**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**



**MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

**BÀI TẬP THỰC HÀNH 4**

GVHD: Nguyễn Ngọc Quí

Sinh viên thực hiện: Bùi Ngọc Thái

🙡🙢 Tp. Hồ Chí Minh, 02/2024 🙠🙣

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*……., ngày……...tháng……năm 2024*

**Người nhận xét**

*(Ký tên và ghi rõ họ tên****)***

**MỤC LỤC**

[DANH MỤC HÌNH ẢNH 4](#_Toc167560021)

[NỘI DUNG BÀI LÀM 7](#_Toc167560022)

[**Bài tập 1 7**](#_Toc167560023)

[**1. Thực hiện xây dựng lớp 8**](#_Toc167560024)

[**2. Class diagram của bài tập 1 9**](#_Toc167560025)

[**3. Thuộc tính, phương thức của các lớp 10**](#_Toc167560026)

[**4. Xác định Input, Output 14**](#_Toc167560027)

[**5. Hàm main của chương trình 14**](#_Toc167560028)

[**6. Kết quả chạy của chương trình 15**](#_Toc167560029)

[**Bài tập 2 15**](#_Toc167560030)

[**1. Thực hiện xây dựng lớp 16**](#_Toc167560031)

[**2. Class diagram của bài tập 2 17**](#_Toc167560032)

[**3. Thuộc tính, phương thức của các lớp 18**](#_Toc167560033)

[**4. Xác định input, output: 20**](#_Toc167560034)

[**5. Hàm main của chương trình 20**](#_Toc167560035)

[**6. Kết quả chạy của chương trình 22**](#_Toc167560036)

[**Bài tập 3: 23**](#_Toc167560037)

[**1. Thực hiện xây dựng lớp 23**](#_Toc167560038)

[**2. Class diagram của bài tập 3 25**](#_Toc167560039)

[**3. Thuộc tính, phương thức của các lớp 26**](#_Toc167560040)

[**4. Xác định input, output 31**](#_Toc167560041)

[**5. Hàm main của chương trình 31**](#_Toc167560042)

[**6. Kết quả chạy của chương trình 32**](#_Toc167560043)

[**Mã nguồn: 32**](#_Toc167560044)

[TÀI LIỆU THAM KHẢO 33](#_Toc167560045)

# DANH MỤC HÌNH ẢNH

[Hình 1.1: Thực hiện xây dựng lớp NhanVien 8](#_Toc167560135)

[Hình 1.2: Thực hiện xây dựng lớp QuanLy 8](#_Toc167560136)

[Hình 1.3: Thực hiện xây dựng lớp KySu 9](#_Toc167560137)

[Hình 1.4: Class diagram của bài tập 1 9](#_Toc167560138)

[Hình 1.5: Constructor mặc định của lớp NhanVien 10](#_Toc167560139)

[Hình 1.6: Constructor với tham số của lớp NhanVien 10](#_Toc167560140)

[Hình 1.7: Hàm chuẩn hóa tên để đảm bảo nhập và hiển thị đúng thông tin 11](#_Toc167560141)

[Hình 1.8: Phương thức Nhap() của lớp NhanVien 11](#_Toc167560142)

[Hình 1.9: Phương thức Xuat() của lớp NhanVien 11](#_Toc167560143)

[Hình 1.10: Constructor mặc định của lớp QuanLy 12](#_Toc167560144)

[Hình 1.11: Constructor với tham số của lớp QuanLy 12](#_Toc167560145)

[Hình 1.12: Phương thức TienThuong() của lớp QuanLy 12](#_Toc167560146)

[Hình 1.13: Phương thức Nhap() của lớp QuanLy ghi đè từ lớp NhanVien 12](#_Toc167560147)

[Hình 1.14: Phương thức Xuat() của lớp QuanLy ghi đè từ lớp NhanVien 12](#_Toc167560148)

[Hình 1.15: Constructor mặc định của lớp KySu 13](#_Toc167560149)

[Hình 1.16: Constructor với tham số của lớp KySu 13](#_Toc167560150)

[Hình 1.17: Phương thức TienThuong() của lớp KySu 13](#_Toc167560151)

[Hình 1.18: Phương thức Nhap() của lớp KySu ghi đè từ lớp NhanVien 13](#_Toc167560152)

[Hình 1.19: Phương thức Xuat() của lớp KySu ghi đè từ lớp NhanVien 14](#_Toc167560153)

[Hình 1.20: Hàm main của chương trình bài tập 1 14](#_Toc167560154)

[Hình 1.21: Kết quả chạy của chương trình bài tập 1 15](#_Toc167560155)

[Hình 2.1: Thực hiện xây dung lớp Room 16](#_Toc167560156)

[Hình 2.2: Thực hiện xây dựng lớp Deluxe 16](#_Toc167560157)

[Hình 2.3: Thực hiện xây dựng lớp Premium 17](#_Toc167560158)

[Hình 2.4: Thực hiện xây dựng lớp Business 17](#_Toc167560159)

[Hình 2.5: Class diagram của bài tập 2 17](#_Toc167560160)

[Hình 2.6: Phương thức doanhThu() của lớp Deluxe 18](#_Toc167560161)

[Hình 2.7: Phương thức loaiPhong() của lớp Deluxe 18](#_Toc167560162)

[Hình 2.8: Phương thức doanhThu() của lớp Premium 19](#_Toc167560163)

[Hình 2.9: Phương thức loaiPhong() của lớp Premium 19](#_Toc167560164)

[Hình 2.10: Phương thức doanhThu() của lớp Business 19](#_Toc167560165)

[Hình 2.11: Phương thức loaiPhong() của lớp Business 19](#_Toc167560166)

[Hình 2.12: Hàm toàn cục taoPhong 20](#_Toc167560167)

[Hình 2.13: Hàm main của chương trình bài tập 2 (1) 20](#_Toc167560168)

[Hình 2.14: Hàm main của chương trình bài tập 2 (2) 21](#_Toc167560169)

[Hình 2.15: Hàm main của chương trình bài tập 2 (3) 21](#_Toc167560170)

[Hình 2.16: Kết quả chạy của chương trình bài tập 2 22](#_Toc167560171)

[Hình 3.1: Thực hiện xây dựng lớp GiaSuc 23](#_Toc167560172)

[Hình 3.2: Thực hiện xây dựng lớp Bo 24](#_Toc167560173)

[Hình 3.3: Thực hiện xây dựng lớp Cuu 24](#_Toc167560174)

[Hình 3.4: Thực hiện xây dựng lớp De 24](#_Toc167560175)

[Hình 3.5: Class diagram của bài tập 3 25](#_Toc167560176)

[Hình 3.6: Constructor mặc định và constructor có tham số của lớp GiaSuc 26](#_Toc167560177)

[Hình 3.7: Phương thức Nhap() của lớp GiaSuc 26](#_Toc167560178)

