ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

BÀI TẬP REVIEW FINAL

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Sinh viên: Đồng Quốc Thắng - 23521421

Sinh viên: Bùi Ngọc Thiên Thanh - 23521436

Sinh viên: Cao Lê Công Thành - 23521437

Giáo viên hướng dẫn: Nguyễn Ngọc Quí

Ngày 4 tháng 12 năm 2024

Mục lục

1 Bài 1 - Đề 9

a. Phân biệt các phạm vi truy cập private, protected và public.

1. Private:
   - **Định nghĩa:** Chỉ có thể được truy cập và sử dụng trong chính lớp mà nó được khai báo. Không có đối tượng nào bên ngoài lớp có thể truy cập được các thành viên này.
   - **Mục đích:** Bảo vệ dữ liệu khỏi việc bị thay đổi hoặc truy cập từ bên ngoài lớp, đảm bảo tính đóng gói (encapsulation).
2. Protected:
   - **Định nghĩa:** Chỉ có thể được truy cập trong lớp mà chúng được khai báo và trong các lớp kế thừa (subclass). Tuy nhiên, chúng không thể được truy cập trực tiếp từ các đối tượng bên ngoài lớp.
   - **Mục đích:** Cung cấp sự linh hoạt khi các lớp con cần truy cập các thành viên của lớp cha, đồng thời vẫn giữ sự bảo vệ khỏi sự truy cập không mong muốn từ bên ngoài.
3. Public:
   - **Định nghĩa:** Chỉ có thể được truy cập từ bất kỳ đâu trong chương trình, bất kể lớp, đối tượng, hay package.
   - **Mục đích:** Cung cấp sự truy cập toàn cầu cho các thành viên của lớp. Đây là phạm vi truy cập rộng nhất và có thể gây ra các vấn đề về bảo mật nếu không được sử dụng cẩn thận.

b. Cho biết ý nghĩa và mục đích của các hàm get/set trong một lớp.

Ý nghĩa của hàm Get/Set: Hàm `get` và `set` thường được sử dụng để truy cập và thay đổi giá trị của các thuộc tính (biến) trong một lớp. Các hàm này được gọi là *getter* và *setter*.

1. **Hàm Get**
   - **Định nghĩa:** Hàm `get` (hay `getter`) được sử dụng để truy cập giá trị của một thuộc tính riêng tư (private) trong lớp.

- **Mục đích:** Cho phép truy cập giá trị của thuộc tính mà không cần truy cập trực tiếp vào biến đó từ bên ngoài lớp, giúp bảo vệ tính đóng gói (encapsulation).

2. **Hàm Set**

- **Định nghĩa:** Hàm `set` (hay `setter`) được sử dụng để thay đổi giá trị của một thuộc tính riêng tư.
- **Mục đích:** Cho phép thay đổi giá trị của thuộc tính, nhưng có thể bao gồm các kiểm tra hoặc logic để đảm bảo rằng giá trị được gán hợp lệ. Điều này giúp bảo vệ các thuộc tính khỏi giá trị không hợp lệ hoặc không mong muốn.

Mục đích chung:

- **Get:** Để đọc giá trị thuộc tính.
- **Set:** Để thay đổi giá trị thuộc tính, có thể kèm theo kiểm tra tính hợp lệ.

2 Bài 2 - Đề 9

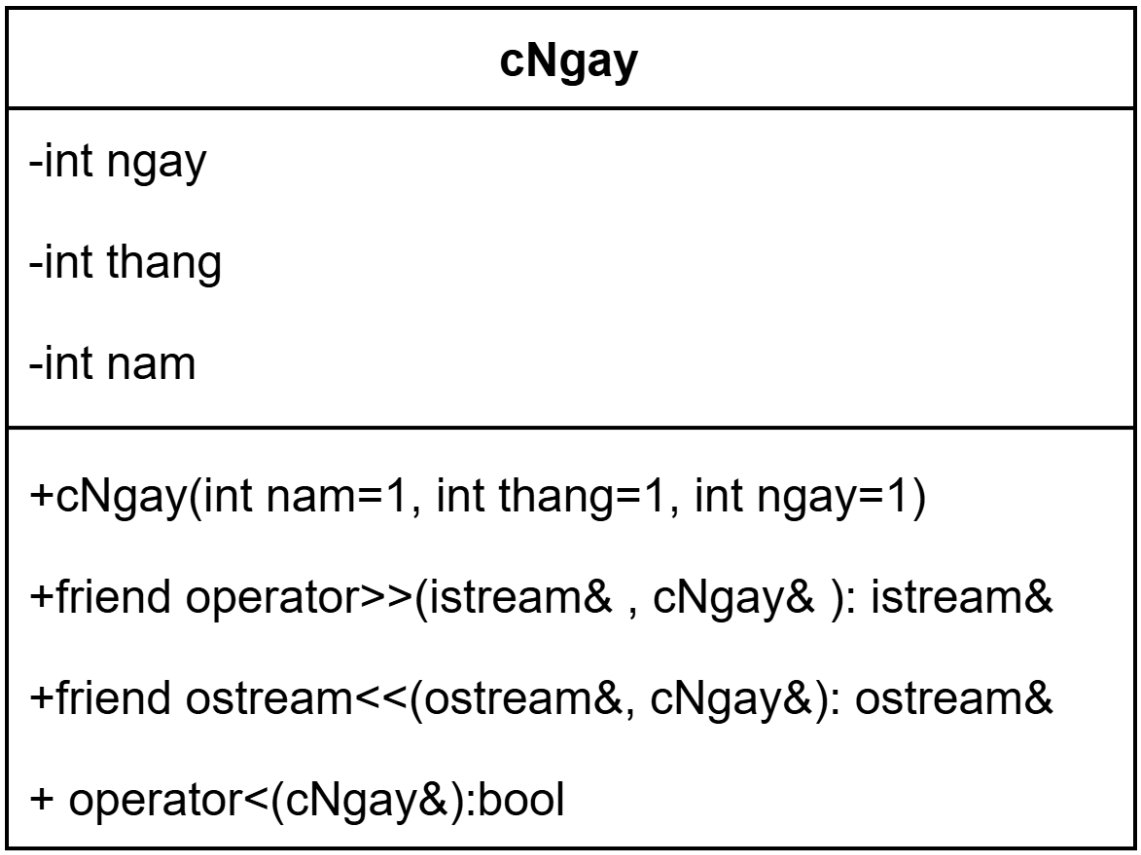
2.1 Đề Bài:

