

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

ĐOÀN VIỆT KHẢI - 23520673

MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
BÀI TẬP THỰC HÀNH 4

**OBJECT-ORIENTED PROGRAMMING
EXERCISE 4**

CỬ NHÂN NGÀNH AN TOÀN THÔNG TIN

GIẢNG VIÊN HƯỚNG DẪN
CN. NGUYỄN NGỌC QUÍ

TP. HỒ CHÍ MINH, NĂM 2024

Mục lục

| | |
|---------------------------------------|-----------|
| Mục lục | i |
| 1. Bài tập 1 | 1 |
| 1.1 Class diagram | 2 |
| 1.2 Mô tả đầu vào và đầu ra | 3 |
| 1.3 Ý tưởng | 3 |
| 1.4 Thực thi code | 6 |
| 1.5 Kiểm thử | 13 |
| 2. Bài tập 2 | 15 |
| 2.1 Class diagram | 15 |
| 2.2 Mô tả đầu vào và đầu ra | 16 |
| 2.3 Ý tưởng | 16 |
| 2.4 Thực thi code | 18 |
| 2.5 Kiểm thử | 25 |
| 3. Bài tập 3 | 27 |
| 3.1 Class diagram | 28 |
| 3.2 Mô tả đầu vào và đầu ra | 29 |
| 3.3 Ý tưởng | 29 |
| 3.4 Thực thi code | 35 |
| 3.5 Kiểm thử | 47 |

1. Bài tập 1

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:

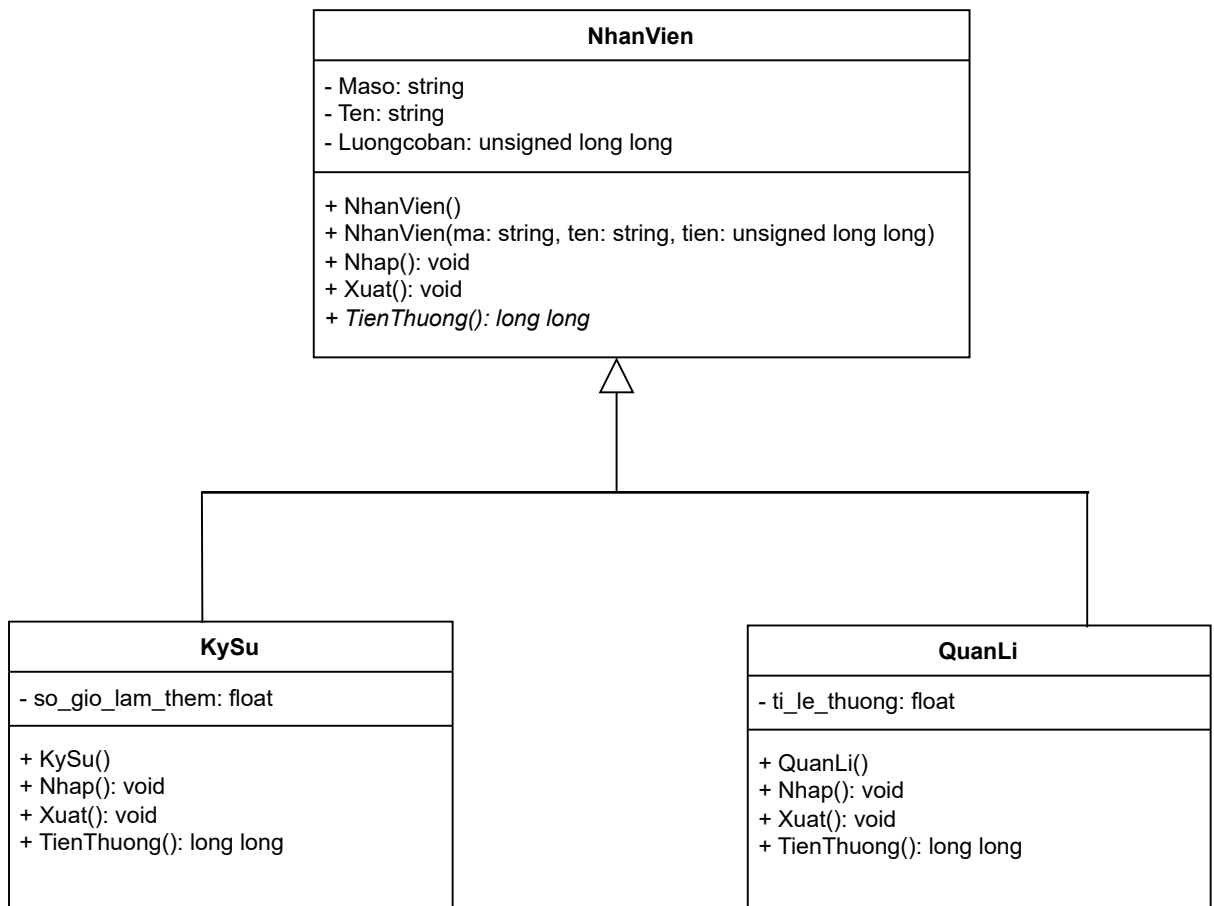
- Cho phép người dùng nhập thông tin của một quản lý và một kỹ sư.
- Hiển thị thông tin chi tiết của cả hai nhân viên, bao gồm mã số, tên, lương cơ bản và tiền thưởng tính được.
- Sử dụng tính kế thừa để thực hiện việc tính toán tiền thưởng dựa trên loại nhân viên.
- Sử dụng hàm Xuất() để hiển thị thông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

1.1 Class diagram



Hình 1.1: Class diagram bài 1

1.2 Mô tả đầu vào và đầu ra

- **Đầu vào:** Nhập vào các thông tin cơ bản của một quản lí và một kĩ sư, bao gồm *mã số nhân viên, tên và lương cơ bản*.
- **Đầu ra:**
 - Thực hiện việc tính tiền thưởng theo phân loại.
 - Xuất thông tin của một nhân viên, bao gồm: *mã số nhân viên, tên, lương cơ bản và tiền thưởng tính được*.

1.3 Ý tưởng

1. Xây dựng class Nhanvien làm class cha.

- Private: Mã số nhân viên, tên và lương cơ bản.
- Public:
 - Phương thức nhập: Nhập các thông tin như trên

```
void NhanVien :: Nhap(){
    cout << "Nhap ma so nhan vien: ";
    cin >> maso;
    cout << "Nhap ten nhan vien: ";
    cin >> ten;
    cout << "Nhap lương cơ bản cho nhân viên: ";
    cin >> lương_cơ_bản;
}
```

- Phương thức tính tiền thưởng: Vì hai class con kế thừa class Nhanvien cũng sử dụng phương thức này nên ta khai báo phương thức ảo để thuận tiện cho việc xuất dữ liệu. Do việc tính tiền thưởng có dùng lương cơ bản, nên ở class Nhanvien, chỉ đơn giản là ta trả về lương cơ bản.

```
long long NhanVien :: TienThuong(){
    return lương_cơ_bản;
};
```

- Phương thức xuất: Xuất các thông tin như đề bài.

```
void NhanVien :: Xuat(){
    cout << "\tMa so nhan vien: " << maso << "\n";
    cout << "\tTen nhan vien: " << ten << "\n";
    cout << "\tLuong co ban: " << luong_co_ban << "\n";
    cout << "\tTien Thuong: " << TienThuong() << "\n";
}
```

2. Xây dựng lớp Quanli kế thừa lớp Nhanvien.

- Private: Bổ sung thêm thuộc tính tỉ lệ thưởng.
- Public:
 - Phương thức nhập: Gọi phương thức nhập của lớp Nhanvien, sau đó nhập thêm tỉ lệ thưởng.

```
void Quanli :: Nhap(){
    cout << "Nhập thông tin cho quan li: " << "\n";
    NhanVien :: Nhap();
    cout << "Nhập tỉ lệ thưởng: ";
    cin >> ti_le_thuong;
}
```

- Phương thức tính tiền thưởng: Cho phép ghi đè (override) để có thể xuất dữ liệu.

$\text{Tiền thưởng của quản lí} = \text{Lương cơ bản} \times \text{tỉ lệ thưởng}$

Mà lương cơ bản đã được trả về trong phương thức tiền thưởng của lớp cha, nên ta chỉ cần gọi lại là được.

```
long long Quanli :: TienThuong(){
    return NhanVien :: TienThuong()*ti_le_thuong;
}
```

- Phương thức xuất: Gọi lại phương thức xuất của lớp cha.

```
void Quanli :: Xuat(){
    cout << "Quan li: " << "\n";
    NhanVien :: Xuat();
}
```

3. Xây dựng class Kysu kế thừa từ lớp NhanVien.

- Private: Bổ sung thêm thuộc tính số giờ làm thêm.
- Public:
 - Phương thức nhập: Gọi phương thức nhập của lớp Nhanvien, sau đó nhập thêm tỉ lệ thưởng.

```
void Kysu :: Nhap(){
    cout << "Nhập thông tin cho kỹ sư: \n";
    NhanVien :: Nhap();
    cout << "Nhập số giờ làm thêm: ";
    cin >> so_gio_lam_them;
}
```

- Phương thức tính tiền thưởng: Cho phép ghi đè (override) để có thể xuất dữ liệu.

