

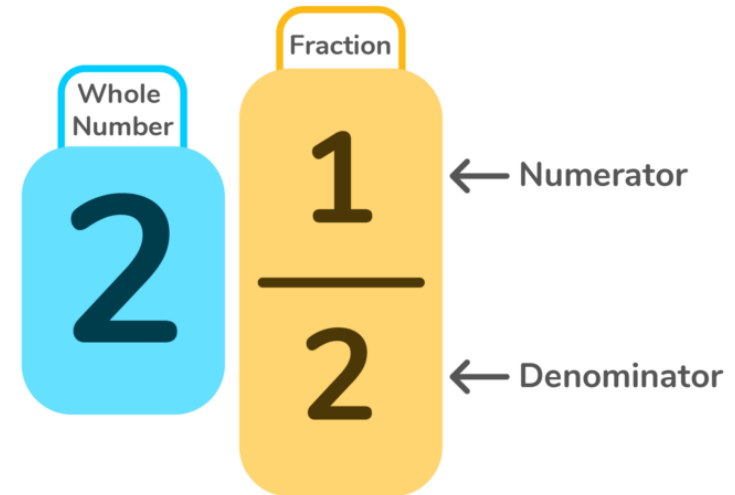
# Chương 9 – THIẾT KẾ LỚP HỒN SỐ

- Nguyễn Hữu Lợi
- Đoàn Chánh Thống
- ThS. Nguyễn Thành Hiệp
- ThS. Trương Quốc Dũng
- ThS. Võ Duy Nguyên
- ThS. Nguyễn Văn Toàn
- TS. Nguyễn Duy Khánh
- TS. Nguyễn Tấn Trần Minh Khang

# THIẾT KẾ LỚP HỖN SỐ

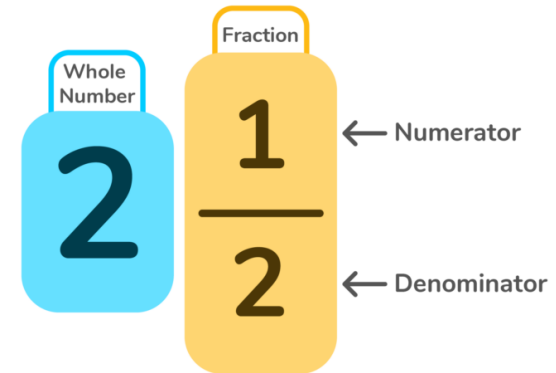
# Thiết kế lớp CHonSo

- Thuộc tính
  - + Phần Nguyên.
  - + Tử số.
  - + Mẫu số.



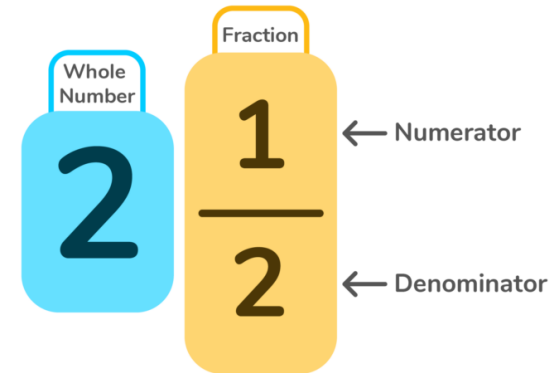
# Thiết kế lớp CHonSo

- Các nhóm phương thức
  - + 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 CHonSo

- Nhóm phương thức khởi tạo
  - + Phương thức Nhập.
  - + Toán tử vào (operator >>).
  - + ...



# Thiết kế lớp CHonSo

– Nhóm phương thức khởi tạo (tiếp tục)

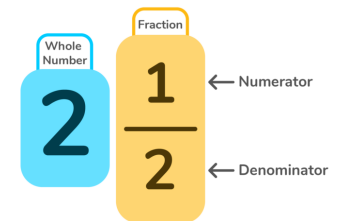
+ ...

+ Phương thức thiết lập mặc định.

+ Phương thức thiết lập sao chép.

+ Phương thức thiết lập khi biết đầy đủ thông tin.

+ ...



# Thiết kế lớp CHonSo

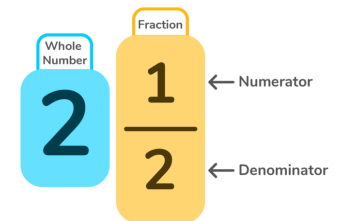
– Nhóm phương thức khởi tạo (tiếp tục)

+ ...