Cho đoạn chương trình tính toán với lớp đối tượng ngày tháng năm (`cNgay`) như sau:

```
void main()
{
    cNgay ng1;                // ng1 sẽ có giá trị là ngày 1 tháng 1 năm 1
    cNgay ng2(2017, 1);       // ng2 sẽ có giá trị là ngày 1 tháng 1 năm 2017
    cNgay ng3(2017, 1, 7);    // ng3 sẽ có giá trị là ngày 7 tháng 1 năm 2017
    cin >> ng1;
    cout << ng1;
    if(ng1 < ng2)
        cout << "Ngay 1 truoc ngay 2" << endl;
    else
        cout << "Ngay 1 khong truoc ngay 2" << endl;
}
```

Hãy định nghĩa lớp `cNgay` thích hợp để chương trình không bị lỗi biên dịch và chạy đúng. Lưu ý rằng không được chỉnh sửa hàm `main` và sinh viên cần viết cả các lệnh `#include` thích hợp.

2.2 Class Diagram:



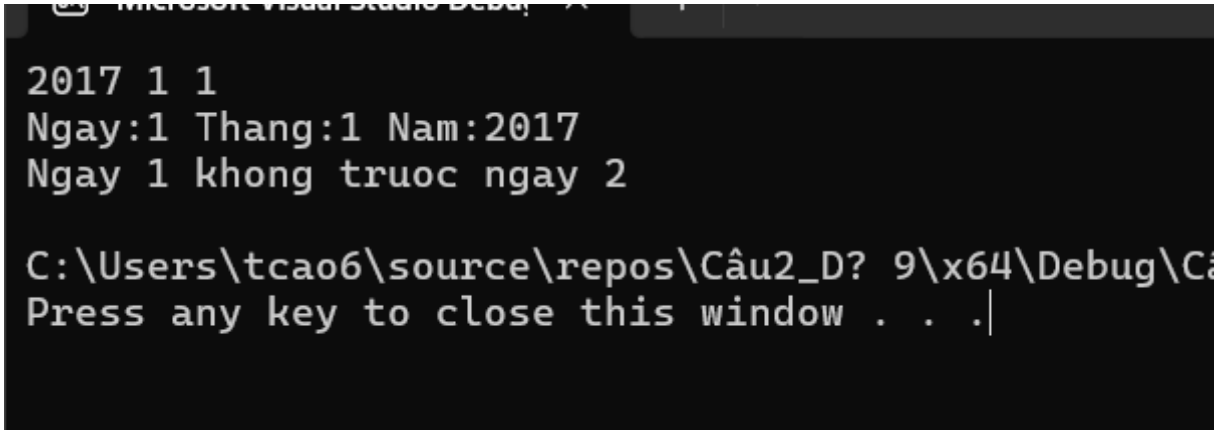
2.3 Mô tả và ý tưởng:

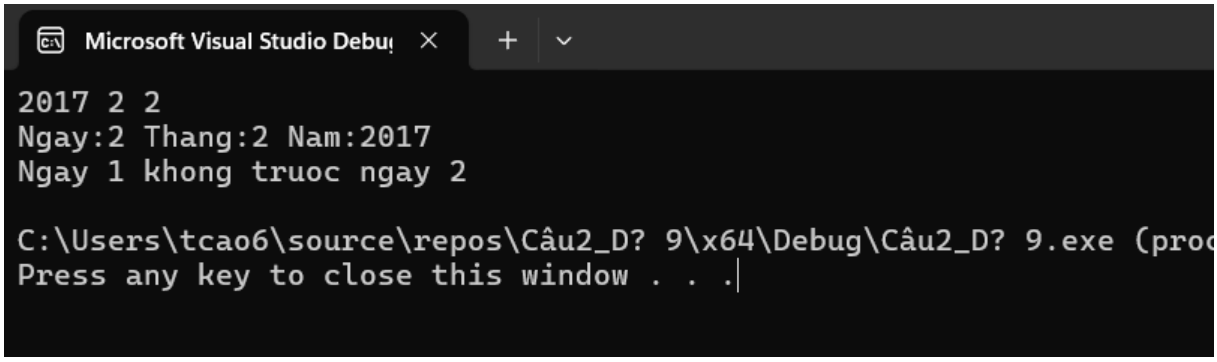
Input: Nhập vào thông tin của một ngày theo định dạng năm_tháng_ngày.
Output: Xuất ra thông tin ngày vừa nhập và kiểm tra ngày đó có trước ngày 2 trong đề bài hay không.
Solution:Nạp chồng các toán tử nhập ,xuất và so sánh cho lớp cNgay. Đối với phương thức so sánh mình sẽ so sánh năm trước, nếu năm bằng nhau thì so sánh tháng, nếu tháng bằng nhau thì so sánh ngày.

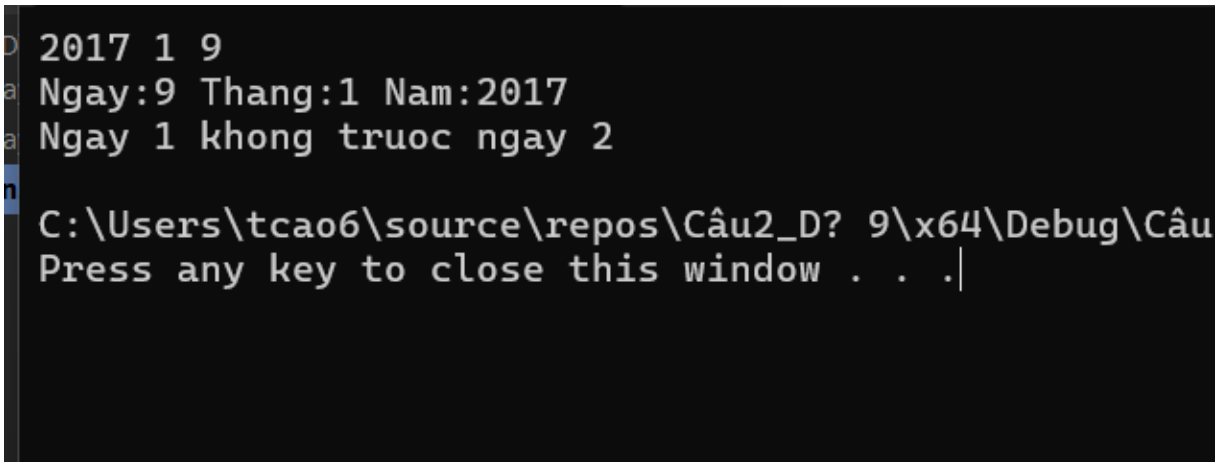
2.4 Code:

Link code Bài 2

2.5 Test case:







3 Bài 3 - Đề 9

3.1 Đề Bài:

Câu 3 (5 điểm):

Công ty quản lý ca sỹ XYZ cần quản lý các thông tin để tính lương cho các ca sỹ thuộc công ty. Giả sử công ty XYZ chia các ca sỹ thành **2 nhóm**: **ca sỹ "chưa" nổi tiếng** và **ca sỹ nổi tiếng**.

Thông tin chung của cả 2 nhóm bao gồm:

- Họ tên ca sỹ.
- Số năm làm việc cho công ty.
- Số đĩa đã bán được.
- Số buổi trình diễn đã tham gia.

Ngoài ra, ca sỹ nổi tiếng được mời tham gia nhiều **Gameshow** nên còn có thêm thông tin:

- Số gameshow tham gia.

Công ty quy định cách tính và trả lương cho ca sỹ như sau:
Cách tính lương:

- **Với ca sỹ "chưa" nổi tiếng:**

$$\begin{aligned}\text{Lương} = 3.000.000 &+ 500.000 \times \text{số năm làm việc} \\ &+ 1.000 \times \text{số đĩa bán được} \\ &+ 200.000 \times \text{số buổi trình diễn}\end{aligned}$$

- **Với ca sỹ nổi tiếng:**

$$\begin{aligned}\text{Lương} = 5.000.000 &+ 500.000 \times \text{số năm làm việc} \\ &+ 1.200 \times \text{số đĩa bán được} \\ &+ 500.000 \times \text{số buổi trình diễn} \\ &+ 500.000 \times \text{số Gameshow tham gia}\end{aligned}$$

Yêu cầu:

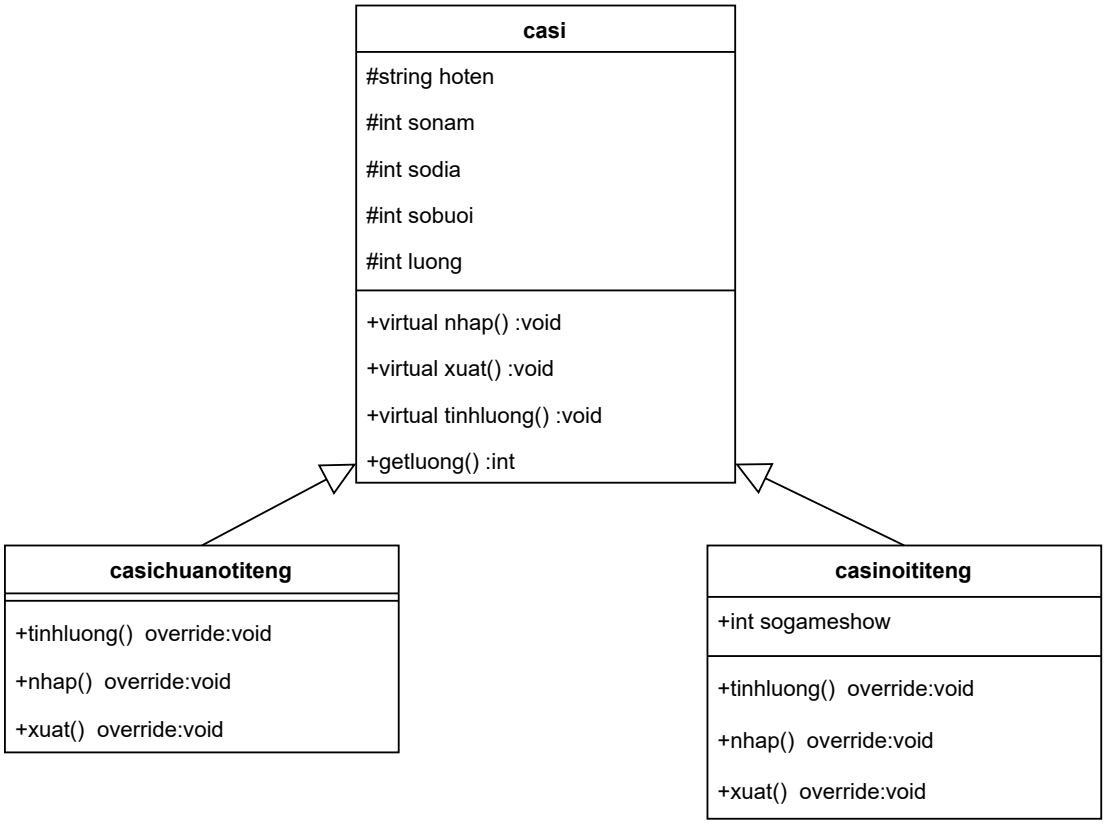
1. **Đề xuất thiết kế các lớp đối tượng cần thiết** (vẽ sơ đồ lớp chi tiết) để quản lý danh sách các ca sỹ của công ty và hỗ trợ tính lương cho ca sỹ theo quy định như trên. (**3 điểm**)
2. **Viết chương trình bằng C++** cho phép thực hiện các yêu cầu sau:

- Nhập danh sách ca sỹ (lưu trữ trong một mảng duy nhất). (**1 điểm**)
- Tìm ca sỹ có lương cao nhất trong công ty. Nếu có nhiều ca sỹ có cùng mức lương cao nhất, chỉ cần trả về **một ca sỹ** trong số đó. (**1 điểm**)

Lưu ý:

- Sử dụng tính chất **kế thừa** và **đa hình**.
- Sử dụng **'string'** để lưu trữ họ tên.
- Vẽ **sơ đồ lớp** mô tả các lớp, các thuộc tính, các hàm và mối liên hệ giữa các lớp. (**1.5 điểm**)
- Khai báo và định nghĩa chi tiết các lớp. (**1.5 điểm**)

3.2 Class Diagram:



3.3 Mô tả và ý tưởng:

Input: Người dùng nhập số lượng ca sĩ cần quản lý. Thông tin từng ca sĩ: Với mỗi ca sĩ, người dùng sẽ nhập:

- Loại ca sĩ: Ca sĩ nổi tiếng (`type = 1`) hoặc ca sĩ chưa nổi tiếng (`type = 2`).
- Thông tin chung của ca sĩ: Họ tên, số năm làm việc, số đĩa đã bán được, số buổi trình diễn đã tham gia.
- Với ca sĩ nổi tiếng (`type = 1`): Số gameshow tham gia cũng cần được nhập.

Output: Cuối cùng, chương trình sẽ in ra thông tin về ca sĩ có lương cao nhất và lương của người đó.

Solution:Sử dụng tính chất đa hình để tạo một vector có kiểu dữ liệu là con trỏ trỏ đến lớp cơ sở ca sĩ để chứa tất cả các ca sĩ. Sau khi nhập tất cả các thông tin và tính lương cho mỗi ca sĩ ta sẽ sắp xếp vector chứa ca sĩ ấy giảm dần theo lương, sau đó hiển thị thông tin ca sĩ có lương cao nhất chính là phần tử đầu tiên của vector.