[Hình 3.8: Phương thức keu() của lớp GiaSuc 27](#_Toc167560179)

[Hình 3.9: Phương thức tinhTongLitSua() của lớp GiaSuc 27](#_Toc167560180)

[Hình 3.10: Phương thức sinhCon() của lớp GiaSuc 27](#_Toc167560181)

[Hình 3.11: Phương thức getSoLuongCon() của lớp GiaSuc 28](#_Toc167560182)

[Hình 3.12: Phương thức choSua() của lớp GiaSuc 28](#_Toc167560183)

[Hình 3.13: Phương thức getSoLuong() của lớp GiaSuc 28](#_Toc167560184)

[Hình 3.14: Hàm hủy của lớp GiaSuc 28](#_Toc167560185)

[Hình 3.15: Constructor mặc định và constructor có tham số của lớp Bo 28](#_Toc167560186)

[Hình 3.16: Phương thức keuMotLan() của lớp Bo 29](#_Toc167560187)

[Hình 3.17: Constructor mặc định và constructor có tham số của lớp Cuu 29](#_Toc167560188)

[Hình 3.18: Phương thức keuMotLan() của lớp Cuu 29](#_Toc167560189)

[Hình 3.19: Phương thức choSua() của lớp Cuu 29](#_Toc167560190)

[Hình 3.20: Constructor mặc định và constructor có tham số của lớp De 30](#_Toc167560191)

[Hình 3.21: Phương thức keuMotLan() của lớp De 30](#_Toc167560192)

[Hình 3.22: Phương thức choSua() của lớp De 30](#_Toc167560193)

[Hình 3.23: Hàm thống kê các thông tin người chủ mong muốn 30](#_Toc167560194)

[Hình 3.24: Hàm main của chương trình bài tập 3 31](#_Toc167560195)

[Hình 3.25: Kết quả chạy của chương trình bài tập 3 32](#_Toc167560196)

# NỘI DUNG BÀI LÀM

Bài tập 1: Quản lý thông tin nhân viên

Viết một chương trình C++ để quản lý thông tin nhân viên trong một công ty. Sử dụng

tính kế thừa để tạo lớp cơ sở NhanVien và hai lớp con Quản lý và Kỹ sư. Mỗi loại nhân viên có các thuộc tính như sau:

* Nhân viên: Mã số nhân viên, Tên, Lương cơ bản.
* Quản lý: Thừa kế từ Nhân viên với thêm thuộc tính là Tỷ lệ thưởng. Phương thức

TienThuong () tính toán tiền thưởng dựa trên tỷ lệ thưởng và lương cơ bản.

* Kỹ sư: Thừa kế từ Nhân viên với thêm thuộc tính là Số giờ làm thêm. Phương

thức TienThuong () tính toán tiền thưởng dựa trên số giờ làm thêm (mỗi giờ làm

thêm được trả 100.000).

Chương trình cần thực hiện các công việc sau:

* Cho phép người dùng nhập thông tin của một quản lý và một kỹ sư.
* Hiển thị thông tin chi tiết của cả hai nhân viên, bao gồm mã số, tên, lương cơ bản và tiền thưởng tính được.
* Sử dụng tính kế thừa để thực hiện việc tính toán tiền thưởng dựa trên loại nhân viên.
* Sử dụng hàm Xuat() để hiển thị thông tin của nhân viên.

**Lưu ý: Đảm bảo rằng chương trình có khả năng nhập và hiển thị thông tin đúng**

**cho cả quản lý và kỹ sư.**

***Gợi ý:***

* Sử dụng các lớp và tính kế thừa để cấu trúc dữ liệu nhân viên.
* Sử dụng phương thức TienThuong() để tính toán tiền thưởng.
* Sử dụng phương thức Xuat() để hiển thị thông tin.

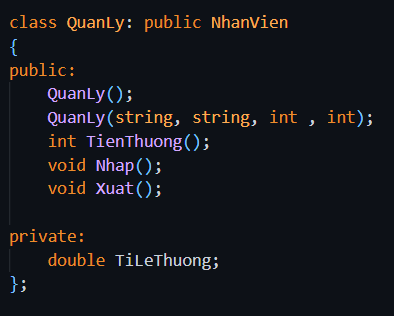
### Thực hiện xây dựng lớp

* Thực hiện xây dựng lớp NhanVien (lớp cha):



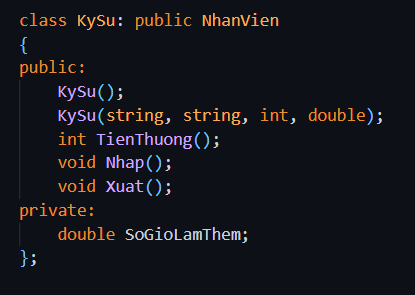
Hình 1.1: Thực hiện xây dựng lớp NhanVien

* Thực hiện xây dựng lớp QuanLy (kế thừa lớp NhanVien):



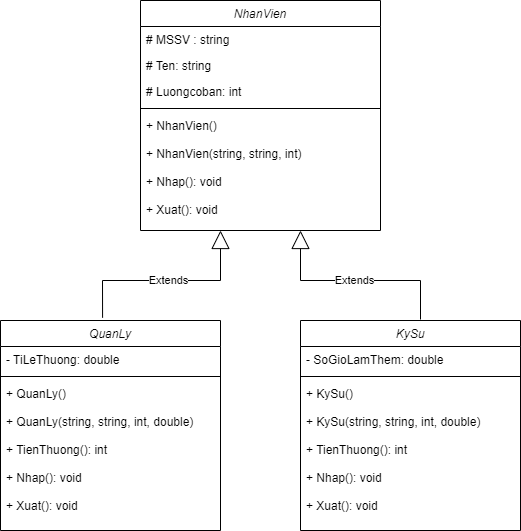
Hình 1.2: Thực hiện xây dựng lớp QuanLy

* Thực hiện xây dựng lớp KySu (kế thừa lớp NhanVien):



Hình 1.3: Thực hiện xây dựng lớp KySu

### Class diagram của bài tập 1

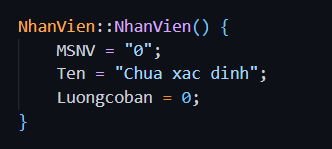


Hình 1.4: Class diagram của bài tập 1

### Thuộc tính, phương thức của các lớp

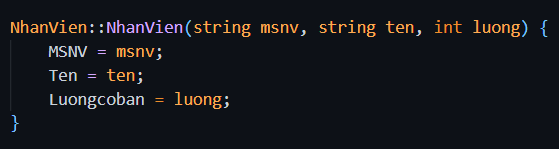
#### Lớp NhanVien

* Thuộc tính:
* MSNV: Mã số nhân viên (kiểu string).
* Ten: Tên nhân viên (kiểu string).
* Luongcoban: Lương cơ bản (kiểu int).
* Phương thức:
* NhanVien(): Constructor mặc định, khởi tạo mặc định mã số nhân viên là 0, tên nhân viên “Chua xac dinh” và lương cơ bản bằng 0.