$\text{Tiền thưởng của kỹ sư} = 100000 \times \text{số giờ làm thêm}$

```
long long Kysu :: TienThuong(){
    return 100000*so_gio_lam_them;
}
```

- Phương thức xuất: Gọi lại phương thức xuất của lớp cha.

```
void Kysu :: Xuat(){
    cout << "\nKỹ sư: " << "\n";
    NhanVien :: Xuat();
}
```

1.4 Thực thi code

Link code: [Bài 1](#)

NhanVien.h

```
#ifndef NhanVien_H
#define NhanVien_H
#include <iostream>
using namespace std;

class NhanVien{
    private:
        string maso, ten;
        unsigned long long luong_co_ban;
    public:
        NhanVien();
        void Nhap();
        void Xuat();
        virtual long long TienThuong();
};

#endif
```

NhanVien.cpp

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

/* Constructor mặc định
    Đầu vào: Không có tham số đầu vào.
    Đầu ra: Khởi tạo một đối tượng NhanVien với giá trị mặc định:
        - maso: chuỗi rỗng
        - ten: chuỗi rỗng
        - luong_co_ban: 0.0
```


1.4. Thực thi code

```
Hoạt động: Gán giá trị mặc định cho các thành viên của lớp.
*/
NhanVien :: NhanVien() : maso(""), ten(""), luong_co_ban(0.0){}

/* Phương thức Nhập()
Đầu vào: Nhập thông tin từ người dùng gồm:
    - Mã số nhân viên (maso)
    - Tên nhân viên (ten)
    - Lương cơ bản (luong_co_ban)
Đầu ra: Không có giá trị trả về.
Hoạt động: Nhận dữ liệu từ người dùng và gán vào các thuộc tính tương
ứng của đối tượng.
*/
void NhanVien :: Nhập(){
    cout << "Nhập ma so nhan vien: ";
    cin >> maso;
    cout << "Nhập ten nhan vien: ";
    cin >> ten;
    cout << "Nhập lương cơ bản cho nhân viên: ";
    cin >> luong_co_ban;
}

/* Phương thức Xuất()
Đầu vào: Không có tham số đầu vào.
Đầu ra: Không có giá trị trả về.
Hoạt động: In ra thông tin của nhân viên, bao gồm:
    - Mã số nhân viên
    - Tên nhân viên
    - Lương cơ bản
    - Tiền thưởng (gọi phương thức TienThuong)
*/
void NhanVien :: Xuất(){
    cout << "\tMa so nhan vien: " << maso << "\n";
    cout << "\tTen nhan vien: " << ten << "\n";
    cout << "\tLương cơ bản: " << luong_co_ban << "\n";
    cout << "\tTiền Thuong: " << TienThuong() << "\n";
}
```

```
    }

    /* Phương thức TienThuong()
       Đầu vào: Không có tham số đầu vào.
       Đầu ra: Tiền thưởng của nhân viên, tính dựa trên lương cơ bản.
               Giá trị mặc định trả về là lương cơ bản.
       Hoạt động: Trả về giá trị lương cơ bản như tiền thưởng mặc định.
    */
    long long NhanVien :: TienThuong(){
        return lương_co_ban;
    };
```

QuanLi.h

```
#ifndef QuanLi_H
#define QuanLi_H
#include <iostream>
#include "NhanVien.h"
using namespace std;

class Quanli : public NhanVien{
private:
    float ti_le_thuong;
public:
    Quanli();
    void Nhap();
    void Xuat();
    long long TienThuong() override;
};

#endif
```

QuanLi.cpp

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

/* Constructor mặc định
   Đầu vào: Không có tham số đầu vào.
   Đầu ra: Khởi tạo một đối tượng Quanli với giá trị mặc định:
           - Kế thừa các giá trị mặc định của lớp NhanVien.
           - ti_le_thuong: 0.
   Hoạt động: Gọi constructor của lớp cơ sở (NhanVien) và khởi tạo
   ti_le_thuong với 0.
*/
Quanli :: Quanli() : NhanVien(), ti_le_thuong(0){}

/* Phương thức Nhập()
   Đầu vào: Nhập thông tin từ người dùng gồm:
           - Thông tin chung của nhân viên (gọi phương thức Nhập() của
           lớp NhanVien).
           - Tỷ lệ thưởng (ti_le_thuong).
   Đầu ra: Không có giá trị trả về.
   Hoạt động: Gọi phương thức Nhập() của lớp NhanVien để nhận thông tin
   cơ bản, sau đó nhập thêm giá trị cho ti_le_thuong từ người dùng.
*/
void Quanli :: Nhập(){
    cout << "Nhập thông tin cho quan li: " << "\n";
    NhanVien :: Nhập();
    cout << "Nhập ti le thuong: ";
    cin >> ti_le_thuong;
}

/* Phương thức Xuất()
   Đầu vào: Không có tham số đầu vào.
   Đầu ra: Không có giá trị trả về.
```

1.4. Thực thi code

Hoạt động: Gọi phương thức Xuat() của lớp NhanVien để in thông tin chung, hiển thị thông tin của đối tượng Quanli.

```
*/  
void Quanli :: Xuat(){  
    cout << "Quan li: " << "\n";  
    NhanVien :: Xuat();  
}  
  
/* Phương thức TienThuong()  
Đầu vào: Không có tham số đầu vào.  
Đầu ra: Tiền thưởng của quản lý, tính dựa trên:  
    - Tiền thưởng cơ bản từ lớp NhanVien.  
    - Tỷ lệ thưởng (ti_le_thuong).  
Hoạt động: Gọi phương thức TienThuong() của lớp NhanVien, nhân kết  
quả với ti_le_thuong và trả về.  
*/  
long long Quanli :: TienThuong(){  
    return NhanVien :: TienThuong()*ti_le_thuong;  
}
```

KySu.h

```
#ifndef KySu_H  
#define KySu_H  
#include <iostream>  
#include "NhanVien.h"  
using namespace std;  
  
class Kysu : public NhanVien{  
private:  
    float so_gio_lam_them;  
public:  
    Kysu();  
    void Nhap();
```

```
void Xuat();  
long double TienThuong() override;  
  
};  
#endif
```

KySu.cpp

```
#include "KySu.h"  
#include <iostream>  
#include <string>  
using namespace std;  
  
/* Constructor mặc định  
   Đầu vào: Không có tham số đầu vào.  
   Đầu ra: Khởi tạo một đối tượng Kysu với giá trị mặc định:  
           - Kế thừa các giá trị mặc định của lớp NhanVien.  
           - so_gio_lam_them: 0.  
   Hoạt động: Gọi constructor của lớp cơ sở (NhanVien) và khởi tạo  
   so_gio_lam_them là 0.  
*/  
  
Kysu :: Kysu() : NhanVien(), so_gio_lam_them(0){}  
  
/* Phương thức Nhap()  
   Đầu vào: Nhập thông tin từ người dùng gồm:  
           - Thông tin chung của nhân viên (gọi phương thức Nhap() của  
             lớp NhanVien).  
           - Số giờ làm thêm (so_gio_lam_them).  
   Đầu ra: Không có giá trị trả về.  
   Hoạt động: Gọi phương thức Nhap() của lớp NhanVien để nhận thông tin  
   cơ bản, sau đó nhập thêm giá trị cho so_gio_lam_them từ người dùng.  
*/  
  
void Kysu :: Nhap(){  
    cout << "Nhap thong tin cho ky su: \n";  
    NhanVien :: Nhap();  
    cout << "Nhap so gio lam them: ";
```

```
        cin >> so_gio_lam_them;
    }

    /* Phương thức Xuat()
       Đầu vào: Không có tham số đầu vào.
       Đầu ra: Không có giá trị trả về.
       Hoạt động: Gọi phương thức Xuat() của lớp NhanVien để in thông tin
       chung, hiển thị thông tin của đối tượng Kysu.
    */
    void Kysu :: Xuat(){
        cout << "\nKy su: " << "\n";
        NhanVien :: Xuat();
    }

    /* Phương thức TienThuong()
       Đầu vào: Không có tham số đầu vào.
       Đầu ra: Tiền thưởng của kỹ sư, tính dựa trên:
           - Số giờ làm thêm (so_gio_lam_them).
           - Hệ số thưởng cố định là 100,000 đồng/giờ.
       Hoạt động: Nhân số giờ làm thêm (so_gio_lam_them) với hệ số 100,000
       và trả về kết quả.
    */
    long long Kysu :: TienThuong(){
        return 100000*so_gio_lam_them;
    }
```

main_bai1.cpp

```
#include "QuanLi.h"
#include "KySu.h"
#include "NhanVien.h"
#include <iostream>
using namespace std;

int main(){
```

```
Quanli ql;
Kysu ks;
ql.Nhap();
ks.Nhap();

cout << "\n~~~ THONG TIN NHAN VIEN ~~~~\n";
ql.Xuat();
ks.Xuat();

}
```

1.5 Kiểm thử

- Trường hợp 1: Không có giờ làm thêm và không có tỉ lệ thưởng.

```
Nhap thong tin cho quan li:
Nhap ma so nhan vien: 0673
Nhap ten nhan vien: Khai
Nhap luong co ban cho nhan vien: 28000200
Nhap ti le thuong: 0
Nhap thong tin cho ky su:
Nhap ma so nhan vien: 1414
Nhap ten nhan vien: Thai
Nhap luong co ban cho nhan vien: 26320780
Nhap so gio lam them: 0

~~~ THONG TIN NHAN VIEN ~~~~
Quan li:
    Ma so nhan vien: 0673
    Ten nhan vien: Khai
    Luong co ban: 28000200
    Tien Thuong: 0

Ky su:
    Ma so nhan vien: 1414
    Ten nhan vien: Thai
    Luong co ban: 26320780
    Tien Thuong: 0
```

- Trường hợp 2: Giờ làm thêm và tỉ lệ thưởng là số nguyên.

```
Nhap thong tin cho quan li:
Nhap ma so nhan vien: 1011
Nhap ten nhan vien: Khai
Nhap luong co ban cho nhan vien: 10000000
Nhap ti le thuong: 5
Nhap thong tin cho ky su:
Nhap ma so nhan vien: 0403
Nhap ten nhan vien: Thai
Nhap luong co ban cho nhan vien: 12000000
Nhap so gio lam them: 10
```

```
~%~ THONG TIN NHAN VIEN ~%~
```

```
Quan li:
```

```
Ma so nhan vien: 1011
Ten nhan vien: Khai
Luong co ban: 10000000
Tien Thuong: 50000000
```

```
Ky su:
```

```
Ma so nhan vien: 0403
Ten nhan vien: Thai
Luong co ban: 12000000
Tien Thuong: 10000000
```

- Trường hợp 3: Giờ làm thêm và tỉ lệ thưởng là số thực.

