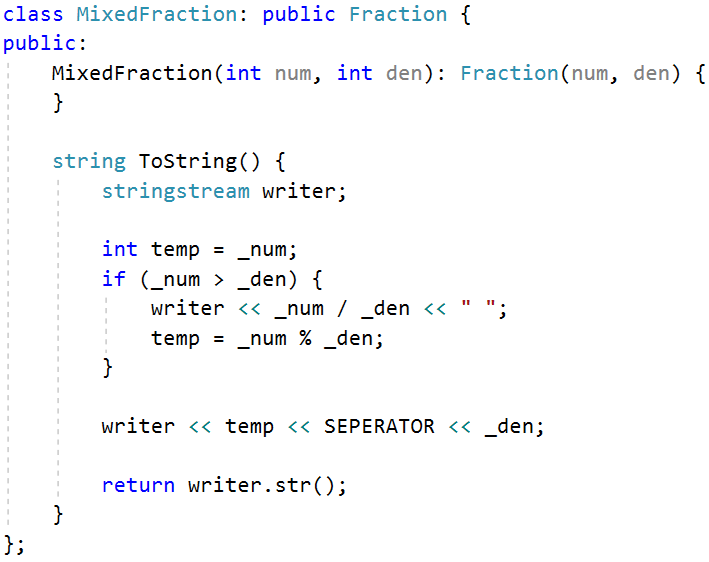
- Đặt tên solution là: **Inheritance**. Đặt tên project là **MixedFraction**

- Nếu sử dụng Visual Studio 2017 trở lên cần vô hiệu hóa **Precompiled header** bằng cách nhấn phải vào project chọn Properties. Vào mục **C / C++** > **All Options**, tìm tới tùy chọn **Precompiled header** và chọn **Not using precompiled headers**.

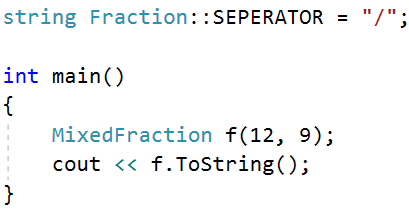
**Bước 2: Sử dụng lớp phân số có sẵn như sau:**

****

**Bước 3: Cài đặt lớp MixedFraction hỗ trợ hỗn số như sau**

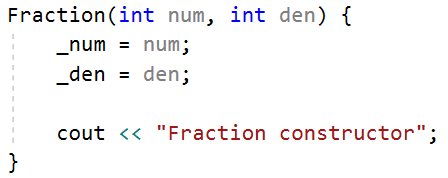


**Bước 4: Cài đặt hàm main để kiểm thử cài đặt**

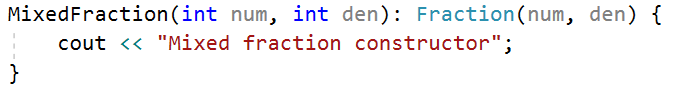


# Hướng dẫn khởi đầu 2 – Thứ tự hàm tạo

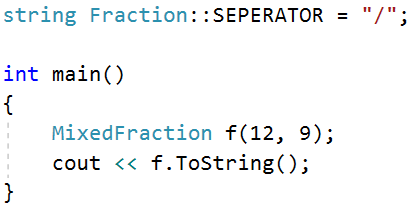
**Bước 1: Hiệu chỉnh hàm tạo lớp Fraction**



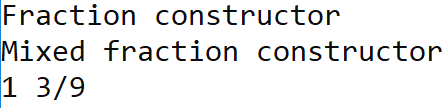
**Bước 2: Hiệu chỉnh hàm tạo lớp MixedFraction**