+ Phương thức thiết lập khi biết tử, biết mẫu, phần nguyên lấy mặc định.

+ Phương thức thiết lập khi biết phần nguyên, tử lấy mặc định, mẫu lấy mặc định.

+ ...



# Thiết kế lớp CHonSo

– Nhóm phương thức khởi tạo (tiếp tục)

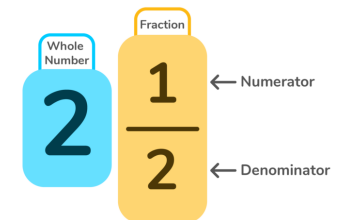
+ ...

+ Phương thức khởi tạo mặc định.

+ Phương thức khởi tạo sao chép.

+ Phương thức khởi tạo khi biết đầy đủ thông tin.

+ ...





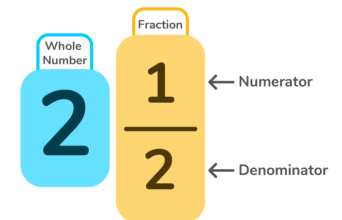
# Thiết kế lớp CHonSo

– Nhóm phương thức khởi tạo (tiếp tục)

+ ...

+ Phương thức khởi tạo khi biết tử, biết mẫu, phần nguyên lấy mặc định.

+ Phương thức khởi tạo khi biết phần nguyên, tử lấy mặc định, mẫu lấy mặc định.



+ Kết thúc nhóm phương thức khởi tạo

# Thiết kế lớp CHonSo

– Nhóm phương thức cung cấp thông tin

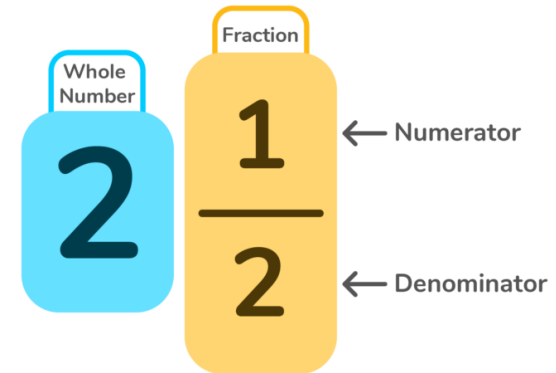
+ Phương thức Xuất.

+ Toán tử ra (operator <<).

+ ...

+ Lớp CHonSo có bao nhiêu thuộc tính.

+ Trả lời: Lớp CHonSo có 3 thuộc tính.



# Thiết kế lớp CHonSo

– Nhóm phương thức cung cấp thông tin (tiếp tục)

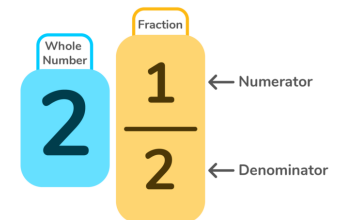
+ ...

+ Phương thức cung cấp phần nguyên.

+ Phương thức cung cấp tử.

+ Phương thức cung cấp mẫu.

+ ...

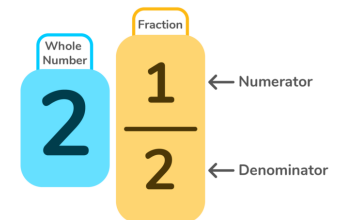


# Thiết kế lớp CHonSo

– Nhóm phương thức cung cấp thông tin (tiếp tục)

+ ...

+ Phương thức cung cấp giá trị thực.



+ Kết thúc nhóm phương thức cung cấp thông tin.

# Thiết kế lớp CHonSo

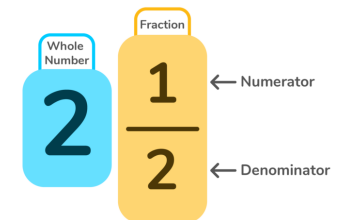
– Nhóm phương thức cập nhật thông tin

+ Toán tử gán (operator =).

+ ...

+ Lớp CHonSo có bao nhiêu thuộc tính.

+ Trả lời: Lớp CHonSo có 3 thuộc tính.



# Thiết kế lớp CHonSo

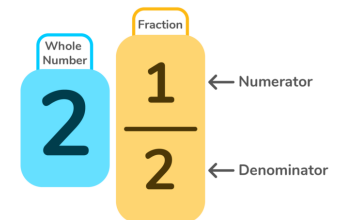
– Nhóm phương thức cập nhật thông tin (tiếp tục)

+ ...