```
Nhap thong tin cho quan li:
Nhap ma so nhan vien: 1111
Nhap ten nhan vien: Luc
Nhap luong co ban cho nhan vien: 16750000
Nhap ti le thuong: 2.5
Nhap thong tin cho ky su:
Nhap ma so nhan vien: 0901
Nhap ten nhan vien: Duyen
Nhap luong co ban cho nhan vien: 8200000
Nhap so gio lam them: 3.75
```

```
~%~ THONG TIN NHAN VIEN ~%~
```

```
Quan li:
```

```
Ma so nhan vien: 1111
Ten nhan vien: Luc
Luong co ban: 16750000
Tien Thuong: 41875000
```

```
Ky su:
```

```
Ma so nhan vien: 0901
Ten nhan vien: Duyen
Luong co ban: 8200000
Tien Thuong: 375000
```

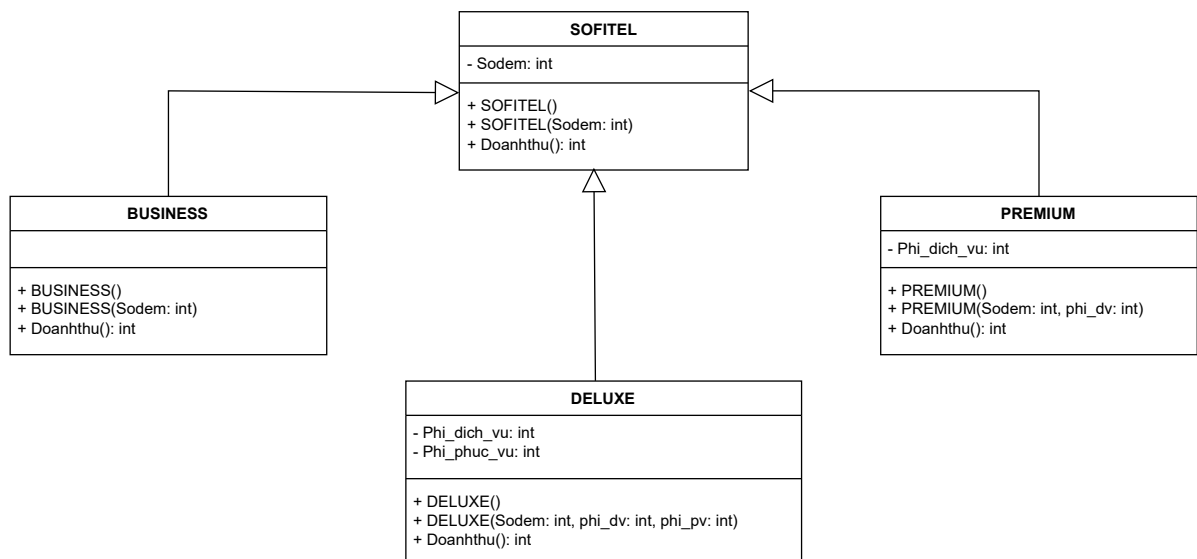

2. Bài tập 2

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).

2.1 Class diagram



Hình 2.2: Class diagram bài 2

2.2 Mô tả đầu vào và đầu ra

- **Đầu vào:** Truyền vào *các đối số* tương ứng với từng *loại phòng* ở các constructor có tham số như đề bài yêu cầu.
- **Đầu ra:** Cho biết loại phòng nào có doanh thu lớn nhất.

2.3 Ý tưởng

1. Xây dựng class SOFITEL làm class cha:

- Private: Số đêm.
- Public: Constructor mặc định, constructor có tham số và phương thức tính doanh thu. Vì phần tính doanh thu của các class con có dùng đến số đêm, nên để đơn giản, trong phương thức tính doanh thu của class SOFITEL ta chỉ cần trả về số đêm là được.

```
SOFITEL:: SOFITEL() : sodem(0){};

SOFITEL:: SOFITEL(int so_dem) : sodem(so_dem){};

int SOFITEL:: DoanhThu(){
    return sodem;
};
```

2. Xây dựng class con BUSINESS kế thừa từ class cha:

- Không có thuộc tính riêng (private).
- Public: Constructor mặc định, constructor có tham số và phương thức tính doanh thu.

$$\text{Doanh thu Business} = \text{Số đêm} \times 300000$$

Mà số đêm đã được trả về trong phương thức tính doanh thu của class cha, nên ta chỉ cần gọi lại là được.

```
BUSINESS :: BUSINESS() : SOFITEL(){};

BUSINESS :: BUSINESS(int so_dem) : SOFITEL(so_dem){};
```

```
int BUSINESS :: DoanhThu(){  
    return SOFITEL :: DoanhThu()*300000;  
}
```

3. Xây dựng class con PREMIUM kế thừa từ class cha:

- Private: Thêm thuộc tính phí dịch vụ.
- Public: Constructor mặc định, constructor có tham số và phương thức tính doanh thu.

Doanh thu Premium = Số đêm \times 500000 + phí dịch vụ

Mà số đêm đã được trả về trong phương thức tính doanh thu của class cha, nên ta chỉ cần gọi lại là được.

```
PREMIUM :: PREMIUM() : SOFITEL(), phi_dich_vu(0){};
```

```
PREMIUM :: PREMIUM(int so_dem, int phi_dv) :  
SOFITEL(so_dem), phi_dich_vu(phi_dv){};
```

```
int PREMIUM :: DoanhThu(){  
    return SOFITEL :: DoanhThu()* 500000 + phi_dich_vu;  
}
```

4. Xây dựng class con DELUXE kế thừa từ class cha:

- Private: Thêm thuộc tính phí dịch vụ và phí phục vụ.
- Public: Constructor mặc định, constructor có tham số và phương thức tính doanh thu.

Doanh thu Deluxe = Số đêm \times 750000 + phí dịch vụ + phí phục vụ

Mà số đêm đã được trả về trong phương thức tính doanh thu của class cha, nên ta chỉ cần gọi lại là được.

```
DELUXE :: DELUXE() : SOFITEL(), phi_dich_vu(0),  
phi_phuc_vu(0){};
```

```
DELUXE :: DELUXE(int so_dem, int phi_dv, int phi_pv) :  
SOFITEL(so_dem), phi_dich_vu(phi_dv), phi_phuc_vu(phi_pv){};  
  
int DELUXE :: DoanhThu(){  
    return SOFITEL :: DoanhThu()*750000 + phi_dich_vu +  
        phi_phuc_vu;  
}
```

2.4 Thực thi code

Link code: [Bài 2](#)

SOFITEL.h

```
#ifndef SOFITEL_H  
#define SOFITEL_H  
#include <iostream>  
using namespace std;  
  
class SOFITEL{  
    private:  
        int sodem;  
    public:  
        SOFITEL();  
        SOFITEL(int so_dem);  
        int DoanhThu();  
};  
  
#endif
```

SOFITEL.cpp

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

```
/* Phương thức khởi tạo mặc định  
Đầu vào: Không có tham số đầu vào.  
Đầu ra: Không có giá trị trả về.  
Hoạt động: Khởi tạo số đếm lưu trữ (sodem) với giá trị mặc định là 0.  
*/  
SOFITEL::SOFITEL() : sodem(0) {}  
  
/* Phương thức khởi tạo có tham số  
Đầu vào:  
    - so_dem (int): Số đếm lưu trữ.  
Đầu ra: Không có giá trị trả về.  
Hoạt động: Khởi tạo số đếm lưu trữ (sodem) bằng giá trị được truyền  
vào. */  
SOFITEL::SOFITEL(int so_dem) : sodem(so_dem) {}  
  
/* Phương thức tính doanh thu  
Đầu vào: Không có tham số đầu vào.  
Đầu ra: (int) Giá trị doanh thu tính được, dựa trên số đếm lưu trữ.  
Hoạt động: Trả về số đếm lưu trữ (sodem) làm giá trị doanh thu cơ  
bản. */  
int SOFITEL::DoanhThu() {  
    return sodem;  
}
```

BUSINESS.h

```
#ifndef BUSINESS_H  
#define BUSINESS_H  
#include <iostream>  
#include "SOFITEL.h"  
using namespace std;  
  
class BUSINESS : public SOFITEL{
```

```
public:
    BUSINESS();
    BUSINESS(int so_dem);
    int DoanhThu();
};
#endif
```

BUSINESS.cpp

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

/* Phương thức khởi tạo không tham số
   Đầu vào: Không có tham số đầu vào.
   Đầu ra: Không có giá trị trả về.
   Hoạt động: Khởi tạo số đêm lưu trú (sodem) với giá trị mặc định là 0.
   */
BUSINESS::BUSINESS() : SOFITEL() {}

/* Phương thức khởi tạo có tham số
   Đầu vào:
       - so_dem (int): Số đêm lưu trú.
   Đầu ra: Không có giá trị trả về.
   Hoạt động: Khởi tạo số đêm lưu trú (sodem) bằng giá trị được truyền
   vào. */
BUSINESS::BUSINESS(int so_dem) : SOFITEL(so_dem) {}

/* Phương thức tính doanh thu
   Đầu vào: Không có tham số đầu vào.
   Đầu ra: (int) Giá trị doanh thu tính được, bao gồm:
       - Doanh thu cơ bản (số đêm lưu trú nhân với 300,000 VND).
   Hoạt động: Tính tổng doanh thu dựa trên số đêm lưu trú và trả về kết
   quả. */
int BUSINESS::DoanhThu() {
```

```
    return SOFITEL::DoanhThu() * 300000;
}
```

PREMIUM.h

```
#ifndef PREMIUM_H
#define PREMIUM_H
#include <iostream>
#include "SOFITEL.h"
using namespace std;

class PREMIUM : public SOFITEL{
    private:
        int phi_dich_vu;
    public:
        PREMIUM();
        PREMIUM(int so_dem, int phi_dv);
        int DoanhThu();
};

#endif
```

PREMIUM.cpp

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

/* Phương thức khởi tạo mặc định
   Đầu vào: Không có tham số đầu vào.
   Đầu ra: Không có giá trị trả về.
   Hoạt động: Khởi tạo số đêm lưu trú (sodem) với giá trị mặc định là 0
   và phí dịch vụ (phi_dich_vu) là 0. */
```

```
PREMIUM::PREMIUM() : SOFITEL(), phi_dich_vu(0) {}
```

```
/* Phương thức khởi tạo có tham số
```

```
Đầu vào:
```

- so_dem (int): Số đêm lưu trú.*
- phi_dv (int): Phí dịch vụ.*

```
Đầu ra: Không có giá trị trả về.
```

```
Hoạt động: Khởi tạo số đêm lưu trú (sodem) và phí dịch vụ (phi_dich_vu) bằng các giá trị được truyền vào. */
```

