

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 Quý

Sinh viên thực hiện: Trần Nam Phong - 23521172

๙๙๓ Tp. Hồ Chí Minh, 05/2024 ๙๙๓

[illegible]

Người nhận xét
(Ký tên và ghi rõ họ tên)

MỤC LỤC

1.	Quản lý thông tin nhân viên	6
1.1.	Class Diagram của bài tập 1	6
1.2.	Thực hiện xây dựng lớp NhanVien trong file <i>NhanVien.h</i>	6
1.3.	Các lớp con của class NhanVien được xây dựng trong file <i>NhanVien.cpp</i>	8
1.4.	Gọi các phương thức trong hàm main(), đặt trong file <i>main.cpp</i>	9
1.5.	Kết quả khi chạy chương trình.....	10
2.	Tính doanh thu phòng.....	10
2.1.	Class Diagram của bài tập 2	11
2.2.	Thực hiện xây dựng lớp, khai báo các thuộc tính, phương thức của lớp các lớp trong file <i>Phong.h</i>	11
2.3.	Các phương thức của các class được xây dựng trong file <i>Phong.cpp</i>	12
2.4.	Gọi các phương thức trong hàm main(), đặt trong file <i>main.cpp</i>	15
2.5.	Kết quả khi chạy chương trình.....	16
3.	Quản lý nông trại	16
3.1.	Class diagram của bài tập 3	17
3.2.	Thực hiện xây dựng lớp, khai báo các thuộc tính, phương thức của các lớp trong file <i>GiaSuc.h</i>	17
3.3.	Các phương thức của các lớp được xây dựng trong file <i>GiaSuc.cpp</i>	18
3.4.	Gọi các phương thức trong hàm main(), đặt trong file <i>main.cpp</i>	20
3.5.	Kết quả khi chạy chương trình.....	22

DANH MỤC BẢNG

Bảng 1. Thực hiện xây dựng lớp NhanVien.....	7
Bảng 2. Các lớp con của class NhanVien được xây dựng trong file NhanVien.cpp	8
Bảng 3. Gọi các phương thức của class NhanVien trong hàm main()	9
Bảng 4. Xây dựng lớp, khai báo các thuộc tính, phương thức của các lớp trong file Phong.h	11
Bảng 5. Phương thức của các class được xây dựng trong file Phong.cpp.....	13
Bảng 6. Gọi các phương thức của các class trong hàm main().....	15
Bảng 7. Xây dựng lớp, khai báo các thuộc tính, phương thức của các lớp trong file Giasuc.h	17
Bảng 8. Các phương thức của các lớp được xây dựng trong file GiaSuc.cpp.....	19
Bảng 9. Gọi các phương thức trong hàm main()	20

DANH MỤC HÌNH ẢNH

Hình 1. Class Diagram của bài tập 1	6
Hình 2. Kết quả khi chạy chương trình Bài tập 1	10
Hình 3. Class Diagram của bài tập 2	11
Hình 4. Kết quả khi chạy chương trình Bài tập 2.....	16
Hình 5. Class diagram của bài tập 3	17
Hình 6. Kết quả khi chạy chương trình Bài tập 3.....	22

NỘI DUNG BÀI LÀM

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:

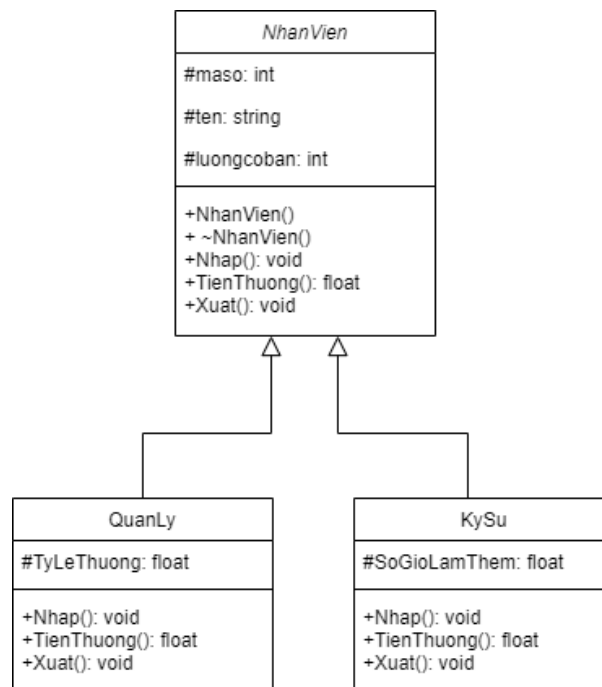
- Nhân viên: Mã số nhân viên, Tên, Lương cơ bản.
- Quản lý: Thừa kế từ Nhân viên với thêm thuộc tính là Tỷ lệ thưởng. Phương thức `TienThuong()` tính toán tiền thưởng dựa trên tỷ lệ thưởng và lương cơ bản.
- Kỹ sư: Thừa kế từ Nhân viên với thêm thuộc tính là Số giờ làm thêm. Phương thức `TienThuong()` 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.

1.1. Class Diagram của bài tập 1



Hình 1. Class Diagram của bài tập 1

1.2. Thực hiện xây dựng lớp `NhanVien` trong file `NhanVien.h`

- Lớp `NhanVien`:

IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

- Nhap(): Nhập thông tin nhân viên (mã số, tên, lương cơ bản).
- TienThuong(): Trả về 0 (nhân viên cơ sở không có tiền thưởng).
- Xuat(): Xuất thông tin nhân viên.

Bảng 1. Thực hiện xây dựng lớp NhanVien

```
#pragma once
#include<bits/stdc++.h>
using namespace std;

class NhanVien {
protected:
    int maso;
    string ten;
    int luongcoban;
public:
    NhanVien() {}
    ~NhanVien() {}
    void Nhap() {
        cout << "Nhap ma so: ";
        cin >> maso;
        cin.ignore();
        cout << "Nhap ten: ";
        getline(cin, ten);
        cout << "Nhap lương cơ bản: ";
        cin >> luongcoban;
    }

    float TienThuong() {
        return 0;
    }

    void Xuat() {
        cout << "Mã số: " << maso << endl;
        cout << "Tên: " << ten << endl;
        cout << "Lương cơ bản: " << luongcoban << endl;
    }
};
```

1.3. Các lớp con của class NhanVien được xây dựng trong file *NhanVien.cpp*

- Lớp KySu:
 - Nhap(): Gọi Nhap() từ lớp cha NhanVien và nhập số giờ làm thêm.
 - TienThuong(): Tính và trả về tiền thưởng dựa trên số giờ làm thêm.
 - Xuat(): Gọi Xuat() từ lớp cha và xuất tiền thưởng.
- Lớp QuanLy:
 - Nhap(): Gọi Nhap() từ lớp cha NhanVien và nhập tỷ lệ thưởng.
 - TienThuong(): Tính và trả về tiền thưởng dựa trên tỷ lệ thưởng và lương cơ bản.
 - Xuat(): Gọi Xuat() từ lớp cha và xuất tiền thưởng.

Bảng 2. Các lớp con của class NhanVien được xây dựng trong file NhanVien.cpp

```
#include "NhanVien.h"
#include<bits/stdc++.h>
using namespace std;

class KySu : public NhanVien {
protected:
    float SoGioLamThem;
public:
    void Nhap() {
        NhanVien::Nhap();
        cout << "Nhap so gio lam them: ";
        cin >> SoGioLamThem;
    }
    float TienThuong() {
        return SoGioLamThem * 100000;
    }
    void Xuat() {
        NhanVien::Xuat();
        cout << "Tien thuong: " << TienThuong() << endl;
        cout << endl;
    }
};

class QuanLy : public NhanVien {
protected:
    float TyLeThuong;
public:
    void Nhap() {
```



```

        NhanVien::Nhap();
        cout << "Nhap ty le thuong: ";
        cin >> TyLeThuong;
    }

    float TienThuong() {
        return TyLeThuong * luongcoban;
    }
    void Xuat() {
        NhanVien::Xuat();
        cout << "Tien thuong: " << TienThuong() << endl;
        cout << endl;
    }
};

