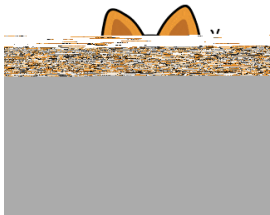




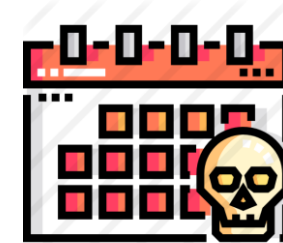
# THIẾT KẾ LỚP CPHANSO

WELCOME





NGUYỄN VĂN LINH -20520613  
HỒNG GIA HY -20520561  
NGUYỄN HUY TRÍ DŨNG -20520459  
ĐỖ TRỌNG TÌNH -20520318  
NGUYỄN CÔNG ĐOÀN-20520447



Sit Dolor Amet



# CPHANSO

Thiết Kế

## Thiết kế lớp CPhanSo

- Thuộc tính :

- + Tử
- + Mẫu

- Phương thức:

- + Nhóm phương thức khởi tạo
- + Nhóm phương thức chung cấp thông tin
- + Nhóm phương thức cập nhật thông tin
- + Nhóm phương thức xử lý
- + Nhóm phương thức kiểm tra

### -Nhóm phương thức khởi tạo:

1. Mặc định
2. Sao chép
3. Khi biết đầy đủ thông tin
4. Phương thức thiết lập mặc định
5. Phương thức thiết lập sao chép
6. Phương thức thiết lập khi biết đầy đủ thông tin
7. Phương pháp nhập
8. Toán tử vào



## Thiết kế lớp CPhanSo

- Nhóm phương thức cung cấp thông tin:

1. Phương thức xuất.

2. Toán tử ra.

3. Phương thức cung cấp Tử

4. Phương thức cung cấp Mẫu

## Thiết kế lớp CPhanSo

- Nhóm phương thức cập nhật thông tin:

1. Toán tử bằng.

2. Phương thức cập nhật Tử

3. Phương thức cập nhật Mẫu

## Thiết kế lớp CPhanSo

- Nhóm phương thức kiểm tra thông tin:
  - + Phương thức kiểm tra có phải là phân số hay không
  - + Phương thức kiểm tra có bằng không hay không?
  - + Phương thức kiểm tra phân số có dương hay không?
  - + Phương thức kiểm tra phân số có âm hay không?



## Thiết kế lớp CPhanSo

- Nhóm phương thức xử lý thông tin:

- + Toán tử so sánh bằng
- + Toán tử so sánh khác
- + Toán tử so sánh lớn hơn
- + Toán tử so sánh lớn hơn bằng
- + Toán tử so sánh bé hơn
- + Toán tử so sánh bé hơn bằng
- + Toán tử cộng
- + Toán tử -

## Thiết kế lớp CPhanSo

- Nhóm phương thức xử lý thông tin:

+ Toán tử \*

+ Toán tử /

+ Toán tử +=

+ Toán tử -=

+ Toán tử \*=

+ Toán tử /=

+ Phương thức phá hủy



# CPHANSO

Khai báo lớp



## Thiết kế lớp CPhanSo

```
class CDonThuc
{
private:
    float tu;
    float mau;
public:
    // Nhóm phương thức khởi tạo
    // Nhóm phương thức cung cấp thông tin
    // Nhóm phương thức cập nhật thông tin
    // Nhóm phương thức kiểm tra
    // Nhóm phương thức xử lý
```

## Thiết kế lớp CPhanSo

// Phương thức khởi tạo

CPhanSo();

CPhanSo(float, float);

CPhanSo(const CPhanSo &);

void KhoiTao();

void KhoiTao(float, float);

void KhoiTao(const CPhanSo &);

friend istream &operator>>(istream &, CPhanSo &);

void Nhap();

## Thiết kế lớp CPhanSo

// Phương thức cung cấp thông tin

float getTu();

float getMau();

friend ostream &operator<<(ostream &, CPhanSo &);

void Xuat();



## Thiết kế lớp CPhanSo

// Phương thức cập nhập thông tin

```
void setHs(float);
```

```
void setMu_x(float);
```

```
CDonThuc &operator=(const CDonThuc &);
```

// Phương thức kiểm tra

```
int isPhanSo();
```

```
int BangKhong();
```

```
int PSDuong();
```

```
int PSAm();
```

## Thiết kế lớp CPhanSo

// Phương thức xử lý

```
int operator==(CPhanSo);  
int operator!=(CPhanSo);  
int operator>(CPhanSo);  
int operator>=(CPhanSo);  
int operator<(CPhanSo);  
int operator<=(CPhanSo);  
CPhanSo operator+(const CPhanSo &);  
CPhanSo operator-(const CPhanSo &);
```