3.4 Code:

```cpp
#include"casichuanoitieng.h"
#include"casinoitieng.h"
#include<vector>
#include<algorithm>
using namespace std;
bool cmp(casi* s1, casi* s2) {
    if (s1->getluong() > s2->getluong()) {
        return true;
    }
    else {
        return false;
    }
}

int main() {
    int soluong;
    cout << "Nhap so luong ca si: ";
    cin >> soluong;
    vector<casi*>vt;
    for (int i = 0; i < soluong; i++) {
        int type;
        cout << "1: ca si noi tieng" << endl;
        cout << "2:ca si chua noi tieng" << endl;
        cout << "Nhap loai ca si: ";
        cin >> type;

        if (type == 1) {
            casi* singer = new casinoitieng();
            singer->nhap();
            vt.push_back(singer);



        }
        else {
            casi* singer = new casichuanoitieng();
            singer->nhap();
            vt.push_back(singer);

        }
    }

    for (auto x : vt) {
        x->tinhluong();


    }

    sort(vt.begin(), vt.end(), cmp);
```

Hình 1: Link code

3.5 Test case:

```
So dia ban duoc: 8

So buoi trinh dien da tham gia: 5

Nhap so game show: 3
1: ca si noi tieng
2:ca si chua noi tieng
Nhap loai ca si: 2

Nhap thong tin ca sy:
Ho ten: Cao Cong

So nam lam viec: 9

So dia ban duoc: 5

So buoi trinh dien da tham gia: 2

Ca si co luong cao nhat:

Thong tin ca sy:
Ho ten: Cao Thanh
So nam lam viec: 9
So dia ban duoc: 8
So buoi trinh dien: 5
So game show: 3
Luong:13509600
C:\Users\tcao6\source\repos\Cau3_De9\x64\Debug\Cau3_De9.exe (pr
Press any key to close this window . . .
```

```
So buoi trinh dien da tham gia: 7

Nhap so game show: 8
1: ca si noi tieng
2:ca si chua noi tieng
Nhap loai ca si: 1

Nhap thong tin ca sy:
Ho ten: Le Lan

So nam lam viec: 10

So dia ban duoc: 6

So buoi trinh dien da tham gia: 3

Nhap so game show: 0

Ca si co luong cao nhat:

Thong tin ca sy:
Ho ten: Nguyen Nam
So nam lam viec: 9
So dia ban duoc: 5
So buoi trinh dien: 7
So game show: 8
Luong:17006000
C:\Users\tcao6\source\repos\Cau3_De9\x64\Debug\Cau3_De9.exe (process 18652
Press any key to close this window . . .
```

```
Nhap so luong ca si: 1
1: ca si noi tieng
2:ca si chua noi tieng
Nhap loai ca si: 2

Nhap thong tin ca sy:
Ho ten: Le Lai

So nam lam viec: 9

So dia ban duoc: 9

So buoi trinh dien da tham gia: 0

Ca si co luong cao nhat:

Thong tin ca sy:
Ho ten: Le Lai
So nam lam viec: 9
So dia ban duoc: 9
So buoi trinh dien: 0
Luong:7509000
C:\Users\tcao6\source\repos\Cau3_De9\x64\Debug\Cau3_De9.exe (process 7332)
Press any key to close this window . . .
```

4 Bài 1 - Đề 12

a. Phân biệt khái niệm overload (tải chồng) và override (ghi đè).

1. Overload (Tải chồng):
   - **Định nghĩa:** Xảy ra khi trong cùng một lớp, bạn định nghĩa nhiều phương thức có cùng tên nhưng khác nhau về số lượng hoặc kiểu tham số.
   - **Mục đích:** Cho phép sử dụng cùng một tên phương thức nhưng với các tham số khác nhau, giúp phương thức linh hoạt hơn trong việc xử lý nhiều loại dữ liệu hoặc tham số khác nhau.
   - **Đặc điểm:** Thực hiện tại thời gian biên dịch (compile time). Phương thức được xác định và gọi dựa trên số lượng và kiểu tham số.
2. Override (Ghi đè):
   - **Định nghĩa:** Xảy ra khi một lớp con cung cấp một triển khai mới cho một phương thức đã được định nghĩa trong lớp cha.

- **Mục đích:** Cho phép lớp con thay đổi hoặc cung cấp một cách thực hiện khác cho phương thức đã có trong lớp cha, giúp phương thức phù hợp hơn với nhu cầu của lớp con.
- **Đặc điểm:** Thực hiện tại thời gian chạy (runtime). Phương thức gọi phụ thuộc vào đối tượng thực tế tại runtime, không phải là kiểu của biến tham chiếu.