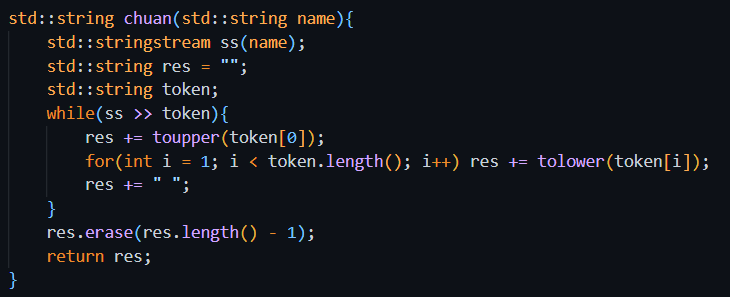
Hình 1.5: Constructor mặc định của lớp NhanVien

* NhanVien(string, string, int): Constructor với tham số.



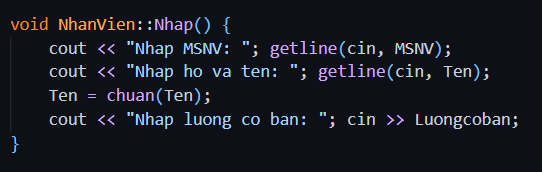
Hình 1.6: Constructor với tham số của lớp NhanVien

* Nhap(): Phương thức nhập thông tin nhân viên.
* Vì để đảm bảo rằng chương trình có khả năng nhập và hiển thị thông tin đúng cho cả quản lý và kỹ sư, ta xây dựng hàm chuẩn hóa tên sao cho hợp định dạng tên. Hàm chuẩn hóa được xây dựng trong đoạn chương trình như ảnh dưới đây.



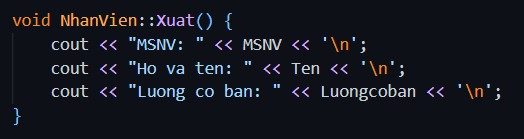
Hình 1.7: Hàm chuẩn hóa tên để đảm bảo nhập và hiển thị đúng thông tin

* Từ hàm, ta có được phương thức Nhap():



Hình 1.8: Phương thức Nhap() của lớp NhanVien

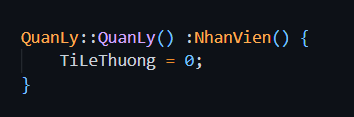
* Xuat(): Phương thức xuất thông tin nhân viên.



Hình 1.9: Phương thức Xuat() của lớp NhanVien

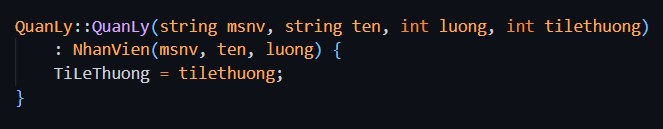
#### Lớp QuanLy (kế thừa từ lớp NhanVien)

* Thuộc tính:
* TiLeThuong: Tỉ lệ thưởng (kiểu double).
* Phương thức:
* QuanLy(): Constructor mặc định., khởi tạo mặc định tỉ lệ thưởng bằng 0.



Hình 1.10: Constructor mặc định của lớp QuanLy

* QuanLy(string, string, int , int): Constructor với tham số.



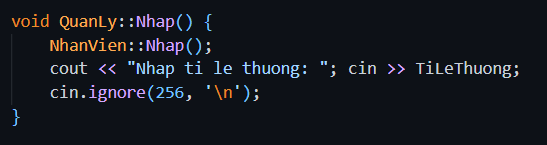
Hình 1.11: Constructor với tham số của lớp QuanLy

* TienThuong(): Tính tiền thưởng dựa trên tỷ lệ thưởng và lương cơ bản.



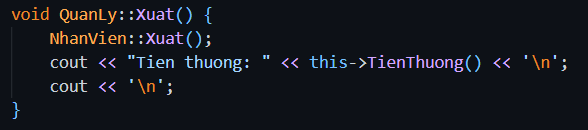
Hình 1.12: Phương thức TienThuong() của lớp QuanLy

* Nhap(): Nhập thông tin quản lý, ghi đè phương thức Nhap() từ lớp NhanVien.



Hình 1.13: Phương thức Nhap() của lớp QuanLy ghi đè từ lớp NhanVien

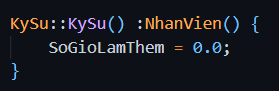
* Xuat(): Xuất thông tin quản lý, ghi đè phương thức Xuat() từ lớp NhanVien.



Hình 1.14: Phương thức Xuat() của lớp QuanLy ghi đè từ lớp NhanVien

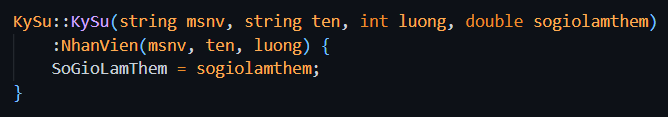
#### Lớp KySu (kế thừa từ lớp NhanVien)

* Thuộc tính:
* SoGioLamThem: Số giờ làm thêm (kiểu double).
* Phương thức:
* KySu(): Constructor mặc định của lớp KySu, khởi tạo mặc định số giờ làm thêm bằng 0.



Hình 1.15: Constructor mặc định của lớp KySu

* KySu(string, string, int, double): Constructor với tham số của lớp KySu.



Hình 1.16: Constructor với tham số của lớp KySu

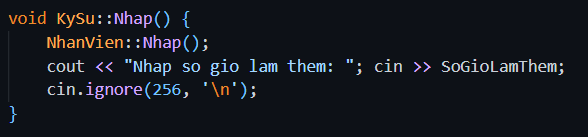
* TienThuong(): Tính tiền thưởng dựa trên số giờ làm thêm (mỗi giờ làm

thêm được trả 100.000).



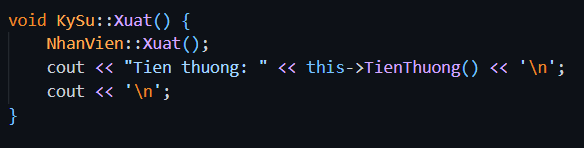
Hình 1.17: Phương thức TienThuong() của lớp KySu

* Nhap(): Nhập thông tin kỹ sư, ghi đè phương thức Nhap() từ lớp NhanVien.



Hình 1.18: Phương thức Nhap() của lớp KySu ghi đè từ lớp NhanVien

* Xuat(): Xuất thông tin kỹ sư. ghi đè phương thức Xuat() từ lớp NhanVien.