## Thiết kế lớp CPhanSo

// Phương thức xử lý

```
CPhanSo operator*(const CPhanSo &);  
CPhanSo operator/(const CPhanSo &);  
CPhanSo operator+=(const CPhanSo &);  
CPhanSo operator-=(const CPhanSo &);  
CPhanSo operator*=(const CPhanSo &);  
CPhanSo operator/=(const CPhanSo &);  
~CPhanSo();
```





# PHƯƠNG THỨC KHỞI TẠO

CPhanSo

## Nhóm phương thức khởi tạo

// Phương thức thiết lập mặc định

CPhanSo::CPhanSo()

{

tu = 0;

mau = 1;

}

## Nhóm phương thức khởi tạo

// Phương thức thiết lập khi biết đầy đủ thông tin

```
CPhanSo::CPhanSo(float tu, float mau)
```

```
{
```

```
    this->tu = tu;
```

```
    this->mau = mau;
```

```
}
```

// Phương thức thiết lập sao chép

```
CPhanSo::CPhanSo(const CPhanSo &a)
```

```
{
```

```
    tu = a.tu;
```

```
    mau = a.mau;
```

```
}
```



## Nhóm phương thức khởi tạo

// Phương thức khởi tạo mặc định

```
void CPhanSo::KhoiTao()
```

```
{
```

```
    tu = 0;
```

```
    mau = 1;
```

```
}
```

// Phương thức khởi tạo khi biết đầy đủ thông tin

```
void CPhanSo::KhoiTao(float tu, float mau)
```

```
{
```

```
    this->tu = tu;
```

```
    this->mau = mau;
```

```
}
```

// Phương thức khởi tạo sao chép

```
void CPhanSo::KhoiTao(const CPhanSo &a){
```

```
    tu = a.tu;
```

```
    mau = a.mau;
```

```
}
```

// Toán tử nhập

```
istream &operator>>(istream &is, CPhanSo &a){
```

```
    cout << "Nhap tu: ";
```

```
    is >> a.tu;
```

```
    cout << "Nhap mau: ";
```

```
    is >> a.mau;
```

```
    return is;
```

```
}
```

## Nhóm phương thức khởi tạo

// Phương thức nhập

```
void CPhanSo::Nhap(){  
    cout << "Nhap tu: ";  
    cin >> tu;  
    cout << "Nhap mau: ";  
    cin >> mau;  
}
```



# PHƯƠNG THỨC CUNG CẤP THÔNG TIN

CPhanSo



## Nhóm phương thức cung cấp thông tin

// Phương thức cung cấp tử.

```
float CPhanSo::getTu(){  
    return tu;  
}
```

// Phương thức cung cấp mẫu

```
float CPhanSo::getMau(){  
    return mau;  
}
```

## Nhóm phương thức cung cấp thông tin

// Toán tử xuất

```
ostream &operator<<(ostream &os, CPhanSo &a){  
    os << "Tu: " << a.tu << endl;  
    os << "Mau: " << a.mau << endl;  
    return os;  
}
```

// Phương thức xuất

```
void CPhanSo::Xuat(){  
    cout << "Tu: " << tu << endl;  
    cout << "Mau: " << mau << endl;  
}
```



# PHƯƠNG THỨC CẬP NHẬT THÔNG TIN

CPhanSo

## Nhóm phương thức cập nhật thông tin

// Phương thức cập nhật tử

```
void CPhanSo::setTu(float tu)
```

```
{
```

```
    this->tu = tu;
```

```
}
```

// Phương thức cập nhật mẫu

```
void CPhanSo::setMau(float mau)
```

```
{
```

```
    this->mau = mau;
```

```
}
```

## Nhóm phương thức cập nhật thông tin

```
// Toán tử =  
CPhanSo &CPhanSo::operator=(const CPhanSo &a)  
{  
    tu = a.tu;  
    mau = a.mau;  
    return *this;  
}
```





# PHƯƠNG THỨC KIỂM TRA

CPhanSo

## Nhóm phương thức kiểm tra thông tin

// Phương thức kiểm tra có phải là phân số hay không?

```
int CPhanSo::isPhanSo()  
{  
    return (mau != 0);  
}
```

// Phương thức kiểm tra có bằng không hay không?

```
int CPhanSo::BangKhong()  
{  
    return (tu == 0 && mau != 0);  
}
```

## Nhóm phương thức kiểm tra thông tin

// Phương thức kiểm tra phân số có dương hay không?

```
int CPhanSo::PSDuong()
```

```
{
```

```
    return (tu * mau > 0);
```

```
}
```