b. Phân biệt các kiểu kế thừa: private, protected, public

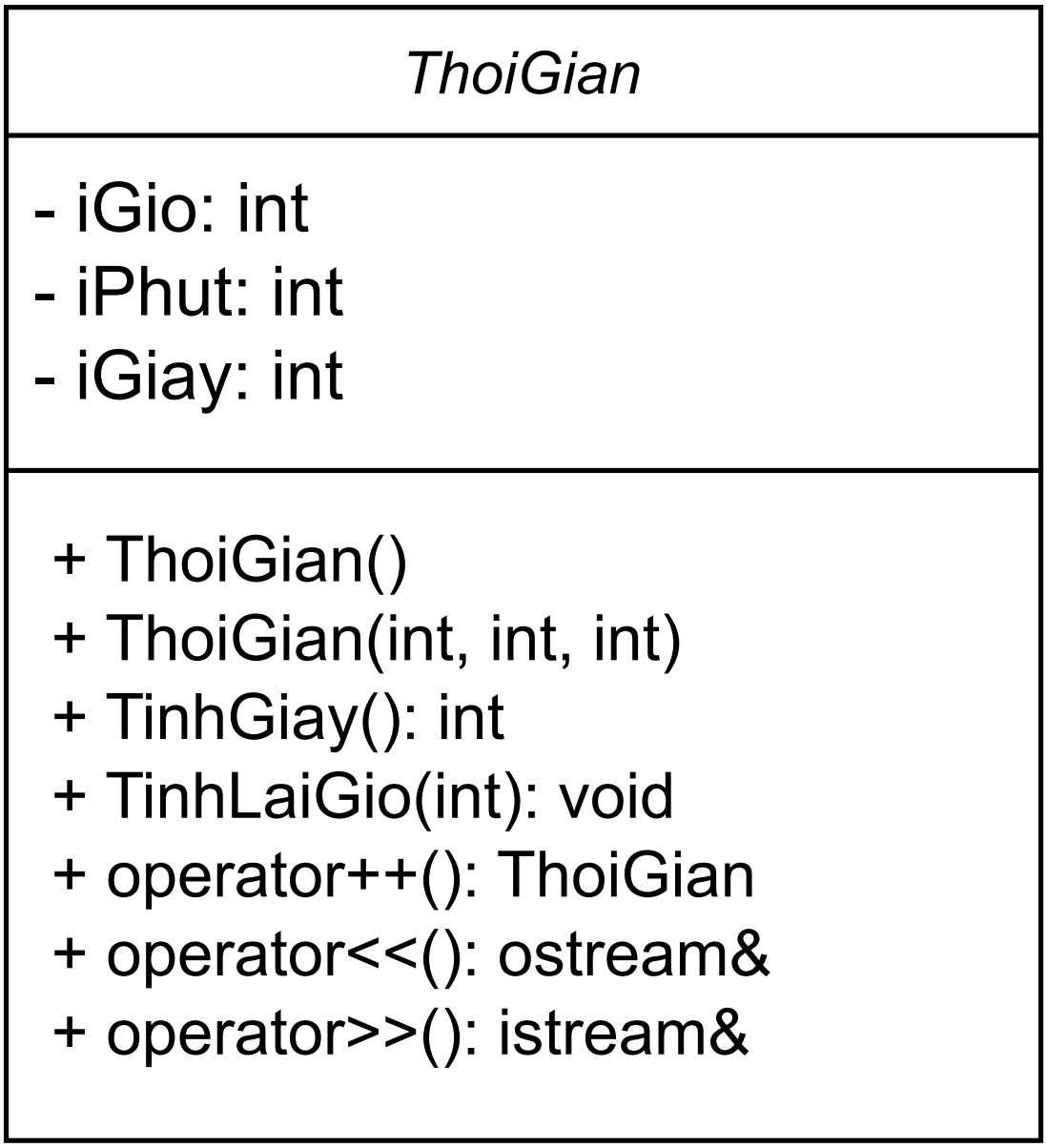
- **Thành phần private:** Các thành phần 'private' của lớp cha không thể truy xuất được từ lớp con. Chúng chỉ có thể được truy cập trong lớp cha.
- **Kế thừa public:** Khi lớp con kế thừa 'public' từ lớp cha:
  - Các thành phần 'protected' của lớp cha trở thành 'protected' trong lớp con.
  - Các thành phần 'public' của lớp cha trở thành 'public' trong lớp con.
- **Kế thừa private:** Khi lớp con kế thừa 'private' từ lớp cha:
  - Các thành phần 'protected' và 'public' của lớp cha trở thành 'private' trong lớp con, không thể truy cập từ bên ngoài.
- **Kế thừa protected:** Khi lớp con kế thừa 'protected' từ lớp cha:
  - Các thành phần 'protected' và 'public' của lớp cha trở thành 'protected' trong lớp con, có thể truy cập từ lớp con nhưng không từ bên ngoài.

5 Bài 2 - Đề 12

5.1 Đề bài:

Xây dựng lớp thời gian(giờ, phút, giây) với các toán tử », « để nhập xuất và toán tử ++ để tăng thời gian thêm 1 giây.