+ Phương thức cập nhật phần nguyên.

+ Phương thức cập nhật tử.

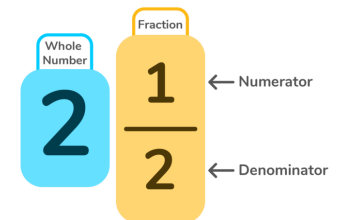
+ Phương thức cập nhật mẫu.



+ Kết thúc nhóm phương thức cung cấp thông tin.

# Thiết kế lớp CHonSo

- Nhóm phương thức kiểm tra
  - + Toán tử so sánh bằng.
  - + Toán tử so sánh khác.
  - + Toán tử so sánh lớn hơn.
  - + Toán tử so sánh nhỏ hơn.
  - + Toán tử so sánh lớn hơn bằng.
  - + Toán tử so sánh nhỏ hơn bằng.
  - + ...



# Thiết kế lớp CHonSo

## – Nhóm phương thức kiểm tra (tiếp tục)

+ ...

+ Kiểm tra hỗn số có nghĩa hay không?

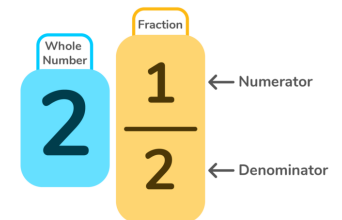
+ Kiểm tra hỗn số có tối giản hay không?

+ Kiểm tra hỗn số có bằng không hay không?

+ Kiểm tra hỗn số dương không?

+ Kiểm tra hỗn số âm không?

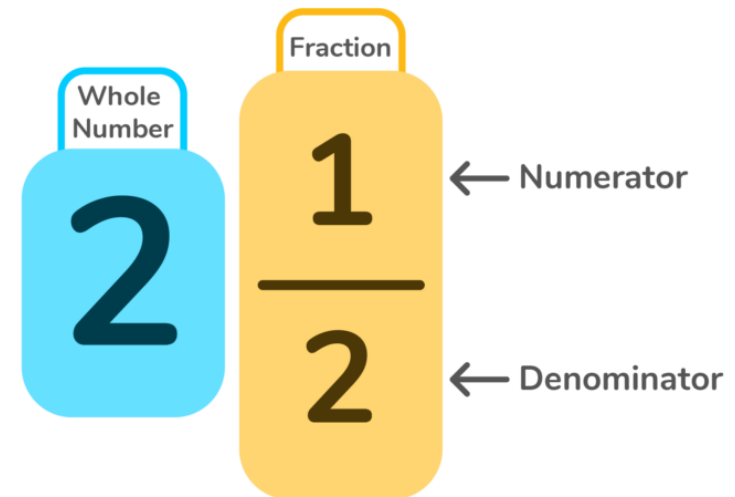
+ Kết thúc nhóm phương thức kiểm tra.





# Thiết kế lớp CHonSo

- Nhóm phương thức xử lý
  - + Phương thức phá hủy.
  - + ...



# Thiết kế lớp CHonSo

## – Nhóm phương thức xử lý (tiếp tục)

+ ...

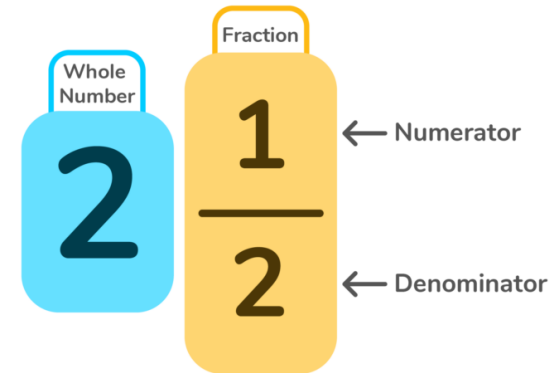
+ Toán tử cộng hai hỗn số.

+ Toán tử trừ hai hỗn số.

+ Toán tử nhân hai hỗn số.

+ Toán tử chia hai hỗn số.

+ ...



# Thiết kế lớp CHonSo

## – Nhóm phương thức xử lý (tiếp tục)

+ ...