// Phương thức kiểm tra phân số có âm hay không?

```
int CPhanSo::PSAm()
```

```
{
```

```
    return (tu * mau < 0);
```

```
}
```



# PHƯƠNG THỨC XỬ LÝ

CPhanSo

## Nhóm phương thức xử lý thông tin

// Toán tử so sánh bằng

```
int CPhanSo::operator==(CPhanSo a)
{
    return (a.tu == tu && a.mau == mau);
}
```

// Toán tử so sánh khác

```
int CPhanSo::operator!=(CPhanSo a)
{
    return (a.tu != tu || a.mau != mau);
}
```



## Nhóm phương thức xử lý thông tin

// Toán tử so sánh lớn hơn

```
int CPhanSo::operator>(CPhanSo a)
```

```
{
```

```
    float temp = tu / mau;
```

```
    float temp1 = a.tu / a.mau;
```

```
    if (temp - temp1 > 0)
```

```
        return 1;
```

```
    return 0;
```

```
}
```

## Nhóm phương thức xử lý thông tin

// Toán tử so sánh lớn hơn bằng

```
int CPhanSo::operator>=(CPhanSo a)
```

```
{
```

```
    float temp = tu / mau;
```

```
    float temp1 = a.tu / a.mau;
```

```
    if (temp - temp1 >= 0)
```

```
        return 1;
```

```
    return 0;
```

```
}
```

## Nhóm phương thức xử lý thông tin

// Toán tử so sánh bé hơn

```
int CPhanSo::operator<(CPhanSo a)
{
    float temp = tu / mau;
    float temp1 = a.tu / a.mau;
    if (temp - temp1 < 0)
        return 1;
    return 0;
}
```

## Nhóm phương thức xử lý thông tin

// Toán tử so sánh bé hơn bằng

```
int CPhanSo::operator<=(CPhanSo a)
```

```
{
```

```
    float temp = tu / mau;
```

```
    float temp1 = a.tu / a.mau;
```

```
    if (temp - temp1 <= 0)
```

```
        return 1;
```

```
    return 0;
```

```
}
```

## Nhóm phương thức xử lý thông tin

// Toán tử +

```
CPhanSo CPhanSo::operator+(const CPhanSo &a){  
    CPhanSo temp;  
    temp.tu = tu*a.mau + a.tu*mau;  
    temp.mau = mau*a.mau;  
    return temp;  
}
```



## Nhóm phương thức xử lý thông tin

// Toán tử -

```
CPhanSo CPhanSo::operator-(const CPhanSo &a){  
    CPhanSo temp;  
    temp.tu = tu*a.mau - a.tu*mau;  
    temp.mau = mau*a.mau;  
    return temp;  
}
```

## Nhóm phương thức xử lý thông tin

// Toán tử \*

```
CPhanSo CPhanSo::operator*(const CPhanSo &a){  
    CPhanSo temp;  
    temp.tu = tu* a.tu;  
    temp.mau = mau*a.mau;  
    return temp;  
}
```

## Nhóm phương thức xử lý thông tin

// Toán tử /

```
CPhanSo CPhanSo::operator/(const CPhanSo &a){  
    CPhanSo temp;  
    temp.tu = tu* a.mau;  
    temp.mau = mau*a.tu;  
    return temp;  
}
```

## Nhóm phương thức xử lý thông tin

// Toán tử +=

```
CPhanSo CPhanSo::operator+=(const CPhanSo &a){  
    tu = tu*a.mau + a.tu*mau;  
    mau = mau*a.mau;  
    return *this;  
}
```

## Nhóm phương thức xử lý thông tin

// Toán tử -=

```
CPhanSo CPhanSo::operator-=(const CPhanSo &a){  
    tu = tu*a.mau - a.tu*mau;  
    mau = mau*a.mau;  
    return *this;  
}
```



## Nhóm phương thức xử lý thông tin

// Toán tử \*=

```
CPhanSo CPhanSo::operator*=(const CPhanSo &a){  
    tu = tu* a.tu;  
    mau = mau*a.mau;  
    return *this;  
}
```

## Nhóm phương thức xử lý thông tin

// Toán tử /=

```
CPhanSo CPhanSo::operator/=(const CPhanSo &a){  
    tu = tu* a.mau;  
    mau = mau*a.tu;  
    return *this;  
}
```

// Phương thức phá hủy

```
CPhanSo::~~CPhanSo(){  
    return;  
}
```

The background is a dark navy blue. It is decorated with various geometric shapes in orange and red. In the top left, there is a large orange semi-circle. Scattered around are smaller shapes: red circles, orange dashed lines, a red triangle, an orange triangle, a red inverted triangle, a red square, and a red parallelogram. The word "THANKYOU" is centered in a large, white, sans-serif font.

# THANKYOU