5.2 Class Diagram:



5.3 Mô tả và ý tưởng:

Input: Giờ phút giây của một thời điểm

Output: Giờ phút giây của thời điểm đó sau khi đã tăng 1 giây

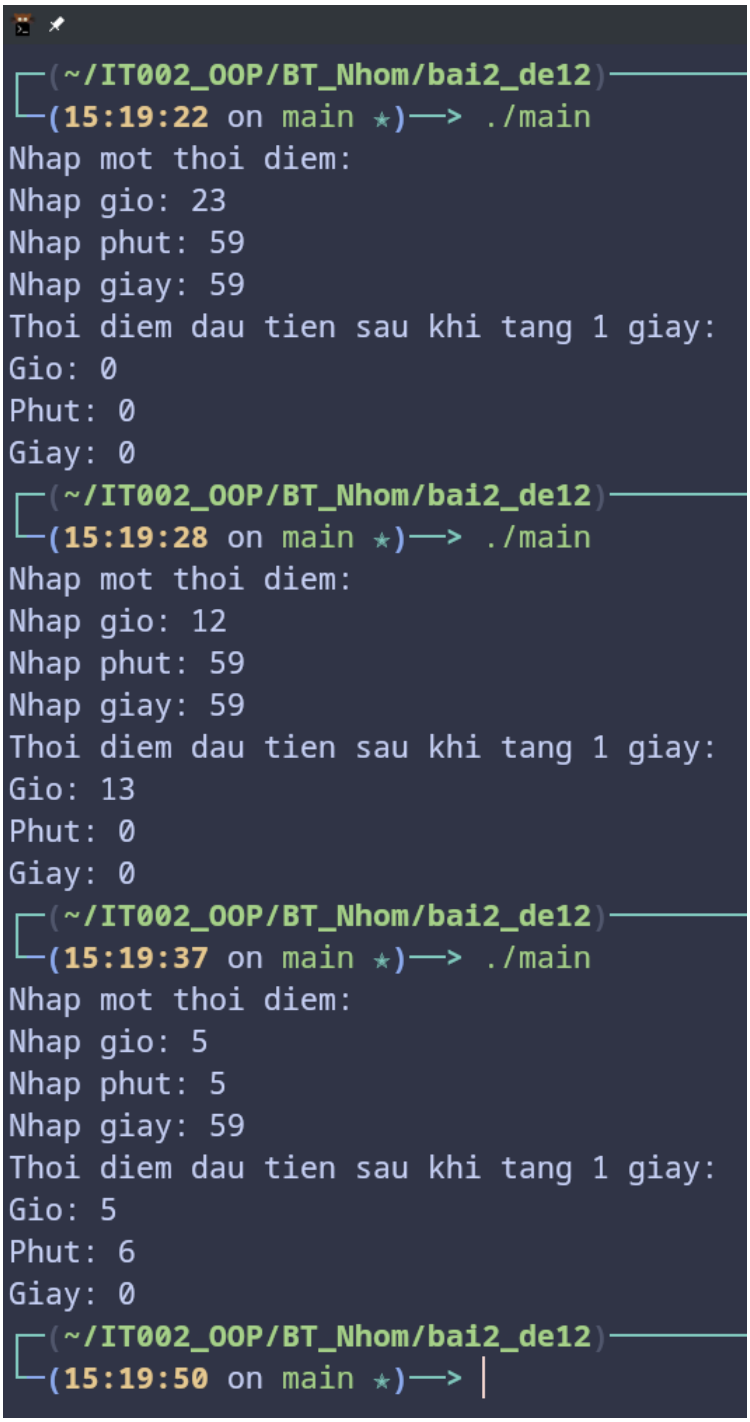
Solution: Sử dụng số giây trong ngày để so sánh các thời điểm trong ngày. Số giây nào nhỏ hơn thì thời điểm đó đến trước trong ngày. Sử dụng 24h format. Các hàm như tính

lại giờ hay cộng trừ thêm giây, đều được đưa về số giây đã qua trong ngày và convert lại thành giờ phút giây.

5.4 Code:

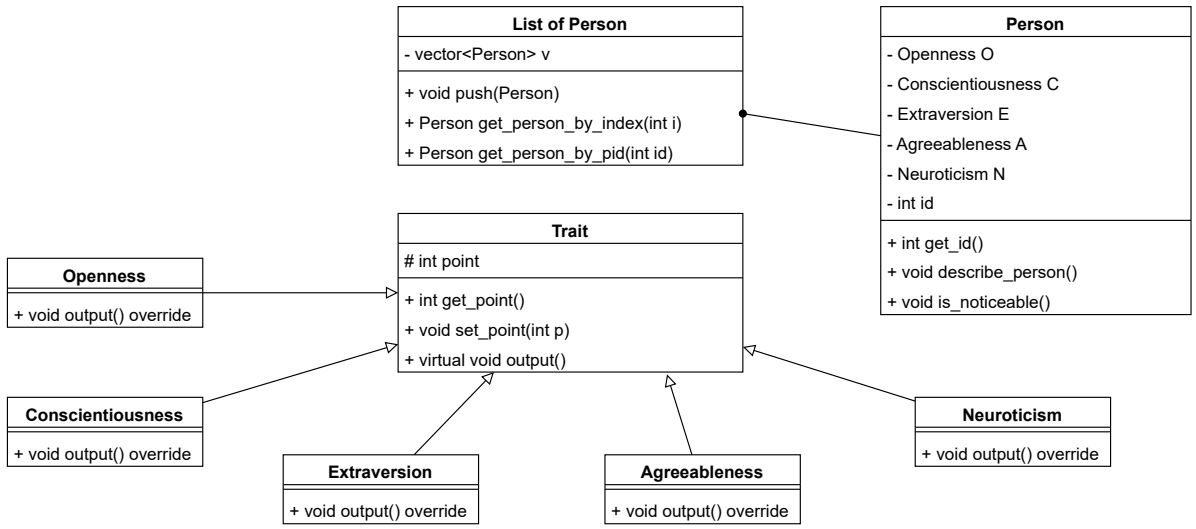
https://github.com/LowTechTurtle/IT002_OOP/tree/main/BT_Nhom/bai2_de12