```

1.4. Gọi các phương thức trong hàm main(), đặt trong file *main.cpp**Bảng 3. Gọi các phương thức của class NhanVien trong hàm main()*

```

#include "NhanVien.h"
#include "NhanVien.cpp"
#include<bits/stdc++.h>
using namespace std;

int main() {
    QuanLy a;
    a.Nhap();

    KySu b;
    b.Nhap();

    cout << "Thong tin cua Quan Ly: \n";
    a.Xuat();
    cout << "Thong tin cua Ky Su: \n";
    b.Xuat();
    return 0;
}

```

1.5. Kết quả khi chạy chương trình

```
Nhap ma so: 1175
Nhap ten: Tran The Phong
Nhap luong co ban: 500000
Nhap ty le thuong: 1.5
Nhap ma so: 2352
Nhap ten: Tran Nam Phong
Nhap luong co ban: 400000
Nhap so gio lam them: 2
Thong tin cua Quan Ly:
Ma so: 1175
Ten: Tran The Phong
Luong co ban: 500000
Tien thuong: 750000

Thong tin cua Ky Su:
Ma so: 2352
Ten: Tran Nam Phong
Luong co ban: 400000
Tien thuong: 200000
```

Hình 2. Kết quả khi chạy chương trình Bài tập 1

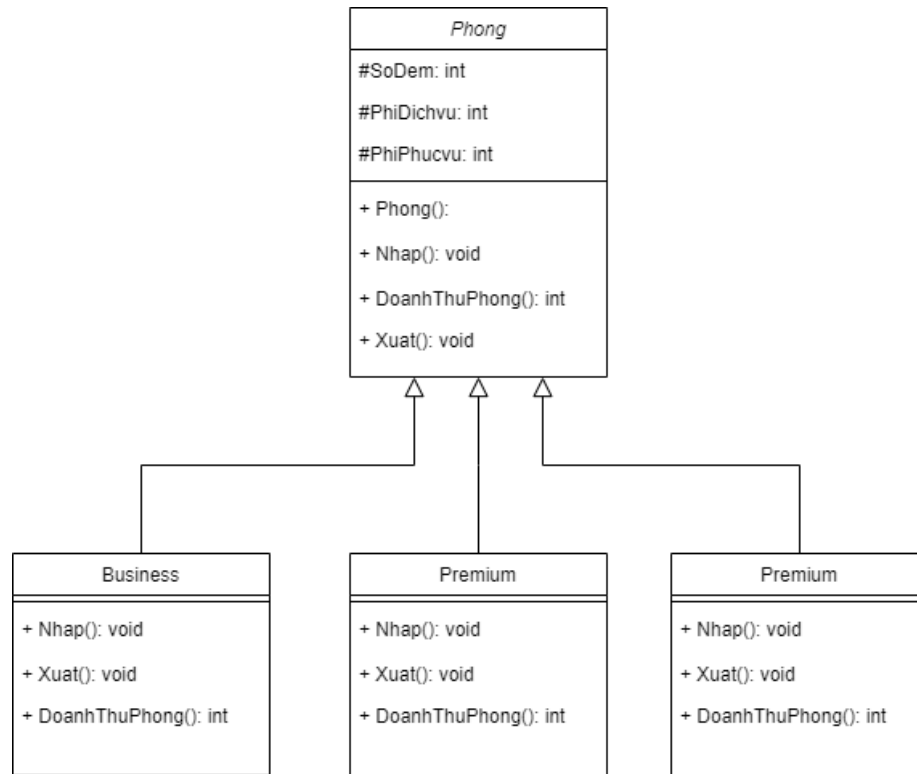
2. Tính doanh thu phòng

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).