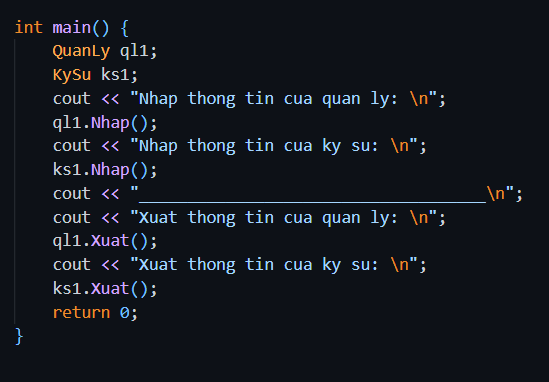


Hình 1.19: Phương thức Xuat() của lớp KySu ghi đè từ lớp NhanVien

### Xác định Input, Output

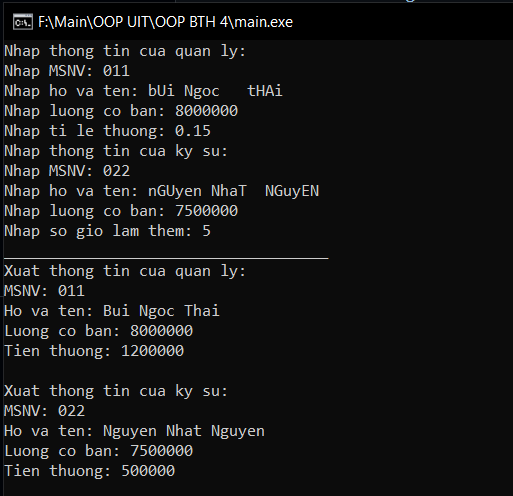
* Input: Thông tin của một quản lý và một kỹ sư.
* Output: Hiển thị thông tin chi tiết của cả hai nhân viên, bao gồm mã số, tên, lương cơ bản và tiền thưởng tính được.

### Hàm main của chương trình



Hình 1.20: Hàm main của chương trình bài tập 1

### Kết quả chạy của chương trình



Hình 1.21: Kết quả chạy của chương trình bài tập 1

Bài tập 2: Khách sạn SOFITEL có 3 loại phòng với doanh thu phòng của tháng được

tính như sau:

* Loại Deluxe: doanh thu phòng = Số đêm \* 750000 + Phí dịch vụ + Phí phục vụ
* Loại Premium: doanh thu phòng = Số đêm \* 500000 + Phí dịch vụ
* Loại Business: doanh thu phòng = Số đêm \* 300000

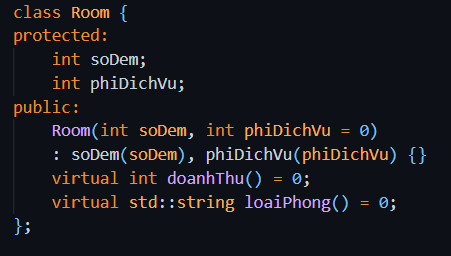
**Yêu cầu:** Hàm main: khởi tạo 5 phòng: Ví dụ: Deluxe a(...); Deluxe b(...); Premium

c(...); Premium d(...); Business e(...). Hãy cho biết doanh thu của loại phòng nào nhiều

nhất? (lưu ý: mỗi loại phòng có thể khởi tạo nhiều phòng).

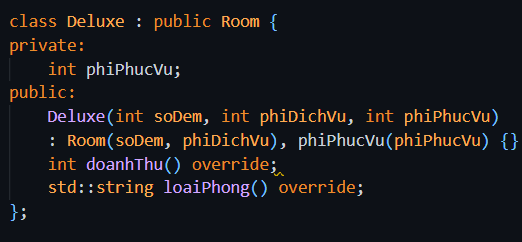
### Thực hiện xây dựng lớp

* Thực hiện xây dựng lớp Room, trong đó đã xây dựng constructor có tham số của lớp Room, khởi tạo với phí dịch vụ bằng 0.



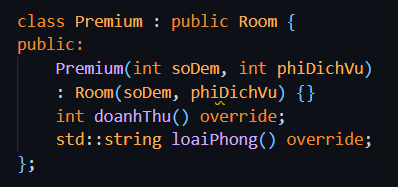
Hình 2.1: Thực hiện xây dung lớp Room

* Thực hiện xây dựng lớp Deluxe (kế thừa lớp Room), trong đó đã xây dựng constructor có tham số của lớp Deluxe.



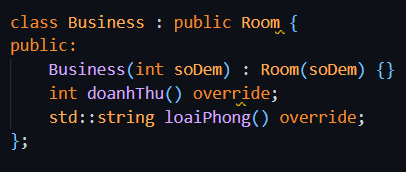
Hình 2.2: Thực hiện xây dựng lớp Deluxe

* Thực hiện xây dựng lớp Premium (kế thừa lớp Room), trong đó đã khởi tạo constructor có tham số của lớp Premium.



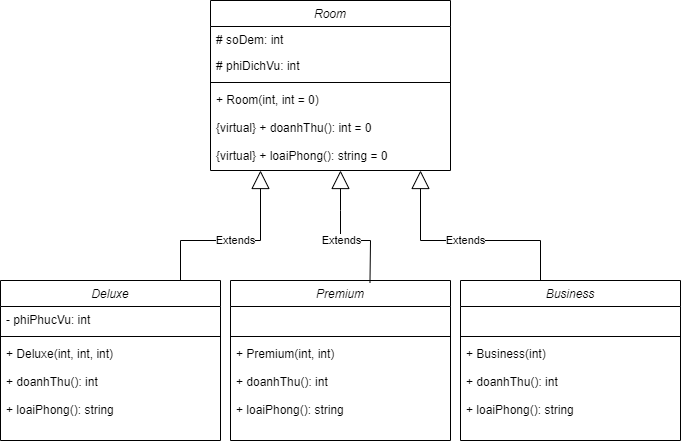
Hình 2.3: Thực hiện xây dựng lớp Premium

* Thực hiện xây dựng lớp Business (kế thừa lớp Room), trong đó đã xây dựng constructor có tham số của lớp Business



Hình 2.4: Thực hiện xây dựng lớp Business

### Class diagram của bài tập 2



Hình 2.5: Class diagram của bài tập 2

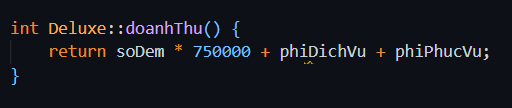
### Thuộc tính, phương thức của các lớp

#### Lớp Room (Lớp trừu tượng)

* Thuộc tính:
* soDem: Số đêm ở lại phòng.
* phiDichVu: Phí dịch vụ áp dụng cho phòng.
* Phương thức: Đã được thể hiện trong phần thực hiện xây dựng lớp Room
* Room(int soDem, int phiDichVu = 0): Hàm khởi tạo cho lớp Room.
* virtual int doanhThu() = 0: Phương thức ảo thuần túy để tính doanh thu, sẽ được triển khai trong các lớp con.
* virtual std::string loaiPhong() = 0: Phương thức ảo thuần túy để trả về loại phòng, sẽ được triển khai trong các lớp con.