```
PREMIUM::PREMIUM(int so_dem, int phi_dv) : SOFITEL(so_dem),  
phi_dich_vu(phi_dv) {}
```

```
/* Phương thức tính doanh thu
```

```
Đầu vào: Không có tham số đầu vào.
```

```
Đầu ra: (int) Giá trị doanh thu tính được, bao gồm:
```

- Doanh thu cơ bản (số đêm lưu trú nhân với 500,000 VND).*
- Phí dịch vụ (phi_dich_vu).*

```
Hoạt động: Tính tổng doanh thu dựa trên số đêm lưu trú và phí dịch vụ, sau đó trả về kết quả. */
```

```
int PREMIUM::DoanhThu() {  
    return SOFITEL::DoanhThu() * 500000 + phi_dich_vu;  
}
```

DELUXE.h

```
#ifndef DELUXE_H  
#define DELUXE_H  
#include <iostream>  
#include "SOFITEL.h"  
using namespace std;  
  
class DELUXE : public SOFITEL{  
    private:  
        int phi_dich_vu, phi_phuc_vu;  
    public:
```



```
DELUXE();  
DELUXE(int so_dem, int phi_dv, int phi_pv);  
int DoanhThu();  
};  
  
#endif
```

DELUXE.cpp

```
#include "SOFITEL.h"  
#include "DELUXE.h"  
#include <iostream>  
using namespace std;  
  
/* Phương thức khởi tạo mặc định  
   Đầu vào: Không có tham số đầu vào.  
   Đầu ra: Không có giá trị trả về.  
   Hoạt động: Gọi constructor của lớp cơ sở SOFITEL để khởi tạo số đêm  
   mặc định,  
               và gán giá trị mặc định 0 cho các thành phần phi_dich_vu  
               và phi_phuc_vu. */  
DELUXE :: DELUXE() : SOFITEL(), phi_dich_vu(0), phi_phuc_vu(0) {}  
  
/* Phương thức khởi tạo có tham số  
   Đầu vào:  
       - so_dem (int): Số đêm lưu trú.  
       - phi_dv (int): Phí dịch vụ.  
       - phi_pv (int): Phí phục vụ.  
   Đầu ra: Không có giá trị trả về.  
   Hoạt động: Gọi constructor của lớp cơ sở SOFITEL để khởi tạo số đêm,  
               đồng thời khởi tạo giá trị cho các thành phần phi_dich_vu  
               và phi_phuc_vu. */  
DELUXE :: DELUXE(int so_dem, int phi_dv, int phi_pv)  
    : SOFITEL(so_dem), phi_dich_vu(phi_dv), phi_phuc_vu(phi_pv) {}  
  
/* Phương thức tính doanh thu
```

2.4. Thực thi code

Đầu vào: Không có tham số đầu vào.

Đầu ra: (int) Giá trị doanh thu tính được.

Hoạt động:

- Tính doanh thu dựa trên số đêm lưu trú (từ lớp cơ sở *SOFITEL*), nhân với hệ số giá phòng (750,000).
- Cộng thêm phí dịch vụ và phí phục vụ.
- Trả về kết quả doanh thu tính được. */

```
int DELUXE :: DoanhThu() {  
    return SOFITEL::DoanhThu() * 750000 + phi_dich_vu + phi_phuc_vu;  
}
```

main_bai2.cpp

```
#include "SOFITEL.h"  
#include "DELUXE.h"  
#include "PREMIUM.h"  
#include "BUSINESS.h"  
  
#include <iostream>  
using namespace std;  
  
int main() {  
    // Tạo các đối tượng với các giá trị doanh thu mẫu  
    DELUXE Khai(3, 250000, 50000);  
    DELUXE Thai(4, 250000, 50000);  
    BUSINESS Luc(5);  
    PREMIUM Phong(2, 150000);  
    PREMIUM Trung(1, 150000);  
  
    // Tính tổng doanh thu cho từng loại phòng  
    int doanhthu_deluxe = Khai.DoanhThu() + Thai.DoanhThu();  
    int doanhthu_business = Luc.DoanhThu();  
    int doanhthu_premium = Phong.DoanhThu() + Trung.DoanhThu();  
  
    // Xác định doanh thu lớn nhất và loại phòng tương ứng
```

```
int doanhthu_lonnhhat = doanhthu_premium;
string loai_phong = "PREMIUM";

if (doanhthu_deluxe > doanhthu_lonnhhat) {
    doanhthu_lonnhhat = doanhthu_deluxe;
    loai_phong = "DELUXE";
}
if (doanhthu_business > doanhthu_lonnhhat) {
    doanhthu_lonnhhat = doanhthu_business;
    loai_phong = "BUSINESS";
}

// In ra loại phòng có doanh thu lớn nhất
cout << "Doanh thu loại phong DELUXE: " << doanhthu_deluxe << "VND"
<< "\n";
cout << "Doanh thu loại phong PREMIUM: " << doanhthu_premium << "VND"
<< "\n";
cout << "Doanh thu loại phong BUSINESS: " << doanhthu_business <<
"VND" << "\n";
cout << "Doanh thu lon nhat la " << loai_phong << "\n";

return 0;
}
```

2.5 Kiểm thử

- Trường hợp 1: DELUXE là loại phòng có doanh thu lớn nhất.
Truyền các đối số như sau:

```
DELUXE Khai(3, 250000, 50000);
DELUXE Thai(4, 250000, 50000);
BUSINESS Luc(5);
PREMIUM Phong(2, 150000);
PREMIUM Trung(1, 150000);
```

Kết quả:

```
Doanh thu loại phong DELUXE: 5850000 VND
Doanh thu loại phong PREMIUM: 1800000 VND
Doanh thu loại phong BUSINESS: 1500000 VND
Doanh thu lớn nhất là DELUXE
```

- Trường hợp 2: PREMIUM là loại phòng có doanh thu lớn nhất.
Truyền các đối số như sau:

```
DELUXE Khai(1, 150000, 20000);
DELUXE Thai(1, 150000, 20000);
BUSINESS Luc(3);
PREMIUM Phong(5, 120000);
PREMIUM Trung(6, 120000);
```

Kết quả:

```
Doanh thu loại phong DELUXE: 1840000 VND
Doanh thu loại phong PREMIUM: 5740000 VND
Doanh thu loại phong BUSINESS: 900000 VND
Doanh thu lớn nhất là PREMIUM
```

- Trường hợp 3: BUSINESS là loại phòng có doanh thu lớn nhất.
Truyền các đối số như sau:

```
DELUXE Khai(1, 100000, 20000);
DELUXE Thai(1, 100000, 20000);
BUSINESS Luc(10);
PREMIUM Phong(2, 50000);
PREMIUM Trung(1, 50000);
```

Kết quả:

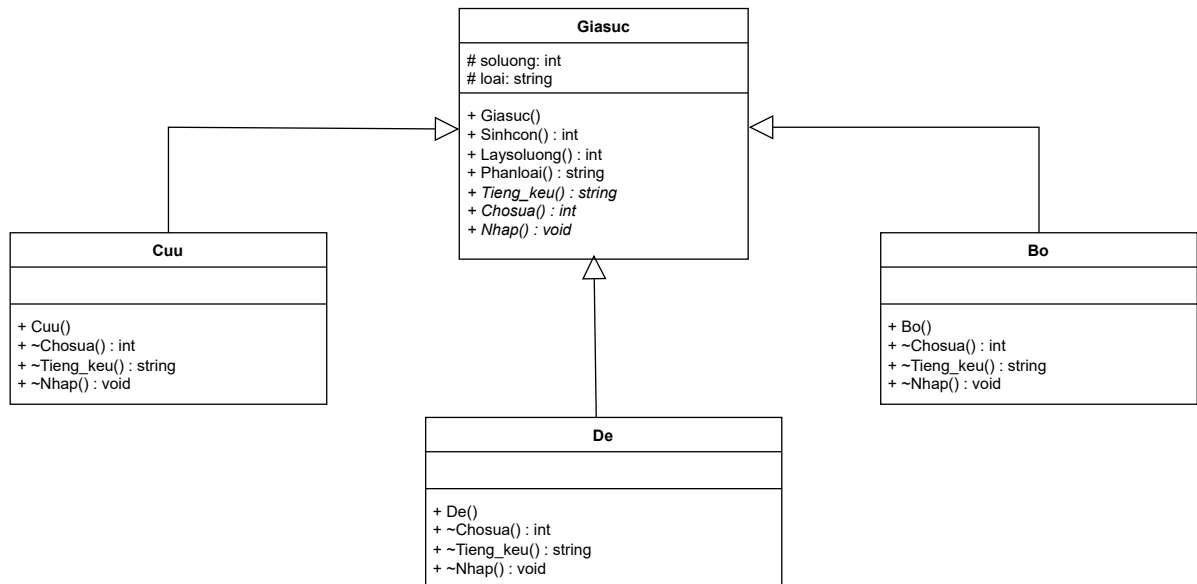
```
Doanh thu loại phong DELUXE: 1740000 VND
Doanh thu loại phong PREMIUM: 1600000 VND
Doanh thu loại phong BUSINESS: 3000000 VND
Doanh thu lớn nhất là BUSINESS
```

3. 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.

- 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



Hình 3.3: Class diagram bài 3

3.2 Mô tả đầu vào và đầu ra

- **Đầu vào:** Nhập vào số lượng của từng loại gia súc ban đầu.
- **Đầu ra:**
 - a) Cho biết tiếng kêu của từng con gia súc.
 - b) Thực hiện các yêu cầu:
 - * Gọi phương thức sinh con.
 - * Gọi phương thức cho sữa.
 - * In ra thông tin của từng loại gia súc: tổng số lượng con (ban đầu và mới sinh), tổng số sữa mà từng loại gia súc đã cho.

3.3 Ý tưởng

1. Xây dựng class Giasuc làm class cha:

- Protected: số lượng và loại gia súc.
- Public: Constructor mặc định, các phương thức nhập, lấy tiếng kêu, cho sữa, sinh con, lấy số lượng và trả về phân loại.
 - Phương thức nhập: Vì các class con cũng dùng nên ta khai báo phương thức ảo. Yêu cầu người dùng nhập vào số lượng của từng loại gia súc.

```
void Giasuc::Nhap() {  
    cin >> soluong;  
}
```

- Phương thức lấy số lượng: Chỉ cần trả về số lượng của từng loại gia súc là được.