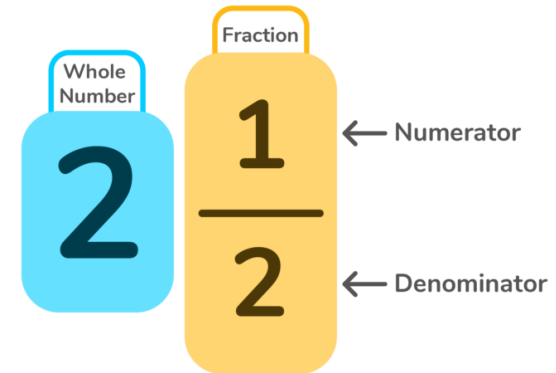
+ Toán tử cộng bằng hai hỗn số.

+ Toán tử trừ bằng hai hỗn số.

+ Toán tử nhân bằng hai hỗn số.

+ Toán tử chia bằng hai hỗn số.

+ ...



# Thiết kế lớp CHonSo

## – Nhóm phương thức xử lý (tiếp tục)

+ ...

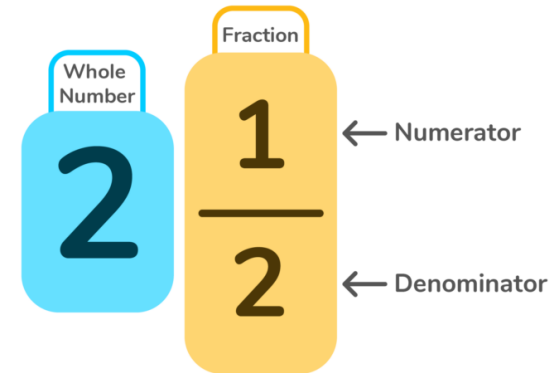
+ Toán tử cộng cộng (trái).

+ Toán tử trừ trừ (trái).

+ Toán tử cộng cộng (phải).

+ Toán tử trừ trừ (phải).

+ ...



# Thiết kế lớp đối tượng hỗn số CHonSo

– Nhóm phương thức xử lý (tiếp tục)

+ ...

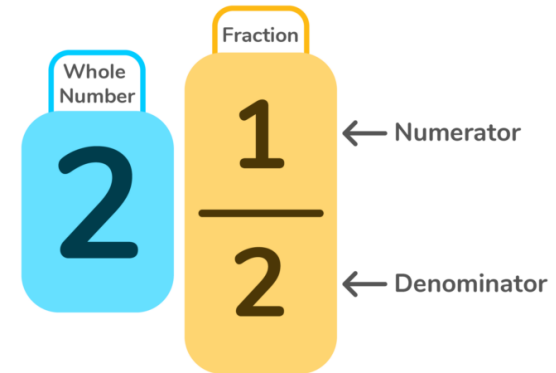
+ Ước chung lớn nhất tử và mẫu.

+ Rút gọn phân số.

+ Rút gọn hỗn số.

+ Xét dấu hỗn số.

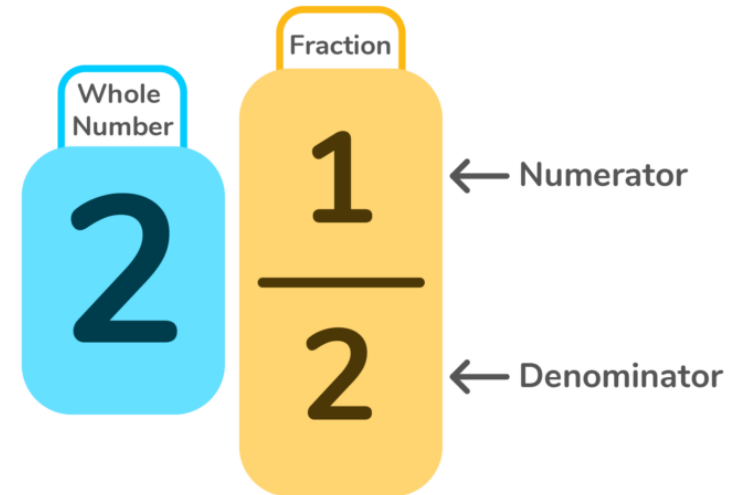
+ Kết thúc nhóm phương thức xử lý.



# KHAI BÁO LỚP

# Thiết kế lớp CHonSo

- Thuộc tính
  - + Phần Nguyên.
  - + Tử số.
  - + Mẫu số.