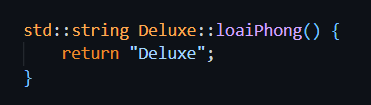
#### Lớp Deluxe (Kế thừa từ Room)

* Thuộc tính:
* private int phiPhucVu: Phí phục vụ áp dụng cho phòng Deluxe.
* Phương thức:
* public Deluxe(int soDem, int phiDichVu, int phiPhucVu): Hàm khởi tạo cho lớp Deluxe.
* public int doanhThu() override: Tính doanh thu của phòng Deluxe, ghi đè phương thức doanhThu() từ lớp Room.



Hình 2.6: Phương thức doanhThu() của lớp Deluxe

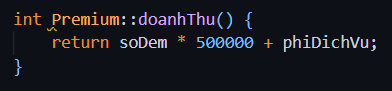
* public std::string loaiPhong() override: Trả về loại phòng "Deluxe", ghi đè phương thức loaiPhong() từ lớp Room.



Hình 2.7: Phương thức loaiPhong() của lớp Deluxe

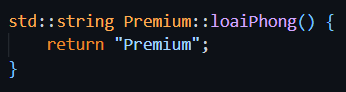
#### Lớp Premium (Kế thừa từ Room)

* Thuộc tính: Không có thuộc tính riêng
* Phương thức:
* public Premium(int soDem, int phiDichVu): Hàm khởi tạo cho lớp Premium.
* public int doanhThu() override: Tính doanh thu của phòng Premium.



Hình 2.8: Phương thức doanhThu() của lớp Premium

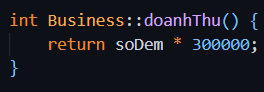
* public std::string loaiPhong() override: Trả về loại phòng "Premium".



Hình 2.9: Phương thức loaiPhong() của lớp Premium

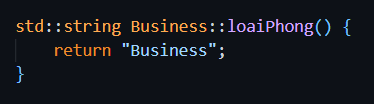
#### Lớp Business (Kế thừa từ Room)

* Thuộc tính: Không có thuộc tính riêng
* Phương thức:
* public Business(int soDem): Hàm khởi tạo cho lớp Business.
* public int doanhThu() override: Tính doanh thu của phòng Business.



Hình 2.10: Phương thức doanhThu() của lớp Business

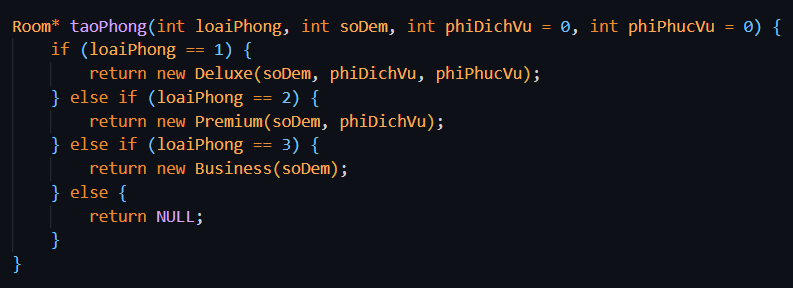
* public std::string loaiPhong() override: Trả về loại phòng "Business".



Hình 2.11: Phương thức loaiPhong() của lớp Business

#### Hàm toàn cục taoPhong

* Phương thức: Room\* taoPhong(int loaiPhong, int soDem, int phiDichVu = 0, int phiPhucVu = 0): Tạo đối tượng phòng dựa trên loại phòng và các thông số đầu vào.

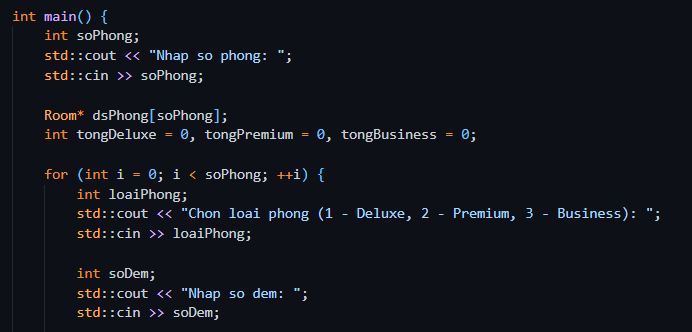


Hình 2.12: Hàm toàn cục taoPhong

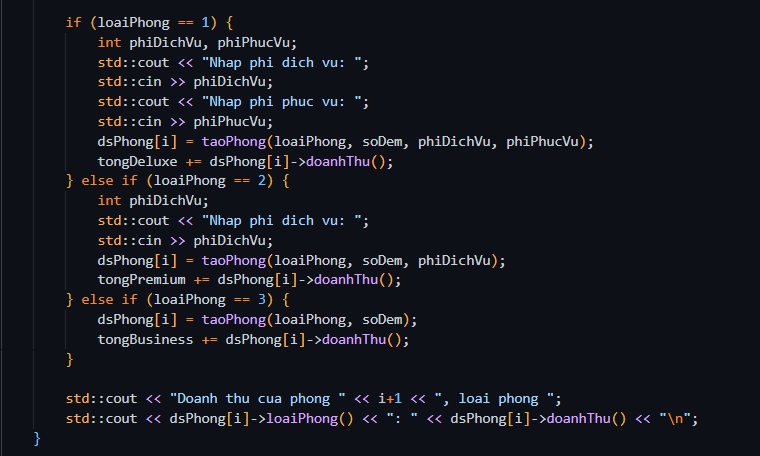
### Xác định input, output:

* Input: Số lượng phòng từ người dùng. Với mỗi phòng, nhập loại phòng, sau đó nhập số đêm, phí dịch vụ và phí phục vụ (nếu có) của phòng.
* Output: Doanh thu của phòng, thể loại của phòng đó. Loại phòng có doanh thu cao nhất.