2.1. Class Diagram của bài tập 2



Hình 3. Class Diagram của bài tập 2

2.2. Thực hiện xây dựng lớp, khai báo các thuộc tính, phương thức của lớp các lớp trong file *Phong.h*

Bảng 4. Xây dựng lớp, khai báo các thuộc tính, phương thức của các lớp trong file *Phong.h*

```
#pragma once
#include <iostream>
using namespace std;
class Phong
{
protected:
    int SoDem;
    int PhiDichvu, PhiPhucvu;
public:
    Phong();
    int getSoDem()
    {
        return SoDem;
    }
    ~Phong() {} ;
    void Nhap();
    int DoanhThuPhong();
}
```

```

        void Xuat();
};
class Business : public Phong
{
public:
    void Nhap();
    int DoanhThuPhong();
    void Xuat();
};
class Premium : public Phong
{
public:
    void Nhap();
    int DoanhThuPhong();
    void Xuat();

};
class Deluxe : public Phong
{
public:
    void Nhap();
    int DoanhThuPhong();
    void Xuat();

};

```

2.3. Các phương thức của các class được xây dựng trong file *Phong.cpp*

- Lớp Phong:
 - Phương thức Nhap(): Dùng để nhập dữ liệu đầu vào.
 - Phương thức DoanhThuPhong(): Tính toán doanh thu của phòng dựa trên đầu vào.
 - Phương thức Xuat(): Hiển thị kết quả doanh thu ra màn hình.
- Lớp Business:
 - Phương thức Nhap(): Gọi Nhap() của lớp cha để nhập số đêm.
 - Phương thức DoanhThuPhong(): Tính doanh thu phòng Business = số đêm * 300.000đ.
 - Phương thức Xuat(): In dòng chữ "Doanh thu phòng loại Business: " và kết quả từ DoanhThuPhong().
- Lớp Premium:
 - Phương thức Nhap(): Gọi Nhap() lớp cha và nhập thêm phí dịch vụ.
 - Phương thức DoanhThuPhong(): Tính doanh thu phòng Premium = số đêm * 500.000đ + phí dịch vụ.

IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

- Phương thức `Xuat()`: In "Doanh thu phòng Premium: " và kết quả `DoanhThuPhong()`.
- Lớp `Deluxe`:
 - Phương thức `Nhap()`: Gọi `Nhap()` lớp cha và nhập thêm phí dịch vụ, phí phục vụ.
 - Phương thức `DoanhThuPhong()`: Tính doanh thu phòng `Deluxe` = số đêm * 750.000đ + phí dịch vụ + phí phục vụ.
 - Phương thức `Xuat()`: In "Doanh thu phòng Deluxe: " và kết quả `DoanhThuPhong()`.

Bảng 5. Phương thức của các class được xây dựng trong file `Phong.cpp`

```
#include<iostream>
#include "Phong.h"
using namespace std;

Phong::Phong() {

}

void Phong::Nhap()
{
    cout << "Nhap so dem:";
    cin >> SoDem;
}

int Phong::DoanhThuPhong()
{
    SoDem = getSoDem();
    return 0;
}

void Phong::Xuat()
{
    cout << DoanhThuPhong() << endl;
}

void Business::Nhap()
{
    Phong::Nhap();
}

int Business::DoanhThuPhong()
{
    SoDem = getSoDem();
    return SoDem * 300000;
}
```

```
void Business::Xuat()
{
    cout << "Danh thu phong loai Business: ";
    cout << DoanhThuPhong() << endl;
}
void Premium::Nhap()
{
    Phong::Nhap();
    cout << "Nhap phi dich vu:";
    cin >> PhiDichvu;
}
int Premium::DoanhThuPhong()
{
    return SoDem * 500000 + PhiDichvu;
}
void Premium::Xuat()
{
    cout << "Danh thu phong loai Premium: ";
    cout << Premium::DoanhThuPhong() << endl;
}
void Deluxe::Nhap()
{
    Phong::Nhap();
    cout << "Nhap phi dich vu:";
    cin >> PhiDichvu;
    cout << "Nhap phi phuc vu:";
    cin>> PhiPhucvu;
}
int Deluxe::DoanhThuPhong()
{
    return SoDem * 750000 + PhiDichvu + PhiPhucvu;
}
void Deluxe::Xuat()
{
    cout << "Danh thu phong loai Deluxe: ";
    cout << Deluxe::DoanhThuPhong() << endl;
}
```