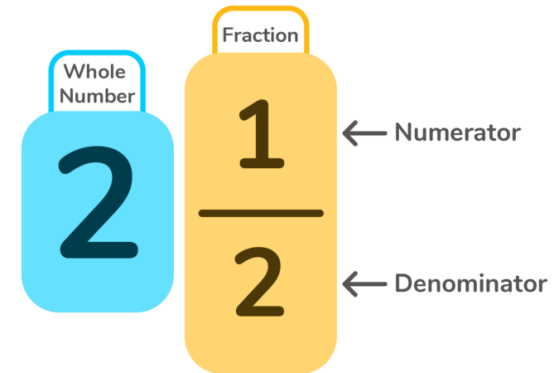
# Khai báo lớp CHonSo

```
11.class CHonSo
12.{
13.    private:
14.        int Nguyen;
15.        int Tu;
16.        int Mau;
17.    public:
18.        ///// Nhóm phương thức khởi tạo
19.        ...
```



# Thiết kế lớp CHonSo

- Nhóm phương thức khởi tạo
  - + Phương thức Nhập.
  - + Toán tử vào (operator >>).
  - + ...



# Khai báo lớp CHonSo

```
11. ...  
12. public:  
13.     ///// Nhóm phương thức khởi tạo  
14.     ///// Chương 03 - Lập Trình HĐT với C++  
15.     void Nhap();  
16.     ///// Chương 04 - Iostream Cơ Bản  
17.     friend istream& operator >>(istream&  
18.                                     CHonSo&);  
19.     ...
```

# Thiết kế lớp CHonSo

– Nhóm phương thức khởi tạo (tiếp tục)

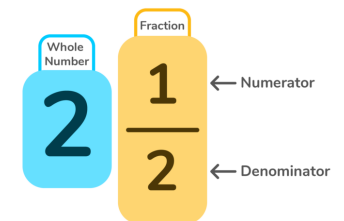
+ ...

+ Phương thức thiết lập mặc định.

+ Phương thức thiết lập sao chép.

+ Phương thức thiết lập khi biết đầy đủ thông tin.

+ ...



# Thiết kế lớp CHonSo

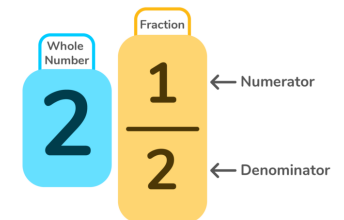
– Nhóm phương thức khởi tạo (tiếp tục)

+ ...

+ Phương thức thiết lập khi biết tử, biết mẫu, phần nguyên lấy mặc định.

+ Phương thức thiết lập khi biết phần nguyên, tử lấy mặc định, mẫu lấy mặc định.

+ ...



# Khai báo lớp CHonSo

```
11.  // /// Nhóm phương thức khởi tạo
12.  // /// Chương 05 - Phương thức thiết lập -
13.  // /// Phương thức phá hủy
14.  ...
15.  CHonSo();
16.  CHonSo(const CHonSo&);
17.  CHonSo(int, int, int);
18.  CHonSo(int, int);
19.  CHonSo(int);
```

# Thiết kế lớp CHonSo

– Nhóm phương thức khởi tạo (tiếp tục)

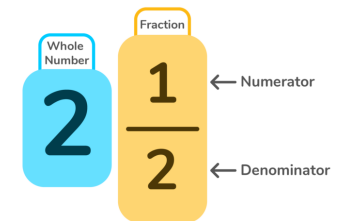
+ ...

+ Phương thức khởi tạo mặc định.

+ Phương thức khởi tạo sao chép.

+ Phương thức khởi tạo khi biết đầy đủ thông tin.

+ ...



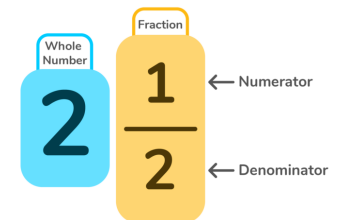
# Thiết kế lớp CHonSo

– Nhóm phương thức khởi tạo (tiếp tục)

+ ...

+ Phương thức khởi tạo khi biết tử, biết mẫu, phần nguyên lấy mặc định.

+ Phương thức khởi tạo khi biết phần nguyên, tử lấy mặc định, mẫu lấy mặc định.