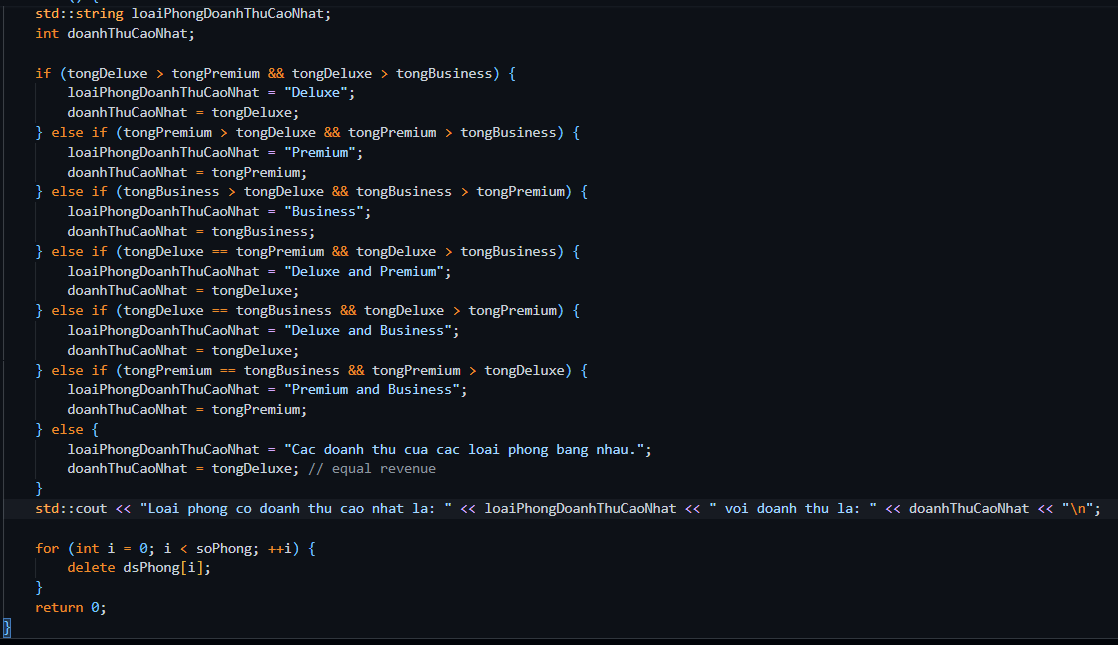
### Hàm main của chương trình



Hình 2.13: Hàm main của chương trình bài tập 2 (1)



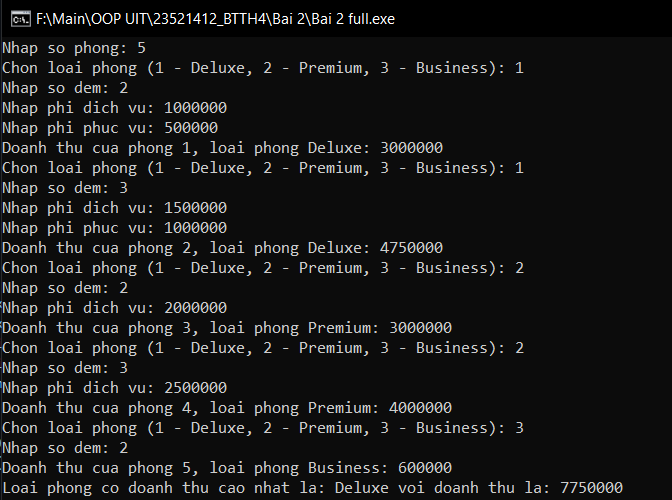
Hình 2.14: Hàm main của chương trình bài tập 2 (2)



Hình 2.15: Hàm main của chương trình bài tập 2 (3)

* Người dùng nhập số lượng phòng, sau đó ta tạo một mảng con trỏ đến các đối tượng Room.
* Tiếp theo, duyệt qua từng phòng và nhập thông tin chi tiết về loại phòng và số đêm ở. Dựa trên loại phòng, ta nhập thông tin thêm và sau đó tạo một đối tượng Room tương ứng.
* Sau đó tính toán doanh thu cho mỗi loại phòng và in ra doanh thu cho mỗi phòng.
* Cuối cùng, xác định loại phòng có doanh thu cao nhất và in ra thông tin này. Trước khi kết thúc, ta giải phóng bộ nhớ đã được cấp phát cho các đối tượng Room.

### Kết quả chạy của chương trình



Hình 2.16: Kết quả chạy của chương trình bài tập 2

Bài tập 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 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 nhập vào số

lượng gia súc ban đầu ở mỗi loại.

1. 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.
2. 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.
* Số lượng sinh của mỗi gia súc là ngẫu nhiên.
* Tất cả gia súc mỗi loại đều cho sữa.
* Số lượng sinh của mỗi gia súc cho sữa là ngẫu nhiên nhưng trong giới hạn sau:
* Bò: 0 – 20 lít
* Cừu: 0 – 5 lít
* Dê: 0 – 10 lít