2.4. Gọi các phương thức trong hàm main(), đặt trong file *main.cpp**Bảng 6. Gọi các phương thức của các class trong hàm main()*

```
#include<iostream>
#include "Phong.h"
using namespace std;

int main()
{
    Deluxe a, b;
    Premium c, d;
    Business e;
    double D,P,B;
    int busi = 0, pre = 0, del = 0;
    e.Nhap();
    busi = e.DoanhThuPhong();
    e.Xuat();
    c.Nhap();
    pre = c.DoanhThuPhong();
    c.Xuat();
    d.Nhap();
    pre += d.DoanhThuPhong();
    d.Xuat();
    b.Nhap();
    del = b.DoanhThuPhong();
    b.Xuat();
    a.Nhap();
    del += a.DoanhThuPhong();
    a.Xuat();
    cout << "Doanh thu cao nhat la loai phong: ";
    if (del >= pre && del >= busi)
    {
        cout << "Deluxe";
    }
    else if (pre >= del && pre >= busi)
    {
        cout << "Premium";
    }
    else
    {

```

```

        cout << "Business";

    }

}

```

2.5. Kết quả khi chạy chương trình

```

Nhap so dem:3
Danh thu phong loai Business: 900000
Nhap so dem:4
Nhap phi dich vu:7
Danh thu phong loai Premium: 2000007
Nhap so dem:5
Nhap phi dich vu:7
Danh thu phong loai Premium: 2500007
Nhap so dem:10
Nhap phi dich vu:6
Nhap phi phuc vu:8
Danh thu phong loai Deluxe: 7500014
Nhap so dem:3
Nhap phi dich vu:6
Nhap phi phuc vu:8
Danh thu phong loai Deluxe: 2250014
Doanh thu cao nhat la loai phong: Deluxe

```

*Hình 4. Kết quả khi chạy chương trình Bài tập 2***3. Quản lý nông trại**

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.

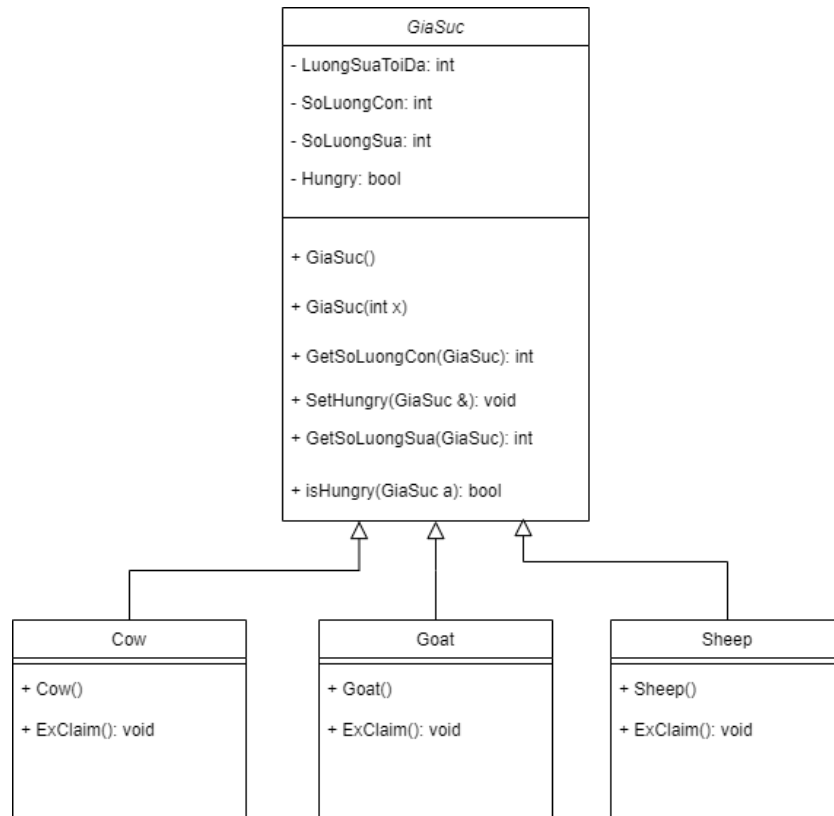
Áp dụng kế thừa, xây dựng chương trình cho phép người chủ nông trại nhập vào số lượng gia súc ban đầu ở mỗi loại.