**Bước 3: Chạy lại hàm main để thấy thứ tự gọi hàm tạo**

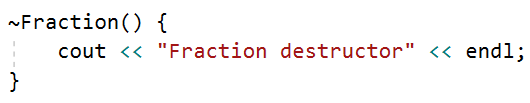
****

**Kết quả như sau:**

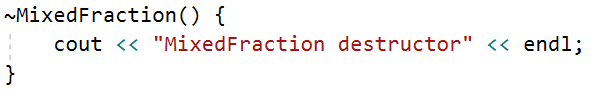


# Hướng dẫn khởi đầu 3 – Thứ tự hàm hủy

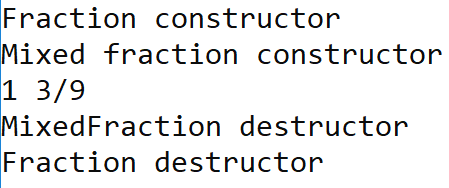
**Bước 1: Hiệu chỉnh bổ sung thêm hàm Hủy bên trong lớp Fraction**



**Bước 2: Hiệu chỉnh bổ sung thêm hàm Hủy bên trong lớp MixedFraction**



**Bước 3: Chạy lại hàm main để thấy thứ tự gọi hàm hủy**



**Câu hỏi**: Bạn có thể rút ra qui luật của hàm tạo trong kế thừa và hàm hủy trong kế thừa hay không?

**Gợi ý**: Giữa cha và con, ai được tạo trước? Ai được hủy trước?

# Bài tập vận dụng

**Yêu cầu**

1. Thực hiện định nghĩa lớp theo thiết kế cho trước vào tập tin .h.
2. Thực hiện cài đặt lớp trong tập tin .cpp cho lớp tương ứng.
3. Viết các đoạn mã nguồn kiểm tra việc định nghĩa lớp trong hàm main.

**Danh sách các lớp cần cài đặt**

1. Giả sử ta có lớp **Xe hơi** (Car) với các khả năng như sau:

+ **StartEngine**(): Khởi động động cơ

Để đơn giản hàm này sẽ in ra một dòng “*Starting engine*”

Cài đặt lớp Xe hơi mui trần (**Carbriolet**) khi khởi động động cơ thì sẽ bỏ đi phần mui.

Cài đặt bằng cách nạp đè hàm StartEngine và xuất ra dòng chữ “*Removing roof*” trước rồi xuất ra chữ “*Starting engine*” sau.

2. Giả sử có lớp **Thời gian** (**Time**) với 3 thành phần **giờ** (số nguyên), **phút** (số nguyên), **giây** (số nguyên).

+ Hàm tạo không đối sẽ tạo ra đối tượng với thời gian hiện tại.

+ Hàm tạo có đối sẽ khởi tạo với giờ phút giây truyền vào

Cài đặt lớp Thời gian theo GMT (**GmtTime**) kế thừa lớp **Time** có các đặc tính sau:

+ Thêm vào hàm **ToString**(int gmt): đối số gmt sẽ cho biết độ lệch thời gian theo giờ.

Ví dụ ta có đối tượng thời gian **GmtTime** now() ứng với 10h 15m 0s.

Nếu gọi now.ToString(0) sẽ in ra với độ lệch là 0, tức 10:15:00

Nếu gọi now.ToString(7) sẽ in ra với độ lệch là +7, tức 17:15:00

Chú ý khi vượt quá 24h! Giờ chỉ có giá trị từ 0h đến 23h59.

3. Một nông trại chăn nuôi có 3 loại gia súc: **bò**, **cừu**, và **dê**. Mỗi loại gia súc đều có thể **sinh con**, **cho sữa** và **phát ra tiếng kêu riêng** của chúng. Khi đói, các gia súc sẽ phát ra tiếng kêu để đòi ăn. Sau một thời gian chăn nuôi, người chủ nông trại muốn thống kê xem trong nông trại có bao nhiêu gia súc ở mỗi loại, tổng số lit sữa mà tất cả các gia súc của ông đã cho.

Áp dụng kế thừa, xây dựng chương trình cho phép người chủ nông trại tạo ra số lượng gia súc ngẫu nhiên lúc ban đầu.

a. Một hôm người chủ nông trại đi vắng, tất cả gia súc trong nông trại đều đói. Hãy cho biết những tiếng kêu nghe được trong nông trại.