+ Kết thúc nhóm phương thức khởi tạo



# Khai báo lớp CHonSo

```
11.  ///// Nhóm phương thức khởi tạo
12.  ///// Chương 09 - Thiết kế lớp
13.  ...
14.  void KhoiTao();
15.  void KhoiTao(const CHonSo&);
16.  void KhoiTao(int, int, int);
17.  void KhoiTao(int, int);
18.  void KhoiTao(int);
19.  ///// Nhóm phương thức cung cấp thông tin
```



# Thiết kế lớp CHonSo

– Nhóm phương thức cung cấp thông tin

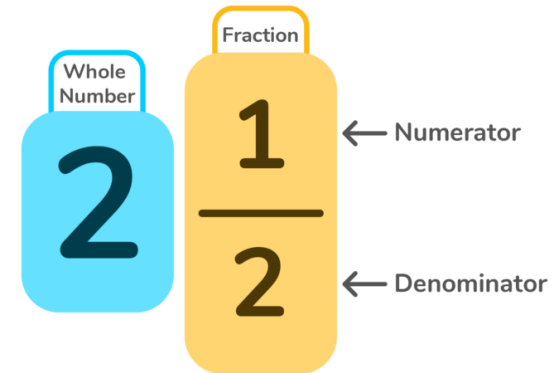
+ Phương thức Xuất.

+ Toán tử ra (operator <<).

+ ...

+ Lớp CHonSo có bao nhiêu thuộc tính.

+ Trả lời: Lớp CHonSo có 3 thuộc tính.



# Thiết kế lớp CHonSo

– Nhóm phương thức cung cấp thông tin (tiếp tục)

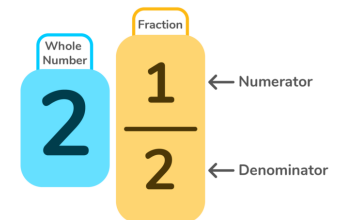
+ ...

+ Phương thức cung cấp phần nguyên.

+ Phương thức cung cấp tử.

+ Phương thức cung cấp mẫu.

+ ...

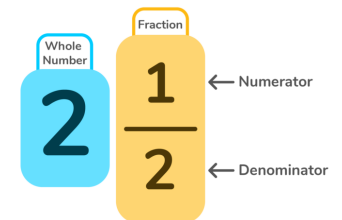


# Thiết kế lớp CHonSo

– Nhóm phương thức cung cấp thông tin (tiếp tục)

+ ...

+ Phương thức cung cấp giá trị thực.



+ Kết thúc nhóm phương thức cung cấp thông tin.

# Khai báo lớp CHonSo

```
11. ...
12. ///// Nhóm phương thức cung cấp thông tin
13. ///// Chương 03 - Lập Trình HĐT với C++
14. void Xuat();
15. ///// Chương 04 - Iostream Cơ Bản
16. friend ostream& operator <<(ostream&,
17.                                CHonSo&);
18. ...
```

# Thiết kế lớp CHonSo

– Nhóm phương thức cung cấp thông tin (tiếp tục)

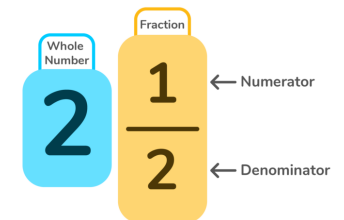
+ ...

+ Phương thức cung cấp phần nguyên.

+ Phương thức cung cấp tử.

+ Phương thức cung cấp mẫu.

+ ...

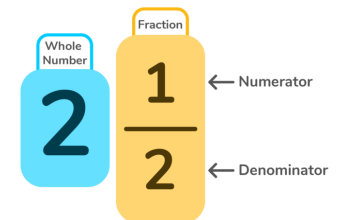


# Thiết kế lớp CHonSo

– Nhóm phương thức cung cấp thông tin (tiếp tục)

+ ...

+ Phương thức cung cấp giá trị thực.



+ Kết thúc nhóm phương thức cung cấp thông tin.

# Khai báo lớp CHonSo

```
11. ...  
12. ///// Nhóm phương thức cung cấp thông tin  
13. ///// Chương 09 - Thiết kế lớp  
14. int getNguyen();  
15. int getTu();  
16. int getMau();  
17. float getGiaTri();  
18. ///// Nhóm phương thức cập nhật thông tin  
19. ...
```

# Thiết kế lớp CHonSo

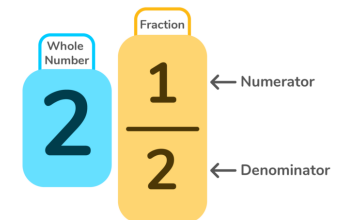
– Nhóm phương thức cập nhật thông tin

+ Toán tử gán (operator =).

+ ...

+ Lớp CHonSo có bao nhiêu thuộc tính.

+ Trả lời: Lớp CHonSo có 3 thuộc tính.