```
int Giasuc::Laysoluong() const {  
    return soluong;  
}
```

- Phương thức sinh con: Vì đề yêu cầu sinh một số ngẫu nhiên, nên ta dùng hàm rand(), và để đúng với thực tế nên ta sẽ giới hạn số con được sinh ra nằm trong đoạn từ 0 đến 5. Và để đảm bảo mỗi lần đều sinh một số khác nhau, ta dùng hàm srand(). Cả ba loại gia súc sẽ dùng chung phương thức này, nên ở class con ta không cần ghi đè (override).

```
int Giasuc::Sinhcon() {  
    srand(time(0) + rand());  
    int ketqua = rand() % 6;  
    return ketqua;  
}
```

- Phương thức cho sữa: Vì mỗi loại gia súc thì có giới hạn cho sữa khác nhau, nên ở đây ta khai báo phương thức ảo. Phần code thì ta chỉ cần gọi hàm rand() là được, vì các class con sẽ ghi đè (override) lên phần code này, nên ta không cần quan tâm quá nhiều.

```
int Giasuc::Chosua() {  
    srand(time(0) + rand());  
    int ketqua = rand();  
    return ketqua;  
}
```

- Phương thức lấy tiếng kêu: Tương tự, ta cũng khai báo phương thức ảo, và trong class cha ta trả về chuỗi rỗng.

```
string Giasuc::Tieng_keu() {  
    return "";  
}
```

- Phương thức phân loại gia súc: Trả về loại gia súc tương ứng.

```
string Giasuc::Phanloai() {  
    return loai;  
}
```

2. Xây dựng class Bò là class con kế thừa từ class cha:

- Không có thêm thuộc tính riêng (private).
- Public: Constructor mặc định và các phương thức ghi đè như nhập, cho sữa và lấy tiếng kêu.
 - Constructor mặc định thì gán loại = Bò.

```
Bo::Bo() : Giasuc() {  
    loi = "Bo";  
}
```

- Phương thức cho sữa: Giới hạn từ 0 đến 20 lít.

```
int Bo::Chosua() {  
    srand(time(0) + rand());  
    return Giasuc::Chosua() % 21;  
}
```

- Phương thức lấy tiếng kêu: Bò kêu ưm bôô.

```
string Bo::Tiemg_keu() {  
    return "Um boo ";  
}
```

- Phương thức nhập: Thêm dòng để biết nhập vào số lượng bò. Gọi lại phương thức nhập của class cha.

```
void Bo::Nhap() {  
    cout << "Nhap so luong bo: ";  
    Giasuc :: Nhap();  
}
```

3. Xây dựng class De là class con kế thừa class cha:

- Không có thêm thuộc tính riêng (Private).
- Public: Constructor mặc định và các phương thức ghi đè như nhập, cho sữa và lấy tiếng kêu.
 - Constructor mặc định thì gán loi = Dê.

```
De::De() : Giasuc() {  
    loi = "De";  
}
```

- Phương thức cho sữa: Giới hạn từ 0 đến 10 lít.

```
int De::Chosua() {  
    srand(time(0) + rand());  
    return Giasuc::Chosua() % 11;  
}
```

- Phương thức lấy tiếng kêu: Dê kêu Eeeee.

```
string De::Tieng_keu() {  
    return "Eeee ";  
}
```

- Phương thức nhập: Thêm dòng để biết nhập vào số lượng dê. Gọi lại phương thức nhập của class cha.

```
void De::Nhap() {  
    cout << "Nhap so luong de: ";  
    Giasuc :: Nhap();  
}
```

4. Xây dựng class Cui là class con kế thừa class cha:

- Không có thêm thuộc tính riêng (Private).
- Public: Constructor mặc định và các phương thức ghi đè như nhập, cho sữa và lấy tiếng kêu.
 - Constructor mặc định thì gán loại = Cừu.

```
Cui::Cui() : Giasuc() {  
    loai = "Cui";  
}
```

- Phương thức cho sữa: Giới hạn từ 0 đến 5 lít.

```
int Cui::Chosua() {  
    srand(time(0) + rand());  
    return Giasuc::Chosua() % 6;  
}
```

- Phương thức lấy tiếng kêu: Cừu kêu Beeee.

```
string Cuu::Tieng_keu() {  
    return "Beee ";  
}
```

- Phương thức nhập: Thêm dòng để biết nhập vào số lượng cừu. Gọi lại phương thức nhập của class cha.

```
void Cuu::Nhap() {  
    cout << "Nhap so luong cuu: ";  
    Giasuc :: Nhap();  
}
```

5. Trong file main.cpp

- Gọi hàm nhập ứng với từng loại gia súc để yêu cầu người dùng nhập vào số lượng ban đầu ứng với từng loại.

```
Bo b;  
De d;  
Cuu c;  
  
b.Nhap();  
c.Nhap();  
d.Nhap();
```

- Tiếp theo ta khai báo một con trỏ cấp hai để tiến hành quản lí số lượng các gia súc cũng như để truy xuất dễ hơn vào phương thức của từng loại gia súc:
 - Kích thước của con trỏ cấp hai bằng tổng số lượng ba loại gia súc.
 - Thêm lần lượt từng loại gia súc vào mảng con trỏ đã khai báo.

```
int so_luong = b.Laysoluong() + c.Laysoluong() +  
d.Laysoluong();  
Giasuc** danh_sach_giasuc = new Giasuc*[so_luong];  
int chi_so = 0;  
  
for (int i = 0; i < b.Laysoluong(); i++) {  
    danh_sach_giasuc[chi_so++] = new Bo();  
}
```

```

    }
    for (int i = 0; i < c.Laysoluong(); i++) {
        danh sach_giasuc[chi_so++] = new Cuu();
    }
    for (int i = 0; i < d.Laysoluong(); i++) {
        danh sach_giasuc[chi_so++] = new De();
    }

```

a) Lặp qua các phần tử trong mảng, và truy xuất đến phương thức lấy tiếng kêu.

```

    cout << "Chu di vang, nhưng tieng keu nghe duoc la: ";
    for (int i = 0; i < so_luong; i++) {
        cout << danh sach_giasuc[i]->Tieng_keu();
    }

```

b) Ở yêu cầu b, ta cần viết thêm một hàm thống kê để thống kê số lượng gia súc sau một lượt sinh và một lượt cho sữa.

- Để dễ hình dung, ta tiến hành thống kê theo từng loại gia súc: Kết quả sẽ trả về số sữa và số con mà từng con của loại gia súc đó đã sinh. Ví dụ ban đầu có 2 con bò, thì kết quả sẽ trả về 3 hàng:
 - Hàng 1: Số con và số sữa con bò thứ nhất đã sinh.
 - Hàng 2: Số con và số sữa con bò thứ hai đã sinh.
 - Hàng 3: Tổng số lượng bò (ban đầu + đã sinh), tổng lượng sữa.
- Để làm được điều đó thì ta cần thêm một biến đếm, để quản lí số lượng của từng loại gia súc. Sau mỗi lần sinh con và cho sữa thì ta lại tăng biến đếm lên 1 để quản lí con gia súc tiếp theo.
- Để truy cập vào đúng loại gia súc trong mảng con trở, thì ta gọi phương thức Phân loại đã khai báo ở trên.
- Một lưu ý nữa là phương thức cho sữa chỉ áp dụng với những gia súc ban đầu, con non không cho sữa.

```

    cout << "Bò: \n";
    int dem_bo = 1;
    for (int i = 0; i < soluong; i++) {
        if (danh sach_giasuc[i]->Phanloai() == "Bò") {

```

```
int so_con_sinh = danh_sach_giasuc[i]->Sinhcon();
int so_sua = danh_sach_giasuc[i]->Chosua();

cout << "Bo " << dem_bo++ << ": Sinh " <<
so_con_sinh << " con, Cho " << so_sua << " lit
sua\n";
tong_con_bo += so_con_sinh;
tong_sua_bo += so_sua;

    }
}
cout << "\tTong so bo: " << tong_con_bo + dem_bo - 1 <<
"\n";
cout << "\tTong sua bo: " << tong_sua_bo << " lit\n";
```

-
- Dê và cừu ta cũng thống kê tương tự.

3.4 Thực thi code

Link code: [Bài 3](#)

Giasuc.h

```
#ifndef GIASUC_H
#define GIASUC_H
#include <iostream>
using namespace std;

class Giasuc {
protected:
    int soluong;
    string loai;

public:
    Giasuc();
```

```
virtual void Nhap();  
virtual string Tieng_keu();  
virtual int Chosua();  
int Sinhcon();  
int Laysoluong() const;  
string Phanloai();  
};
```

```
#endif
```

Giasuc.cpp

```
#include "Giasuc.h"  
#include <iostream>  
#include <algorithm>  
#include <ctime>  
using namespace std;  
  
/* Phương thức khởi tạo không tham số  
Đầu vào: Không có tham số đầu vào.  
Đầu ra: Không có giá trị trả về.  
Hoạt động: Khởi tạo số lượng gia súc là 0 và loại gia súc là chuỗi  
rỗng. */  
Giasuc::Giasuc() : soluong(0), loai("") {}  
  
/* Phương thức Nhap  
Đầu vào: Nhập số lượng gia súc từ người dùng.  
Đầu ra: Không có giá trị trả về.  
Hoạt động: Nhập giá trị số lượng gia súc từ người dùng và lưu vào  
biến soluong. */  
void Giasuc::Nhap() {  
    cin >> soluong;  
}
```

```
/* Phương thức Laysoluong
Đầu vào: Không có tham số đầu vào.
Đầu ra: Trả về số lượng gia súc (soluong) của đối tượng.
Hoạt động: Trả về giá trị của biến soluong, thể hiện số lượng gia súc
hiện tại. */
int Giasuc::Laysoluong() const {
    return soluong;
}

/* Phương thức Sinhcon
Đầu vào: Không có tham số đầu vào.
Đầu ra: Trả về số con sinh ra, một số ngẫu nhiên trong khoảng từ 0
đến 5.
Hoạt động: Sinh ra một số ngẫu nhiên đại diện cho số con được sinh
trong một lần sinh. */
int Giasuc::Sinhcon() {
    srand(time(0) + rand());
    int ketqua = rand() % 6;
    return ketqua;
}