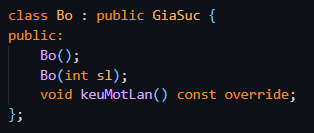
### Thực hiện xây dựng lớp

* Thực hiện xây dựng lớp GiaSuc



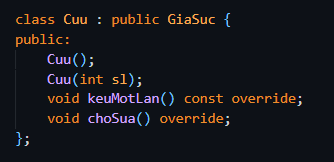
Hình 3.1: Thực hiện xây dựng lớp GiaSuc

* Thực hiện xây dựng lớp Bo



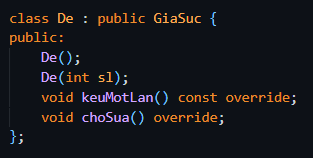
Hình 3.2: Thực hiện xây dựng lớp Bo

* Thực hiện xây dựng lớp Cuu



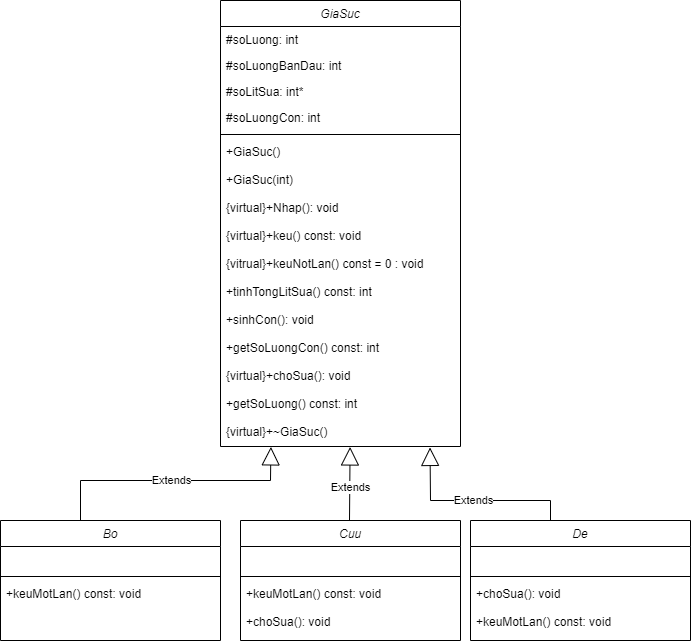
Hình 3.3: Thực hiện xây dựng lớp Cuu

* Thực hiện xây dựng lớp De



Hình 3.4: Thực hiện xây dựng lớp De

### Class diagram của bài tập 3



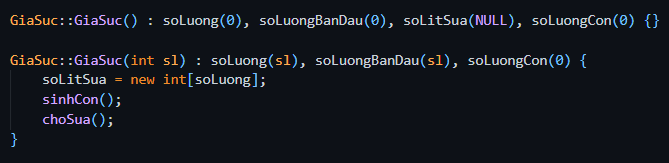
Hình 3.5: Class diagram của bài tập 3

* Theo yêu cầu của đề bài, vì số lượng sinh của mỗi gia súc là ngẫu nhiên, và số lượng sữa mỗi vật mỗi loài là ngẫu nhiên nhưng trong giới hạn theo đề cho. Vì vậy ta khai báo thư viện <cstdlib> của C++ , sử dụng thư viện này để tạo ra các số ngẫu nhiên. Chính vì thế hàm rand() được sử dụng trong mã để mô phỏng quá trình sinh sản và sản xuất sữa của gia súc
* Sử dụng hàm rand() giúp mô phỏng quá trình này một cách ngẫu nhiên, tạo ra sự đa dạng và không dự đoán trước được, giống như trong thực tế. Để có kết quả ngẫu nhiên tốt hơn, thường người ta sẽ gọi hàm srand() với một tham số là thời gian hiện tại (time(NULL)) trước khi gọi rand(), nhằm khởi tạo bộ sinh số ngẫu nhiên với một điểm bắt đầu khác nhau mỗi lần chạy chương trình. Điều này giúp cho các số ngẫu nhiên không bị lặp lại mỗi lần chạy chương trình.

### Thuộc tính, phương thức của các lớp

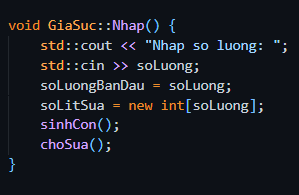
#### Lớp GiaSuc

* Thuộc tính:
* soLuong (int): Số lượng gia súc hiện có.
* soLuongBanDau (int): Số lượng gia súc ban đầu khi khởi tạo.
* soLitSua (int\*): Mảng chứa số lít sữa mỗi gia súc có thể cho.
* soLuongCon (int): Tổng số lượng con mà gia súc đã sinh.
* Phương thức:
* GiaSuc() và GiaSuc(int sl): Các hàm tạo, khởi tạo số lượng gia súc và cấp phát bộ nhớ cho mảng soLitSua.



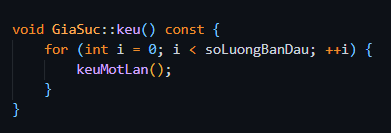
Hình 3.6: Constructor mặc định và constructor có tham số của lớp GiaSuc

* Nhap(): Nhập số lượng gia súc từ người dùng và khởi tạo các giá trị liên quan.



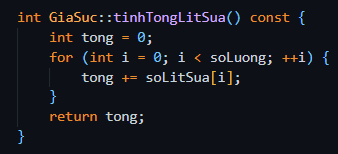
Hình 3.7: Phương thức Nhap() của lớp GiaSuc

* keu(): Phát ra tiếng kêu của đàn gia súc với số tiếng tương ứng với số lượng mỗi loài gia súc ban đầu.



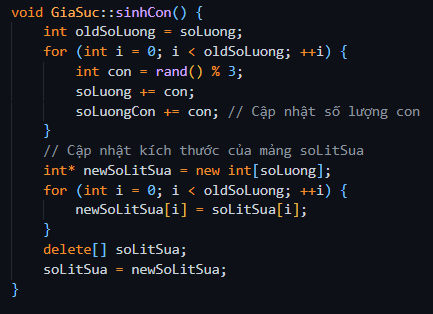
Hình 3.8: Phương thức keu() của lớp GiaSuc

* keuMotLan(): Phương thức thuần ảo, định nghĩa tiếng kêu một lần của gia súc.
* tinhTongLitSua(): Tính tổng số lít sữa mà gia súc có thể cho.



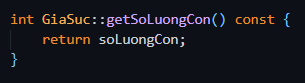
Hình 3.9: Phương thức tinhTongLitSua() của lớp GiaSuc

* sinhCon(): Sinh con và cập nhật số lượng gia súc cũng như mảng soLitSua.



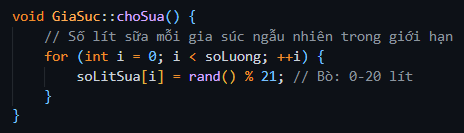
Hình 3.10: Phương thức sinhCon() của lớp GiaSuc

* getSoLuongCon(): Trả về tổng số lượng con mà gia súc đã sinh.



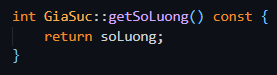
Hình 3.11: Phương thức getSoLuongCon() của lớp GiaSuc

* choSua(): Cập nhật số lít sữa mỗi gia súc có thể cho. Ở đây ta chọn số lít sữa của mỗi con Bò có thể cho, các gia súc khác ghi đè lên phương thức này để thay đổi số lít sữa mà chúng có thể cho.



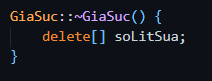
Hình 3.12: Phương thức choSua() của lớp GiaSuc

* getSoLuong(): Trả về số lượng gia súc hiện có.



Hình 3.13: Phương thức getSoLuong() của lớp GiaSuc

* ~GiaSuc(): Hàm hủy, giải phóng bộ nhớ đã cấp phát cho mảng soLitSua.



Hình 3.14: Hàm hủy của lớp GiaSuc

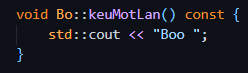
#### Lớp Bo (kế thừa từ GiaSuc)

* Thuộc tính: Không có thuộc tính riêng
* Phương thức:
* Bo() và Bo(int sl): Các hàm tạo, gọi hàm tạo của lớp cơ sở GiaSuc.



Hình 3.15: Constructor mặc định và constructor có tham số của lớp Bo

* keuMotLan(): Định nghĩa tiếng kêu “Boo” cho lớp Bo.



Hình 3.16: Phương thức keuMotLan() của lớp Bo

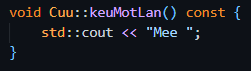
#### Lớp Cuu (kế thừa từ GiaSuc)

* Thuộc tính: Không có thuộc tính riêng
* Phương thức:
* Cuu() và Cuu(int sl): Các hàm tạo, gọi hàm tạo của lớp cơ sở GiaSuc.



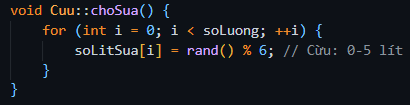
Hình 3.17: Constructor mặc định và constructor có tham số của lớp Cuu

* keuMotLan(): Định nghĩa tiếng kêu “Mee” cho lớp Cuu.