# Thiết kế lớp CHonSo

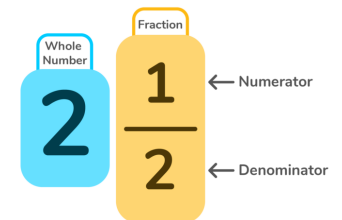
– Nhóm phương thức cập nhật thông tin (tiếp tục)

+ ...

+ Phương thức cập nhật phần nguyên.

+ Phương thức cập nhật tử.

+ Phương thức cập nhật mẫu.



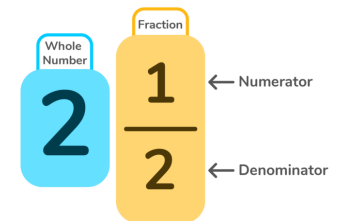
+ Kết thúc nhóm phương thức cung cấp thông tin.

# Khai báo lớp CHonSo

```
11. ...
12. ///// Nhóm phương thức cập nhật thông tin
13. ///// Chương 06 - Toán tử gán
14. CHonSo& operator = (const CHonSo&);
15. ///// Chương 09 - Thiết kế lớp
16. void setNguyen(int);
17. void setTu(int);
18. void setMau(int);
19. ///// Nhóm phương thức kiểm tra
```

# Thiết kế lớp CHonSo

- Nhóm phương thức kiểm tra
  - + Toán tử so sánh bằng.
  - + Toán tử so sánh khác.
  - + Toán tử so sánh lớn hơn.
  - + Toán tử so sánh nhỏ hơn.
  - + Toán tử so sánh lớn hơn bằng.
  - + Toán tử so sánh nhỏ hơn bằng.
  - + ...



# Khai báo lớp CHonSo

```
11. ...
12. ///// Nhóm phương thức kiểm tra
13. ///// Chương 08 - Toán tử so sánh
14. bool operator ==(const CHonSo&);
15. bool operator !=(const CHonSo&);
16. bool operator >(const CHonSo&);
17. bool operator <(const CHonSo&);
18. bool operator >=(const CHonSo&);
19. bool operator <=(const CHonSo&);
```

# Thiết kế lớp CHonSo

## – Nhóm phương thức kiểm tra (tiếp tục)

+ ...

+ Kiểm tra hỗn số có nghĩa hay không?

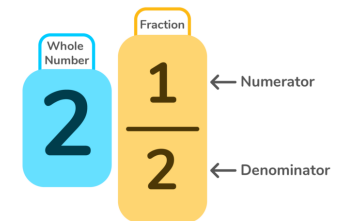
+ Kiểm tra hỗn số có tối giản hay không?

+ Kiểm tra hỗn số có bằng không hay không?

+ Kiểm tra hỗn số dương không?

+ Kiểm tra hỗn số âm không?

+ Kết thúc nhóm phương thức kiểm tra.

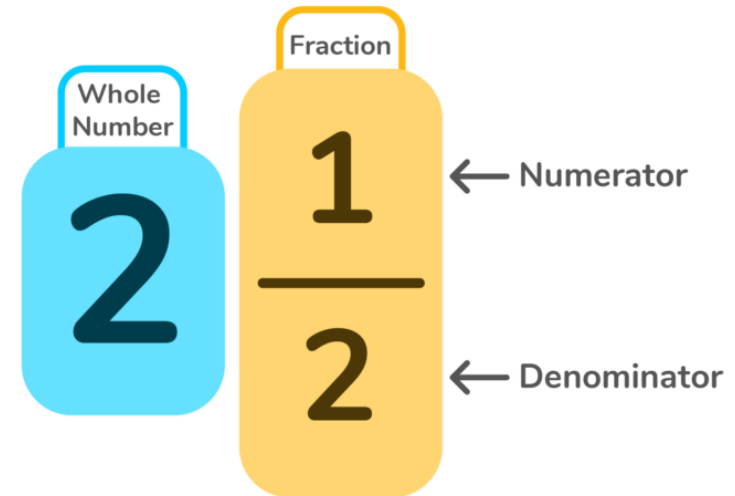


# Khai báo lớp CHonSo

```
11.  ///// Nhóm phương thức kiểm tra
12.  ///// Chương 09 - Thiết kế lớp
13.  ...
14.  bool isCoNghia();
15.  bool isToiGian();
16.  bool isKhong();
17.  bool isDuong();
18.  bool isAm();
19.  ///// Nhóm phương thức xu ly
```

# Thiết kế lớp CHonSo

- Nhóm phương thức xử lý
  - + Phương thức phá hủy.
  - + ...





