**Lab 07**

**Vector & Ngoại lệ**

# Lập trình hướng đối tượng

|  |  |
| --- | --- |
| **Mục tiêu** | 1. Làm quen với kiểu dữ liệu vector của STL 2. Luyện tập xử lí ngoại lệ |

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



# Hướng dẫn khởi đầu

**Mô tả bài tập**

- Tạo ra một vector chứa danh sách các số nguyên

- Thêm một vài số nguyên vào vector này

- Duyệt qua vector và xuất danh sách này ra màn hình

- Bắt ngoại lệ nếu truy cập ngoài chỉ mục của mảng

**Hướng dẫn cài đặt**

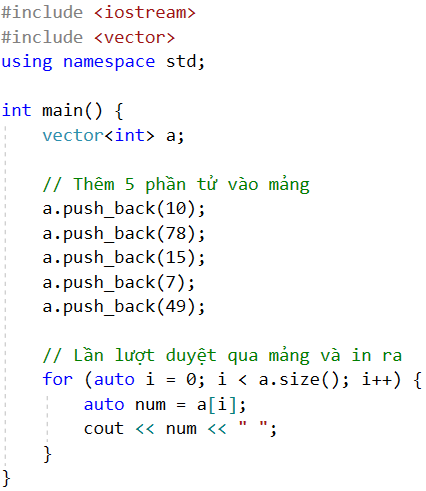
**Bước 1: Tạo mới dự án**

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

- Đặt tên solution là: **VectorException**. Đặt tên project là **Vector**

- 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**.

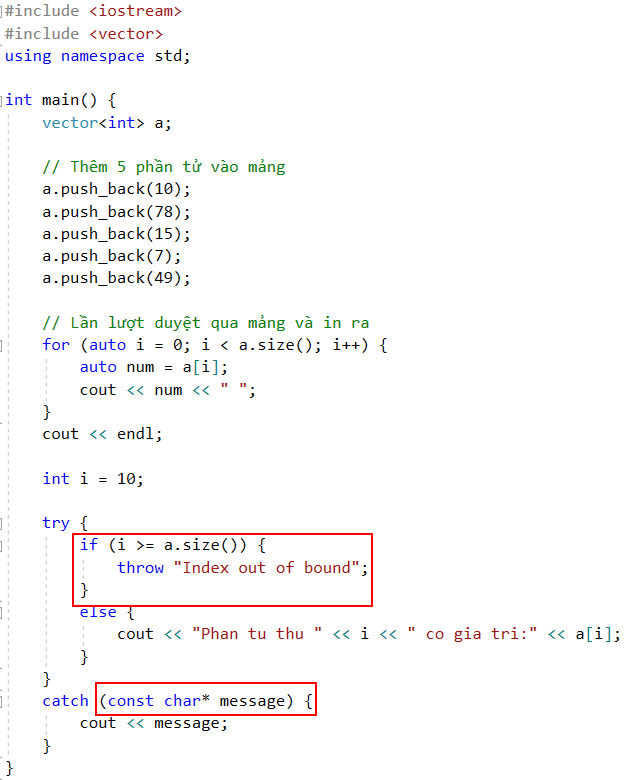
**Bước 2: Tạo mã nguồn trong hàm main như sau:**



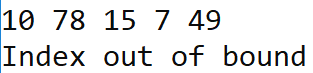
Biên dịch và chạy để thấy kết quả sau:



**Bước 3: Thay đổi mã nguồn để bắt exception**



**Chạy lên để thấy kết quả như sau:**



# 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 trong hàm main như mô tả.

**Danh sách các lớp cần cải tiến cụ thể**

1. Sử dụng lớp **Random**, tự phát sinh ra một số nguyên **n**. Sau đó phát sinh ra ngẫu nhiên **n** số nguyên và đưa vào vector. Xuất ra màn hình danh sách các số ngẫu nhiên vừa sinh ra.

Phát sinh ngẫu nhiên một số nguyên m. In ra số nguyên tại vị trí **m**. Sử dụng cơ chế bắt exception để đưa ra thông báo khi m vượt quá phạm vi của mảng.

2. Tương tự như trên, phát sinh ra ngẫu nhiên **n** **phân số** và đưa vào vector.

Sinh ngẫu nhiên một số nguyên **m**, in ra phân số tại vị trí **m** và phân số nghịch đảo tương ứng.

Nếu phân số này chưa tối giản, hãy in ra thêm phân số tối giản của nó.

Sử dụng cơ chế bắt exception để đưa ra thông báo khi m vượt quá giới hạn phạm vi của mảng.

3. Hoàn thiện lớp mảng động **DynamicArray**, bổ sung:

+ Khi thêm một phần tử vào mảng, nếu quá kích thước tối đa thì tăng lên với số lượng thêm là INITIAL\_SIZE, sau đó copy toàn bộ dữ liệu cũ vào mảng mới và xóa đi vùng nhớ cũ.

+ Khi lấy một phần tử tại vị trí i không hợp lệ **GetAt**(int) thì tạo ra ngoại lệ “Index out of bound”.

# 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-Lab07-VectorException.zip. Tuy nhiên việc này là KHÔNG bắt buộc nhé.

**-- HẾT --**