/* Phương thức Chosua
Đầu vào: Không có tham số đầu vào.
Đầu ra: Trả về số lượng sữa được gia súc cho, là một số ngẫu nhiên.
Hoạt động: Sinh một số ngẫu nhiên và trả về giá trị này như là lượng
sữa gia súc cho. */
int Giasuc::Chosua() {
    srand(time(0) + rand());
    int ketqua = rand();
    return ketqua;
}

/* Phương thức Tieng_keu
Đầu vào: Không có tham số đầu vào.
Đầu ra: Trả về chuỗi rỗng.
```

3.4. Thực thi code

```
Hoạt động: Phương thức mặc định trả về chuỗi rỗng, các lớp con sẽ ghi
đề phương thức này để trả về tiếng kêu của từng loại gia súc. */
string Giasuc::Tieng_keu() {
    return "";
}

/* Phương thức Phanloai
Đầu vào: Không có tham số đầu vào.
Đầu ra: Trả về loại gia súc (loai).
Hoạt động: Trả về thông tin loại gia súc của đối tượng. */
string Giasuc::Phanloai() {
    return loai;
}
```

Bo.h

```
#ifndef BO_H
#define BO_H
#include <iostream>
# include "Giasuc.h"
using namespace std;

class Bo : public Giasuc {
public:
    Bo();
    int Chosua() override;
    string Tieng_keu() override;
    void Nhap() override;
};

#endif
```

Bo.cpp


```
#include "Giasuc.h"
#include "Bo.h"
#include <iostream>
#include <algorithm>
#include <ctime>
using namespace std;

/* Phương thức khởi tạo không tham số
   Đầu vào: Không có tham số đầu vào.
   Đầu ra: Không có giá trị trả về.
   Hoạt động: Khởi tạo đối tượng Bo với số lượng mặc định là 0
              và gán loại là "Bo". */
Bo::Bo() : Giasuc() {
    loai = "Bo";
}

/* Phương thức Chosua
   Đầu vào: Không có tham số đầu vào.
   Đầu ra: Trả về một số nguyên đại diện cho lượng sữa bò cho, nằm trong
           khoảng từ 0 đến 20.
   Hoạt động: Sử dụng số ngẫu nhiên để tính lượng sữa bò cho,
              đảm bảo giá trị không vượt quá 20. */
int Bo::Chosua() {
    srand(time(0) + rand());
    return Giasuc::Chosua() % 21;
}

/* Phương thức Tieng_keu
   Đầu vào: Không có tham số đầu vào.
   Đầu ra: Trả về một chuỗi biểu thị tiếng kêu của bò ("Um boo").
   Hoạt động: Cung cấp tiếng kêu đặc trưng của loài bò. */
string Bo::Tieng_keu() {
    return "Um boo ";
}
```

3.4. Thực thi code

```
/* Phương thức Nhập  
Đầu vào: Nhập số lượng bò từ người dùng.  
Đầu ra: Không có giá trị trả về.  
Hoạt động: Gọi phương thức Nhập() từ lớp cơ sở để nhập số lượng  
và gán thông tin liên quan đến bò. */  
void Bo::Nhap() {  
    cout << "Nhập số lượng bò: ";  
    Giasuc::Nhap();  
}
```

De.h

```
#ifndef DE_H  
#define DE_H  
#include <iostream>  
#include "Giasuc.h"  
using namespace std;  
  
class De : public Giasuc {  
public:  
    De();  
    int Chosua() override;  
    string Tieng_keu() override;  
    void Nhap() override;  
};  
  
#endif
```

De.cpp

```
#include "Giasuc.h"  
#include "De.h"
```

```
#include <iostream>
#include <algorithm>
#include <ctime>
using namespace std;

/* Phương thức khởi tạo không tham số
   Đầu vào: Không có tham số đầu vào.
   Đầu ra: Không có giá trị trả về.
   Hoạt động: Khởi tạo đối tượng De với số lượng mặc định là 0
              và gán loại là "De". */
De::De() : Giasuc() {
    loai = "De";
}

/* Phương thức Chosua
   Đầu vào: Không có tham số đầu vào.
   Đầu ra: Trả về một số nguyên đại diện cho lượng sữa dê cho, nằm trong
           khoảng từ 0 đến 10.
   Hoạt động: Sử dụng số ngẫu nhiên để tính lượng sữa dê cho,
              đảm bảo giá trị không vượt quá 10. */
int De::Chosua() {
    srand(time(0) + rand());
    return Giasuc::Chosua() % 11;
}

/* Phương thức Tieng_keu
   Đầu vào: Không có tham số đầu vào.
   Đầu ra: Trả về một chuỗi biểu thị tiếng kêu của dê ("Eeee").
   Hoạt động: Cung cấp tiếng kêu đặc trưng của loài dê. */
string De::Tieng_keu() {
    return "Eeee ";
}

/* Phương thức Nhap
   Đầu vào: Nhập số lượng dê từ người dùng.
   Đầu ra: Không có giá trị trả về.
```

3.4. Thực thi code

*Hoạt động: Gọi phương thức Nhap() từ lớp cơ sở để nhập số lượng và gán thông tin liên quan đến dê. */*

```
void De::Nhap() {  
    cout << "Nhap so luong de: ";  
    Giasuc::Nhap();  
}
```

Cuu.h

```
#ifndef CUU_H  
#define CUU_H  
#include <iostream>  
#include "Giasuc.h"  
using namespace std;  
  
class Cuu : public Giasuc {  
public:  
    Cuu();  
    int Chosua() override;  
    string Tieng_keu() override;  
    void Nhap() override;  
};  
  
#endif
```

Cuu.cpp

```
#include "Giasuc.h"  
#include "Cuu.h"  
#include <iostream>  
#include <algorithm>  
#include <ctime>
```

```

using namespace std;

/* Phương thức khởi tạo không tham số
   Đầu vào: Không có tham số đầu vào.
   Đầu ra: Không có giá trị trả về.
   Hoạt động: Khởi tạo đối tượng Cuu với số lượng mặc định là 0
               và gán loại là "Cuu". */
Cuu::Cuu() : Giasuc() {
    loai = "Cuu";
}

/* Phương thức Chosua
   Đầu vào: Không có tham số đầu vào.
   Đầu ra: Trả về một số nguyên đại diện cho lượng sữa cừu cho, nằm
           trong khoảng từ 0 đến 5.
   Hoạt động: Sử dụng số ngẫu nhiên để tính lượng sữa cừu cho,
               đảm bảo giá trị không vượt quá 5. */
int Cuu::Chosua() {
    srand(time(0) + rand());
    return Giasuc::Chosua() % 6;
}

/* Phương thức Tieng_keu
   Đầu vào: Không có tham số đầu vào.
   Đầu ra: Trả về một chuỗi biểu thị tiếng kêu của cừu ("Beee").
   Hoạt động: Cung cấp tiếng kêu đặc trưng của loài cừu. */
string Cuu::Tieng_keu() {
    return "Beee ";
}

/* Phương thức Nhap
   Đầu vào: Nhập số lượng cừu từ người dùng.
   Đầu ra: Không có giá trị trả về.
   Hoạt động: Gọi phương thức Nhap() từ lớp cơ sở để nhập số lượng
               và gán thông tin liên quan đến cừu. */
void Cuu::Nhap() {

```

```
    cout << "Nhap so luong cuu: ";
    Giasuc::Nhap();
}
```

main_bai3.cpp

```
#include <iostream>
#include <algorithm>
#include "Giasuc.h"
#include "Bo.h"
#include "De.h"
#include "Cuu.h"

using namespace std;

/* Phương thức Thống kê
    Đầu vào:
    - danh_sach_giasuc: Mảng con trỏ chứa các đối tượng gia súc.
    - soluong: Số lượng gia súc trong mảng.
    Đầu ra: Không có giá trị trả về.
    Hoạt động:
    - Phương thức thống kê thông tin về số lượng con sinh và sữa cho từng
    loại gia súc (Bò, Cừu, Dê).
    - Đếm số lượng con sinh và lượng sữa cho từng loại gia súc.
    - In ra thông tin tổng hợp cho từng loại gia súc, bao gồm tổng số con
    (ban đầu + đã sinh) và tổng lượng sữa. */
void Thống_kê(Giasuc** danh_sach_giasuc, int soluong) {
    int tong_con_bo = 0, tong_sua_bo = 0;
    int tong_con_de = 0, tong_sua_de = 0;
    int tong_con_cuu = 0, tong_sua_cuu = 0;

    cout << "\nThống kê theo từng loại gia súc:\n";

    // Thống kê cho Bò
```

```

cout << "Bo: \n";
int dem_bo = 1;
for (int i = 0; i < soluong; i++) {
    if (danh sach_giasuc[i]->Phanloai() == "Bo") {
        int so_con_sinh = danh sach_giasuc[i]->Sinhcon();
        int so_sua = danh sach_giasuc[i]->Chosua();

        cout << "Bo " << dem_bo++ << ": Sinh " << so_con_sinh << "
con, Cho " << so_sua << " lit sua\n";
        tong_con_bo += so_con_sinh;
        tong_sua_bo += so_sua;
    }
}
cout << "\tTong so bo: " << tong_con_bo + dem_bo - 1 << "\n";
cout << "\tTong sua bo: " << tong_sua_bo << " lit\n";

// Thống kê cho Cừu
cout << "Cuu: \n";
int dem_cuu = 1;
for (int i = 0; i < soluong; i++) {
    if (danh sach_giasuc[i]->Phanloai() == "Cuu") {
        int so_con_sinh = danh sach_giasuc[i]->Sinhcon();
        int so_sua = danh sach_giasuc[i]->Chosua();

        cout << "Cuu " << dem_cuu++ << ": Sinh " << so_con_sinh << "
con, Cho " << so_sua << " lit sua\n";
        tong_con_cuu += so_con_sinh;
        tong_sua_cuu += so_sua;
    }
}
cout << "\tTong so cuu: " << tong_con_cuu + dem_cuu - 1 << "\n";
cout << "\tTong sua cuu: " << tong_sua_cuu << " lit\n";