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.

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.
- 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

3.1. Class diagram của bài tập 3



Hình 5. Class diagram của bài tập 3

3.2. Thực hiện xây dựng lớp, khai báo các thuộc tính, phương thức của các lớp trong file GiaSuc.h

Bảng 7. Xây dựng lớp, khai báo các thuộc tính, phương thức của các lớp trong file Giasuc.h

```

#include <iostream>
using namespace std;

typedef class GiaSuc
{
private:
    int LuongSuaToiDa;
    int SoLuongCon;
    int SoLuongSua;
    bool Hungry;

public:
    GiaSuc() {}
    GiaSuc(int x);
    // void ExClaim();
    friend int GetSoLuongCon (GiaSuc);

```

```
friend void SetHungry(GiaSuc &);  
friend int GetSoLuongSua(GiaSuc);  
friend bool isHungry(GiaSuc a);  
} GS;  
  
class Cow : public GS  
{  
private:  
public:  
    Cow() : GS(20) {};  
    void ExClaim();  
};  
  
class Goat : public GS  
{  
private:  
public:  
    Goat() : GS(10) {}  
    void ExClaim();  
};  
  
class Sheep : public GS  
{  
private:  
public:  
    Sheep() : GS(5) {}  
    void ExClaim();  
};
```

3.3. Các phương thức của các lớp được xây dựng trong file GiaSuc.cpp

- Hướng giải quyết các phương thức:
 - GiaSuc(int x): Khởi tạo đối tượng GiaSuc với lượng sữa tối đa x, số lượng con và số lượng sữa ngẫu nhiên.
 - randomNumber(int n): Trả về một số ngẫu nhiên trong khoảng từ 0 đến n.
 - SetHungry(GiaSuc &a): Đặt trạng thái đói của đối tượng a thành true.
 - isHungry(GiaSuc a): Kiểm tra xem đối tượng a có đói hay không.
 - GetSoLuongCon(GiaSuc a): Trả về số lượng con của đối tượng a.
 - GetSoLuongSua(GiaSuc a): Trả về số lượng sữa của đối tượng a.
 - Cow::ExClaim(): Xuất ra tiếng kêu của bò.
 - Sheep::ExClaim(): Xuất ra tiếng kêu của cừu.

IT002 – LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

- Goat::ExClaim(): Xuất ra tiếng kêu của dê.

Bảng 8. Các phương thức của các lớp được xây dựng trong file GiaSuc.cpp

```
#include <iostream>
#include <random>
#include <chrono>
using namespace std;
#include "GiaSuc.h"
int randomNumber(int n)
{
    auto seed =
std::chrono::system_clock::now().time_since_epoch().count(); // Lấy
thời gian hệ thống làm seed
    std::mt19937 gen(seed);
// Máy phát sinh số ngẫu nhiên với seed là thời gian hệ thống
    std::uniform_int_distribution<> dis(0, n);
// Phân phối đều từ 0 đến n

    return dis(gen); // Trả về số nguyên ngẫu nhiên từ 0 đến n
}
GS::GiaSuc(int x)
{
    LuongSuaToiDa = x;
    SoLuongCon = randomNumber(2);
    SoLuongSua = randomNumber(x);
}

void SetHungry(GiaSuc &a)
{
    a.Hungry = 1;
}

bool isHungry(GiaSuc a)
{
    return a.Hungry;
}

int GetSoLuongCon(GiaSuc a)
{
    return a.SoLuongCon;
}
```

```
int GetSoLuongSua(GiaSuc a)
{
    return a.SoLuongSua;
}

void Cow::ExClaim()
{
    cout << "Um BOoooo'" << endl;
}

void Sheep::ExClaim()
{
    cout << "Beeee" << endl;
}

void Goat::ExClaim()
{
    cout << "Be Be Be" << endl;
}
```