Gợi ý: Duyệt qua mảng các gia súc, lần lượt gọi hàm Kêu của từng con vật.

Nếu là bò sẽ kêu kiểu bò, cừu kêu kiểu cừu, dê kêu kiểu dê. (Đừng quá quan trọng các con này kêu cụ thể ra sao nhé 😊, tượng trưng để hiểu bài là được)

b. Chương trình sẽ đưa ra thống kê các thông tin người chủ mong muốn (nêu trên) sau một lứa sinh và một lượt cho sữa của tất cả gia súc. Biết rằng:

- Tất cả gia súc ở mỗi loại đều sinh con (chỉ có giống cái)

- Số lượng sinh của mỗi gia súc là ngẫu nhiên.

- Tất cả gia súc ở mỗi loại đều cho sữa.

- Số lit sữa mỗi gia súc cho là ngẫu nhiên nhưng trong giới hạn sau:

+ Bò: 0 – 20 lit.

+ Cừu: 0 – 5 lit.

+ Dê: 0 – 10 lit.

# Bài tập vận dụng khó siêu cấp vô địch

1. Cho hai lớp Hình vuông (**Square**) và Hình chữ nhật (**Rectangle**). Ta biết hình vuông là một trường hợp đặc biệt của hình chữ nhật. Lớp nào kế thừa lớp nào?

Thử cài đặt hai lớp này.

2. Cho hai lớp Hình tròn (**Circle**) và Hình Ê líp (**Ellipse**). Ta biết hình tròn là một trường hợp đặc biệt của hình ê líp. Lớp nào kế thừa lớp nào?

Thử cài đặt hai lớp này.

**Chú ý:** Kế thừa cơ học (chỉ vì muốn tái sử dụng thành phần, mã nguồn) mà không quan tâm ngữ nghĩa thì không phải là kế thừa đúng.

Gợi ý: Chỉ nên đọc link bên dưới sau khi đã thử suy nghĩ cài đặt hai bài tập trên.

<https://en.wikipedia.org/wiki/Circle-ellipse_problem>

# Hướng dẫn nộp bài

**Trước khi nộp cần chú ý:**

- Lấy tập tin exe được biên dịch sẵn trong thư mục Debug, copy nó ra thư mục Release bên ngoài mã nguồn.

- Xóa hết tất cả các tập tin trung gian trong quá trình biên dịch bằng cách chọn **Build** > **Clean**.

- Chú ý thư mục ẩn **.vs** rất nặng. Cần hiển thị file ẩn mới thấy và xóa nó đi được.

Nếu bạn muốn biết cách làm đúng thì cần tự tìm cách build ở chế độ Release và copy file exe kết quả ra bên ngoài để nộp mới đúng. Tuy nhiên nếu chưa hiểu ý nghĩa thì cứ lấy đại file exe có sẵn đi nộp cũng được (hiện tại đang trong thư mục Debug ứng với chế độ biên dịch Debug)

**Tổ chức bài nộp**

+ Thư mục **Source**: chứa mã nguồn đã được clean

+ Thư mục **Release**: chứa tập tin thực thi đã được biên dịch từ mã nguồn

+ Tập tin **readme.txt**: chứa thông tin sinh viên, gồm MSSV và họ tên. Ghi chú kèm các thông tin giáo viên cần chú ý khi chấm bài.

Để nộp bài, nén tất cả lại và đặt tên với định dạng **MSSV.zip** hoặc **MSSV.rar** và nộp.

**Nếu làm đúng các bước trên file này sẽ có kích thước < 100 KB!**

(Tuy nhiên cũng đừng quá lo lắng nếu nó khác con số trên, miễn < 12 MB để nộp được trên moodle là okie nhé!)

**Ngoài lề**: Để đảm bảo sau này nhìn vào file nén còn biết ngay nó làm gì, ta nên thêm vào một số thông tin theo sau MSSV. Ví dụ: 0712221-Lab11-Inheritance.zip. Tuy nhiên việc này là KHÔNG bắt buộc nhé.

**-- HẾT --**