// Thống kê cho Dê
cout << "De: \n";

```

```
int dem_de = 1;
for (int i = 0; i < soluong; i++) {
    if (danh sach_giasuc[i]->Phanloai() == "De") {
        int so_con_sinh = danh sach_giasuc[i]->Sinhcon();
        int so_sua = danh sach_giasuc[i]->Chosua();

        cout << "De " << dem_de++ << ": Sinh " << so_con_sinh << "
con, Cho " << so_sua << " lit sua\n";
        tong_con_de += so_con_sinh;
        tong_sua_de += so_sua;
    }
}
cout << "\tTong so de: " << tong_con_de + dem_de - 1 << "\n";
cout << "\tTong sua de: " << tong_sua_de << " lit\n";
}
```

```
int main() {
    Bo b;
    De d;
    Cuu c;

    b.Nhap();
    c.Nhap();
    d.Nhap();

    int so_luong = b.Laysoluong() + c.Laysoluong() + d.Laysoluong();
    Giasuc** danh sach_giasuc = new Giasuc*[so_luong];
    int chi_so = 0;

    for (int i = 0; i < b.Laysoluong(); i++) {
        danh sach_giasuc[chi_so++] = new Bo();
    }
    for (int i = 0; i < c.Laysoluong(); i++) {
        danh sach_giasuc[chi_so++] = new Cuu();
    }
}
```



```
for (int i = 0; i < d.Laysoluong(); i++) {
    danh sach_giasuc[chi_so++] = new De();
}

// Câu a
cout << "Chu di vang, nhưng tieng keu nghe duoc la: ";
for (int i = 0; i < so_luong; i++) {
    cout << danh sach_giasuc[i]->Tieng_keu();
}
cout << "\n";

// Câu b
cout << "\n~~~ THONG TIN GIA SUC ~~~\n" ;
Thongke(danh sach_giasuc, so_luong);

for (int i = 0; i < so_luong; i++) {
    delete danh sach_giasuc[i];
}
delete[] danh sach_giasuc;

return 0;
}
```

3.5 Kiểm thử

- Trường hợp 1: Số lượng gia súc mỗi loại gia súc bằng nhau.

```

Nhap so luong bo: 2
Nhap so luong cuu: 2
Nhap so luong de: 2
Chu di vang, nhưng tieng keu nghe duoc la: Um boo Um boo Beee Beee Eeee Eeee

~%~ THONG TIN GIA SUC ~%~

Thong ke theo tung loai gia suc:
Bo:
Bo 1: Sinh 2 con, Cho 20 lit sua
Bo 2: Sinh 0 con, Cho 8 lit sua
      Tong so bo: 4
      Tong sua bo: 28 lit
Cuu:
Cuu 1: Sinh 3 con, Cho 1 lit sua
Cuu 2: Sinh 3 con, Cho 3 lit sua
      Tong so cuu: 8
      Tong sua cuu: 4 lit
De:
De 1: Sinh 3 con, Cho 9 lit sua
De 2: Sinh 2 con, Cho 6 lit sua
      Tong so de: 7
      Tong sua de: 15 lit

```

- Trường hợp 2: Số lượng gia súc mỗi loại gia súc khác nhau.

```

Nhap so luong bo: 3
Nhap so luong cuu: 1
Nhap so luong de: 4
Chu di vang, nhưng tieng keu nghe duoc la: Um boo Um boo Um boo Beee Eeee Eeee Eeee Eeee