5.5 TestCase:


~/IT002_OOP/BT_Nhom/bai2_de12
(15:19:22 on main ★)—> ./main
Nhap mot thoi diem:
Nhap gio: 23
Nhap phut: 59
Nhap giay: 59
Thoi diem dau tien sau khi tang 1 giay:
Gio: 0
Phut: 0
Giay: 0
~/IT002_OOP/BT_Nhom/bai2_de12
(15:19:28 on main ★)—> ./main
Nhap mot thoi diem:
Nhap gio: 12
Nhap phut: 59
Nhap giay: 59
Thoi diem dau tien sau khi tang 1 giay:
Gio: 13
Phut: 0
Giay: 0
~/IT002_OOP/BT_Nhom/bai2_de12
(15:19:37 on main ★)—> ./main
Nhap mot thoi diem:
Nhap gio: 5
Nhap phut: 5
Nhap giay: 59
Thoi diem dau tien sau khi tang 1 giay:
Gio: 5
Phut: 6
Giay: 0
~/IT002_OOP/BT_Nhom/bai2_de12
(15:19:50 on main ★)—>

6 Bài 3 - Đề 12

6.1 Class Diagram:



6.2 Mô tả và ý tưởng:

Input: Nhập vào số người và kết quả đánh giá tâm lý theo OCEAN của người đó, ngoài ra còn nhập thêm id(pid - person id) cho người đó để có thể xem tính cách của một người trong số các người đã nhập(theo yêu cầu đề bài)

Output: In ra các đánh giá về tâm lý theo big five OCEAN và dự đoán xem người đó có bị các nhà tuyển dụng lưu ý hay không.

Solution: Tạo một lớp Trait để các lớp Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism kế thừa. Các lớp này đều có chung đặc điểm là chứa số điểm của tính cách đó. Sau đó tạo lớp Person và cho lớp người có 6 đặc điểm gồm id, và 5 đặc điểm tính cách trong OCEAN. Sau đó tạo lớp List_of_Person để chứa nhiều người.

6.3 Code:

https://github.com/LowTechTurtle/IT002_OOP/tree/main/BT_Nhom/bai3_de12

6.4 TestCase:

Log của chương trình sau khi chạy:

Test 1

```
1 Nhap so nguoi: 2
```

```
Nhap ket danh gia tam ly cua nguoi thu 1 : O93-C74-E31-A96-N5
Nhap id cua nguoi do 1
Nhap ket danh gia tam ly cua nguoi thu 2 : O70-C30-E60-A96-N10
Nhap id cua nguoi do 2
Hay chon mot nguoi de in ra tinh cach cua nguoi do
Nhap pid cua nguoi do: 2
San sang trai nghiem: Nguoi nay thich nhung y tuong moi thich hieu biet linh
    vuc nhung dong thoi cung thich tu do, khong bi rang buoc
Tu chu tan tam: Nguoi nay thuong de bo cuoc, kha nang chiu ap luc thap
Huong ngoai: Khong the ket luan ve nguoi nay
Hoa dong de chiu: Nguoi nay than thien coi mo, dong cam voi moi nguoi
Bat on cam xuc: Nguoi nay thuong kiem soat duoc cam xuc,ung pho voi cang thang
    tot
Nguoi nay co nguy co bi nha tuyen dung luu y
```

Test 2

```
Nhap so nguoi: 2
Nhap ket danh gia tam ly cua nguoi thu 1 : O93-C74-E31-A96-N5
Nhap id cua nguoi do 1
Nhap ket danh gia tam ly cua nguoi thu 2 : O70-C30-E60-A96-N10
Nhap id cua nguoi do 2
Hay chon mot nguoi de in ra tinh cach cua nguoi do
Nhap pid cua nguoi do: 1
San sang trai nghiem: Nguoi nay thich nhung y tuong moi thich hieu biet linh
    vuc nhung dong thoi cung thich tu do, khong bi rang buoc
Tu chu tan tam: Nguoi nay thuong la nguoi cham chi co kha nang chiu ap luc
    tot, la nguoi gan bo, trung thanh voi to chuc
Huong ngoai: Khong the ket luan ve nguoi nay
Hoa dong de chiu: Nguoi nay than thien coi mo, dong cam voi moi nguoi
Bat on cam xuc: Nguoi nay thuong kiem soat duoc cam xuc,ung pho voi cang thang
    tot
Nguoi nay khong co nguy co cao bi nha tuyen dung luu y
```