Hình 3.18: Phương thức keuMotLan() của lớp Cuu

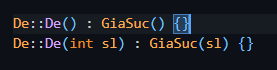
* choSua(): Ghi đè phương thức choSua của lớp cơ sở để cập nhật số lít sữa mỗi cừu có thể cho (0-5 lít).



Hình 3.19: Phương thức choSua() của lớp Cuu

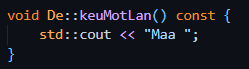
#### Lớp De (kế thừa từ GiaSuc)

* Thuộc tính: Không có thuộc tính riêng
* Phương thức:
* De() và De(int sl): Các hàm tạo, gọi hàm tạo của lớp cơ sở GiaSuc.



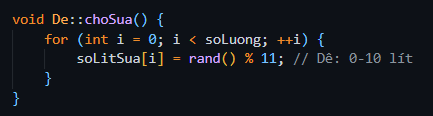
Hình 3.20: Constructor mặc định và constructor có tham số của lớp De

* keuMotLan(): Định nghĩa tiếng kêu “Maa” cho lớp De.



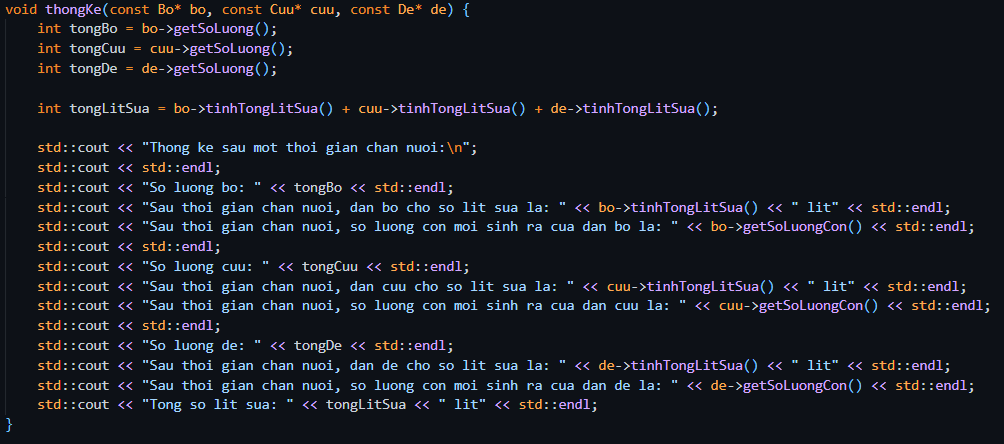
Hình 3.21: Phương thức keuMotLan() của lớp De

* choSua(): Ghi đè phương thức choSua của lớp cơ sở để cập nhật số lít sữa mỗi dê có thể cho (0-10 lít).



Hình 3.22: Phương thức choSua() của lớp De

* Để đưa ra thống kê các thông tin người chủ mong muốn sau một lứa sinh và một lượt cho sữa của tất cả gia súc. Ta xây dựng hàm thống kê như sau, hàm thống kê sẽ in ra số lượng gia súc, số lượng lít sữa của mỗi đàn thu được sau một thời gian chăn nuôi và số lượng con mới sinh ra của mỗi đàn gia súc và số tổng lít sữa thu được từ các đàn gia súc.



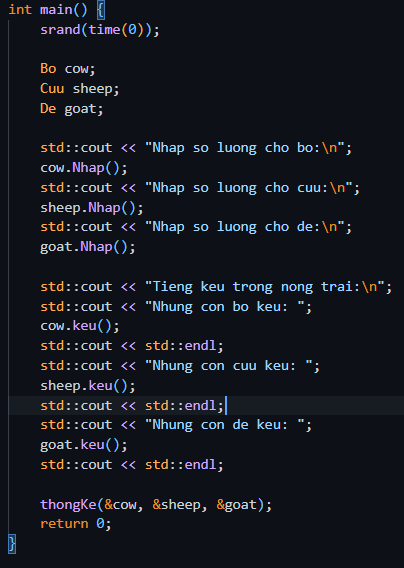
Hình 3.23: Hàm thống kê các thông tin người chủ mong muốn

* Trong phương thức sinhCon(), rand() % 3 tạo ra một số ngẫu nhiên từ 0 đến 2, mô phỏng số con mà mỗi cá thể gia súc có thể sinh ra.
* Trong phương thức choSua(), rand() % 21, rand() % 6, và rand() % 11 tạo ra một số ngẫu nhiên trong khoảng giới hạn tương ứng, mô phỏng lượng sữa mà mỗi cá thể gia súc có thể cho.

### Xác định input, output

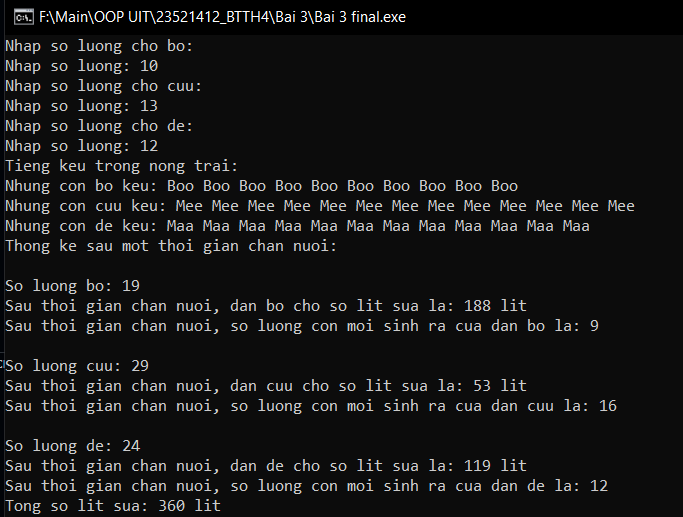
* Input: Số lượng con của mỗi loài gia súc.
* Output: Tiếng kêu trong nông trại, thống kê theo ý chủ nông trai mong muốn.

### Hàm main của chương trình



Hình 3.24: Hàm main của chương trình bài tập 3

### Kết quả chạy của chương trình



Hình 3.25: Kết quả chạy của chương trình bài tập 3

## Mã nguồn:

<https://drive.google.com/drive/u/1/folders/1NENle9NyjDaE-imYeHBNjGKUUR73wH5H>

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | 28tech, [C++]. Stringstream Trong C++ Và Ứng Dụng, Địa chỉ: https://blog.28tech.com.vn/c-stringstream-trong-c-va-ung-dung: [Truy cập lần cuối: 31/03/2024]. |
| [2] | Howkteam, Phát sinh số ngẫu nhiên trong C++ (Random number generation), Địa chỉ: https://howkteam.vn/course/khoa-hoc-lap-trinh-c-can-ban/phat-sinh-so-ngau-nhien-trong-c-random-number-generation-1376: [Truy cập lần cuối: 20/05/2024]. |