~%~ THONG TIN GIA SUC ~%~

Thong ke theo tung loai gia suc:
Bo:
Bo 1: Sinh 0 con, Cho 10 lit sua
Bo 2: Sinh 1 con, Cho 20 lit sua
Bo 3: Sinh 4 con, Cho 2 lit sua
      Tong so bo: 8
      Tong sua bo: 32 lit
Cuu:
Cuu 1: Sinh 0 con, Cho 1 lit sua
      Tong so cuu: 1
      Tong sua cuu: 1 lit
De:
De 1: Sinh 2 con, Cho 8 lit sua
De 2: Sinh 4 con, Cho 0 lit sua
De 3: Sinh 3 con, Cho 0 lit sua
De 4: Sinh 3 con, Cho 2 lit sua
      Tong so de: 16
      Tong sua de: 10 lit

```

- Trường hợp 3: Nông trại với quy mô nhỏ.

Nhap so luong bo: 10
 Nhap so luong cuu: 12
 Nhap so luong de: 15
 Chu di vang, nhưng tieng keu nghe duoc la: Um boo Um boo Um boo Um boo Um boo Um boo Um boo Um boo Um boo Um boo
 Beee
 Eeee

~%~ THONG TIN GIA SUC ~%~

Thong ke theo tung loai gia suc:

Bo:

Bo 1: Sinh 5 con, Cho 11 lit sua
 Bo 2: Sinh 5 con, Cho 8 lit sua
 Bo 3: Sinh 5 con, Cho 13 lit sua
 Bo 4: Sinh 5 con, Cho 11 lit sua
 Bo 5: Sinh 5 con, Cho 18 lit sua
 Bo 6: Sinh 1 con, Cho 12 lit sua
 Bo 7: Sinh 3 con, Cho 4 lit sua
 Bo 8: Sinh 2 con, Cho 10 lit sua
 Bo 9: Sinh 2 con, Cho 12 lit sua
 Bo 10: Sinh 4 con, Cho 13 lit sua
 Tong so bo: 47
 Tong sua bo: 112 lit

Cuu:

Cuu 1: Sinh 0 con, Cho 1 lit sua
 Cuu 2: Sinh 1 con, Cho 4 lit sua
 Cuu 3: Sinh 2 con, Cho 3 lit sua
 Cuu 4: Sinh 5 con, Cho 4 lit sua
 Cuu 5: Sinh 2 con, Cho 4 lit sua
 Cuu 6: Sinh 2 con, Cho 1 lit sua
 Cuu 7: Sinh 5 con, Cho 5 lit sua
 Cuu 8: Sinh 4 con, Cho 1 lit sua
 Cuu 9: Sinh 2 con, Cho 2 lit sua
 Cuu 10: Sinh 1 con, Cho 4 lit sua
 Cuu 11: Sinh 3 con, Cho 1 lit sua
 Cuu 12: Sinh 4 con, Cho 0 lit sua
 Tong so cuu: 43
 Tong sua cuu: 30 lit

De:

De 1: Sinh 0 con, Cho 6 lit sua
 De 2: Sinh 0 con, Cho 8 lit sua
 De 3: Sinh 0 con, Cho 4 lit sua
 De 4: Sinh 1 con, Cho 10 lit sua
 De 5: Sinh 0 con, Cho 4 lit sua
 De 6: Sinh 0 con, Cho 9 lit sua
 De 7: Sinh 2 con, Cho 6 lit sua
 De 8: Sinh 2 con, Cho 5 lit sua
 De 9: Sinh 4 con, Cho 1 lit sua
 De 10: Sinh 5 con, Cho 6 lit sua
 De 11: Sinh 1 con, Cho 10 lit sua
 De 12: Sinh 0 con, Cho 3 lit sua
 De 13: Sinh 1 con, Cho 7 lit sua
 De 14: Sinh 1 con, Cho 3 lit sua
 De 15: Sinh 1 con, Cho 2 lit sua
 Tong so de: 33
 Tong sua de: 84 lit

- Trường hợp 4: Là một nông trại thật sự.

Bo 17: Sinh 4 con, Cho 5 lit sua

Bo 18: Sinh 5 con, Cho 5 lit sua
Bo 19: Sinh 1 con, Cho 12 lit sua
Bo 20: Sinh 1 con, Cho 8 lit sua
Bo 21: Sinh 2 con, Cho 6 lit sua
Bo 22: Sinh 2 con, Cho 9 lit sua
Bo 23: Sinh 4 con, Cho 13 lit sua
Bo 24: Sinh 5 con, Cho 8 lit sua
Bo 25: Sinh 0 con, Cho 7 lit sua
Bo 26: Sinh 5 con, Cho 11 lit sua
Bo 27: Sinh 3 con, Cho 13 lit sua
Bo 28: Sinh 3 con, Cho 11 lit sua
Bo 29: Sinh 3 con, Cho 5 lit sua
Bo 30: Sinh 1 con, Cho 16 lit sua
Bo 31: Sinh 4 con, Cho 0 lit sua
Bo 32: Sinh 1 con, Cho 10 lit sua
Bo 33: Sinh 2 con, Cho 6 lit sua
Bo 34: Sinh 5 con, Cho 9 lit sua
Bo 35: Sinh 4 con, Cho 2 lit sua
Bo 36: Sinh 4 con, Cho 3 lit sua
Bo 37: Sinh 4 con, Cho 4 lit sua
Bo 38: Sinh 1 con, Cho 2 lit sua
Bo 39: Sinh 5 con, Cho 0 lit sua
Bo 40: Sinh 5 con, Cho 15 lit sua
Bo 41: Sinh 1 con, Cho 11 lit sua
Bo 42: Sinh 2 con, Cho 14 lit sua
Bo 43: Sinh 5 con, Cho 12 lit sua
Bo 44: Sinh 4 con, Cho 5 lit sua
Bo 45: Sinh 1 con, Cho 15 lit sua
Bo 46: Sinh 0 con, Cho 12 lit sua
Bo 47: Sinh 2 con, Cho 17 lit sua
Bo 48: Sinh 0 con, Cho 4 lit sua

Bo 49: Sinh 1 con, Cho 5 lit sua

Bo 50: Sinh 2 con, Cho 8 lit sua

Bo 51: Sinh 2 con, Cho 7 lit sua

Bo 52: Sinh 1 con, Cho 10 lit sua

Bo 53: Sinh 1 con, Cho 8 lit sua

Bo 54: Sinh 4 con, Cho 19 lit sua

Bo 55: Sinh 0 con, Cho 19 lit sua

Bo 56: Sinh 4 con, Cho 3 lit sua

Bo 57: Sinh 2 con, Cho 13 lit sua

Bo 58: Sinh 1 con, Cho 20 lit sua

Bo 59: Sinh 0 con, Cho 8 lit sua

Bo 60: Sinh 3 con, Cho 16 lit sua

Bo 61: Sinh 2 con, Cho 10 lit sua

Bo 62: Sinh 2 con, Cho 11 lit sua

Bo 63: Sinh 2 con, Cho 13 lit sua

Bo 64: Sinh 4 con, Cho 14 lit sua

Bo 65: Sinh 5 con, Cho 12 lit sua

Bo 66: Sinh 0 con, Cho 4 lit sua

Bo 67: Sinh 5 con, Cho 2 lit sua

Bo 68: Sinh 0 con, Cho 7 lit sua

Bo 69: Sinh 3 con, Cho 10 lit sua

Bo 70: Sinh 1 con, Cho 17 lit sua

Bo 71: Sinh 0 con, Cho 5 lit sua

Bo 72: Sinh 1 con, Cho 3 lit sua

Bo 73: Sinh 1 con, Cho 0 lit sua

Bo 74: Sinh 3 con, Cho 3 lit sua

Bo 75: Sinh 1 con, Cho 12 lit sua

Bo 76: Sinh 3 con, Cho 3 lit sua

Bo 77: Sinh 4 con, Cho 12 lit sua

Bo 78: Sinh 3 con, Cho 8 lit sua

Bo 79: Sinh 4 con, Cho 7 lit sua

Bo 80: Sinh 3 con, Cho 10 lit sua

Bo 81: Sinh 1 con, Cho 15 lit sua

Bo 82: Sinh 4 con, Cho 2 lit sua

Tong so bo: 276

Tong sua bo: 751 lit

Cuu:

Cuu 1: Sinh 2 con, Cho 3 lit sua

Cuu 2: Sinh 5 con, Cho 5 lit sua

Cuu 3: Sinh 2 con, Cho 5 lit sua

Cuu 4: Sinh 1 con, Cho 4 lit sua

Cuu 5: Sinh 2 con, Cho 4 lit sua

Cuu 6: Sinh 2 con, Cho 4 lit sua

Cuu 7: Sinh 2 con, Cho 1 lit sua

Cuu 8: Sinh 2 con, Cho 1 lit sua

Cuu 9: Sinh 1 con, Cho 0 lit sua

Cuu 10: Sinh 1 con, Cho 4 lit sua

Cuu 11: Sinh 1 con, Cho 1 lit sua

Cuu 12: Sinh 2 con, Cho 2 lit sua

Cuu 13: Sinh 0 con, Cho 4 lit sua

Cuu 14: Sinh 3 con, Cho 3 lit sua

Cuu 15: Sinh 4 con, Cho 3 lit sua

Cuu 16: Sinh 1 con, Cho 2 lit sua

Cuu 17: Sinh 4 con, Cho 4 lit sua

Cuu 18: Sinh 1 con, Cho 4 lit sua

Cuu 19: Sinh 3 con, Cho 1 lit sua

Cuu 20: Sinh 2 con, Cho 3 lit sua

Cuu 21: Sinh 0 con, Cho 1 lit sua

Cuu 22: Sinh 5 con, Cho 4 lit sua

Cuu 23: Sinh 5 con, Cho 0 lit sua

Cuu 24: Sinh 2 con, Cho 4 lit sua

Cuu 25: Sinh 3 con, Cho 3 lit sua
Cuu 26: Sinh 0 con, Cho 3 lit sua
Cuu 27: Sinh 3 con, Cho 5 lit sua
Cuu 28: Sinh 1 con, Cho 1 lit sua
Cuu 29: Sinh 3 con, Cho 2 lit sua
Cuu 30: Sinh 2 con, Cho 5 lit sua
Cuu 31: Sinh 3 con, Cho 3 lit sua
Cuu 32: Sinh 4 con, Cho 2 lit sua
Cuu 33: Sinh 5 con, Cho 3 lit sua
Cuu 34: Sinh 5 con, Cho 0 lit sua
Cuu 35: Sinh 5 con, Cho 4 lit sua
Cuu 36: Sinh 2 con, Cho 5 lit sua
Cuu 37: Sinh 4 con, Cho 4 lit sua
Cuu 38: Sinh 4 con, Cho 4 lit sua
Cuu 39: Sinh 5 con, Cho 5 lit sua
Cuu 40: Sinh 1 con, Cho 2 lit sua
Cuu 41: Sinh 5 con, Cho 3 lit sua
Cuu 42: Sinh 4 con, Cho 5 lit sua
Cuu 43: Sinh 4 con, Cho 1 lit sua

Tong so cuu: 159

Tong sua cuu: 127 lit

De:

De 1: Sinh 0 con, Cho 1 lit sua
De 2: Sinh 3 con, Cho 5 lit sua
De 3: Sinh 5 con, Cho 2 lit sua
De 4: Sinh 4 con, Cho 6 lit sua
De 5: Sinh 1 con, Cho 0 lit sua
De 6: Sinh 5 con, Cho 6 lit sua
De 7: Sinh 0 con, Cho 2 lit sua
De 8: Sinh 0 con, Cho 9 lit sua
De 9: Sinh 1 con, Cho 3 lit sua

De 10: Sinh 0 con, Cho 1 lit sua
De 11: Sinh 1 con, Cho 10 lit sua
De 12: Sinh 0 con, Cho 10 lit sua
De 13: Sinh 0 con, Cho 1 lit sua
De 14: Sinh 3 con, Cho 10 lit sua
De 15: Sinh 1 con, Cho 2 lit sua
De 16: Sinh 4 con, Cho 10 lit sua
De 17: Sinh 5 con, Cho 7 lit sua
De 18: Sinh 2 con, Cho 6 lit sua
De 19: Sinh 1 con, Cho 6 lit sua
De 20: Sinh 1 con, Cho 2 lit sua
De 21: Sinh 3 con, Cho 6 lit sua
De 22: Sinh 4 con, Cho 7 lit sua
De 23: Sinh 3 con, Cho 9 lit sua
De 24: Sinh 4 con, Cho 2 lit sua
De 25: Sinh 3 con, Cho 3 lit sua
De 26: Sinh 1 con, Cho 0 lit sua
De 27: Sinh 4 con, Cho 2 lit sua
De 28: Sinh 2 con, Cho 10 lit sua
De 29: Sinh 5 con, Cho 3 lit sua
De 30: Sinh 2 con, Cho 9 lit sua
De 31: Sinh 1 con, Cho 2 lit sua
De 32: Sinh 2 con, Cho 5 lit sua
De 33: Sinh 2 con, Cho 8 lit sua
De 34: Sinh 2 con, Cho 0 lit sua
De 35: Sinh 2 con, Cho 5 lit sua
De 36: Sinh 1 con, Cho 1 lit sua
De 37: Sinh 1 con, Cho 3 lit sua
De 38: Sinh 1 con, Cho 9 lit sua
De 39: Sinh 2 con, Cho 9 lit sua
De 40: Sinh 0 con, Cho 4 lit sua

De 41: Sinh 3 con, Cho 8 lit sua
De 42: Sinh 4 con, Cho 9 lit sua
De 43: Sinh 1 con, Cho 2 lit sua
De 44: Sinh 4 con, Cho 0 lit sua
De 45: Sinh 1 con, Cho 2 lit sua
De 46: Sinh 3 con, Cho 6 lit sua
De 47: Sinh 2 con, Cho 0 lit sua
De 48: Sinh 0 con, Cho 8 lit sua
De 49: Sinh 5 con, Cho 1 lit sua
De 50: Sinh 5 con, Cho 5 lit sua
De 51: Sinh 2 con, Cho 0 lit sua
De 52: Sinh 3 con, Cho 10 lit sua
De 53: Sinh 0 con, Cho 8 lit sua
De 54: Sinh 3 con, Cho 3 lit sua
De 55: Sinh 1 con, Cho 10 lit sua
De 56: Sinh 3 con, Cho 3 lit sua
De 57: Sinh 2 con, Cho 2 lit sua
De 58: Sinh 3 con, Cho 4 lit sua
De 59: Sinh 4 con, Cho 3 lit sua
De 60: Sinh 5 con, Cho 0 lit sua
De 61: Sinh 5 con, Cho 4 lit sua
De 62: Sinh 5 con, Cho 6 lit sua
De 63: Sinh 2 con, Cho 3 lit sua
De 64: Sinh 4 con, Cho 1 lit sua
De 65: Sinh 4 con, Cho 8 lit sua
De 66: Sinh 5 con, Cho 10 lit sua
De 67: Sinh 1 con, Cho 10 lit sua
De 68: Sinh 5 con, Cho 2 lit sua
De 69: Sinh 4 con, Cho 10 lit sua
De 70: Sinh 4 con, Cho 9 lit sua
De 71: Sinh 0 con, Cho 1 lit sua

De 72: Sinh 3 con, Cho 5 lit sua
De 73: Sinh 5 con, Cho 2 lit sua
De 74: Sinh 4 con, Cho 6 lit sua
De 75: Sinh 1 con, Cho 0 lit sua
De 76: Sinh 5 con, Cho 6 lit sua
De 77: Sinh 0 con, Cho 2 lit sua
De 78: Sinh 0 con, Cho 9 lit sua
De 79: Sinh 1 con, Cho 3 lit sua
De 80: Sinh 0 con, Cho 1 lit sua
De 81: Sinh 1 con, Cho 10 lit sua
De 82: Sinh 0 con, Cho 10 lit sua
De 83: Sinh 0 con, Cho 1 lit sua
De 84: Sinh 3 con, Cho 10 lit sua
De 85: Sinh 1 con, Cho 2 lit sua
De 86: Sinh 4 con, Cho 10 lit sua
De 87: Sinh 5 con, Cho 7 lit sua
De 88: Sinh 2 con, Cho 6 lit sua
De 89: Sinh 1 con, Cho 6 lit sua
De 90: Sinh 1 con, Cho 2 lit sua
De 91: Sinh 3 con, Cho 6 lit sua
De 92: Sinh 4 con, Cho 7 lit sua
De 93: Sinh 3 con, Cho 9 lit sua
De 94: Sinh 4 con, Cho 2 lit sua
De 95: Sinh 3 con, Cho 3 lit sua
De 96: Sinh 1 con, Cho 0 lit sua
De 97: Sinh 4 con, Cho 2 lit sua
De 98: Sinh 2 con, Cho 10 lit sua
De 99: Sinh 5 con, Cho 3 lit sua
De 100: Sinh 2 con, Cho 9 lit sua
De 101: Sinh 1 con, Cho 2 lit sua

Tong so de: 345

Tong sua de: 495 lit