3.4. Gọi các phương thức trong hàm main(), đặt trong file *main.cpp*

Bảng 9. Gọi các phương thức trong hàm main()

```
#include <iostream>
#include "GiaSuc.cpp"

int main()
{
    int l, m, n;
    cout << "Nhap So luong bo: ";
    cin >> l;
    Cow bo[l];
    for (int i = 0; i < l; i++)
    {
        SetHungry(bo[i]);
    }
    cout << "Nhap So luong cuu: ";
    cin >> m;
    Sheep cuu[m];
    for (int i = 0; i < m; i++)
        SetHungry(cuu[i]);
}
```

```
cout << "Nhap So luong de: ";
cin >> n;
Goat de[n];
for (int i = 0; i < n; i++)
    SetHungry(de[i]);

cout << "Tat ca tieng keu nghe duoc la: " << endl;
if (isHungry(bo[0]))
    bo[0].ExClaim();
if (isHungry(cuu[0]))
    cuu[0].ExClaim();
if (isHungry(de[0]))
    de[0].ExClaim();
cout << endl;

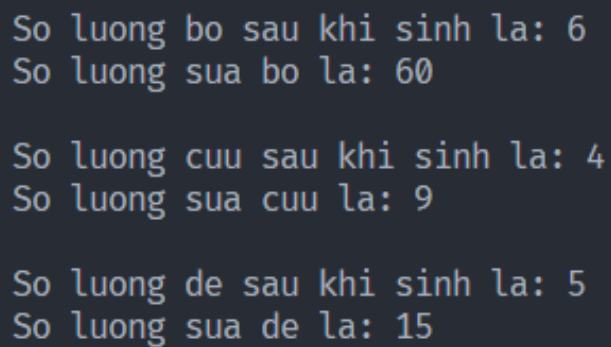
int TongCon = 0;
int TongSua = 0;
for (int i = 0; i < l; i++)
{
    TongCon += GetSoLuongCon(bo[i]);
    TongSua += GetSoLuongSua(bo[i]);
}
cout << "So luong bo sau khi sinh la: " << TongCon + 1 << endl;
cout << "So luong sua bo la: " << TongSua << endl;
cout << endl;

TongCon = 0;
TongSua = 0;
for (int i = 0; i < m; i++)
{
    TongCon += GetSoLuongCon(cuu[i]);
    TongSua += GetSoLuongSua(cuu[i]);
}
cout << "So luong cuu sau khi sinh la: " << TongCon + m << endl;
cout << "So luong sua cuu la: " << TongSua << endl;
cout << endl;

TongCon = 0;
TongSua = 0;
for (int i = 0; i < n; i++)
```

```
{  
    TongCon += GetSoLuongCon(de[i]);  
    TongSua += GetSoLuongSua(de[i]);  
}  
cout << "So luong de sau khi sinh la: " << TongCon + n << endl;  
cout << "So luong sua de la: " << TongSua << endl;  
}
```

3.5. Kết quả khi chạy chương trình



```
So luong bo sau khi sinh la: 6  
So luong sua bo la: 60  
  
So luong cuu sau khi sinh la: 4  
So luong sua cuu la: 9  
  
So luong de sau khi sinh la: 5  
So luong sua de la: 15
```

Hình 6. Kết quả khi chạy chương trình Bài tập 3

[FILE CODE](#)