# Khai báo lớp CHonSo

```
11. ...
12. ///// Nhóm phương thức xử lý
13. ///// Chương 05 - Phương thức thiết lập -
14. ///// Phương thức phá hủy
15. ~CHonSo();
16. ...
17.
```



# Thiết kế lớp CHonSo

## – Nhóm phương thức xử lý (tiếp tục)

+ ...

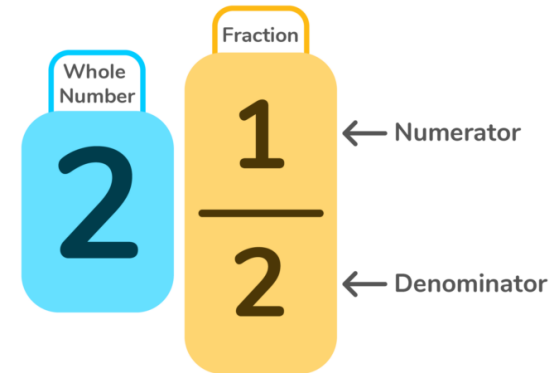
+ Toán tử cộng hai hỗn số.

+ Toán tử trừ hai hỗn số.

+ Toán tử nhân hai hỗn số.

+ Toán tử chia hai hỗn số.

+ ...



# Thiết kế lớp CHonSo

## – Nhóm phương thức xử lý (tiếp tục)

+ ...

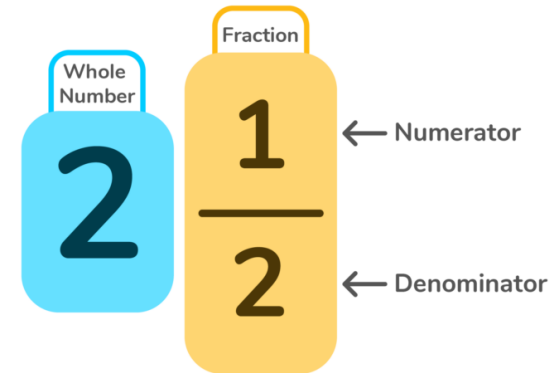
+ Toán tử cộng bằng hai hỗn số.

+ Toán tử trừ bằng hai hỗn số.

+ Toán tử nhân bằng hai hỗn số.

+ Toán tử chia bằng hai hỗn số.

+ ...



# Thiết kế lớp CHonSo

## – Nhóm phương thức xử lý (tiếp tục)

+ ...

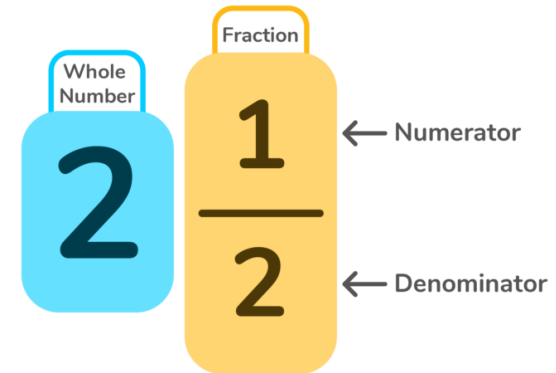
+ Toán tử cộng bằng hai hỗn số.

+ Toán tử trừ bằng hai hỗn số.

+ Toán tử nhân bằng hai hỗn số.

+ Toán tử chia bằng hai hỗn số.

+ ...



# Khai báo lớp CHonSo

```
11.  //... Nhóm phương thức xử lý
12.  //... Chương 07 - Toán tử số học
13.  ...
14.  CHonSo operator + (const CHonSo&);
15.  CHonSo operator - (const CHonSo&);
16.  CHonSo operator * (const CHonSo&);
17.  CHonSo operator / (const CHonSo&);
18.  ...
```

# Khai báo lớp CHonSo

```
11.  //... Nhóm phương thức xử lý
12.  //... Chương 07 - Toán tử số học
13.  ...
14.  CHonSo& operator +=(const CHonSo&);
15.  CHonSo& operator -=(const CHonSo&);
16.  CHonSo& operator *=(const CHonSo&);
17.  CHonSo& operator /=(const CHonSo&);
18.  ...
```

# Thiết kế lớp CHonSo

## – Nhóm phương thức xử lý (tiếp tục)

+ ...

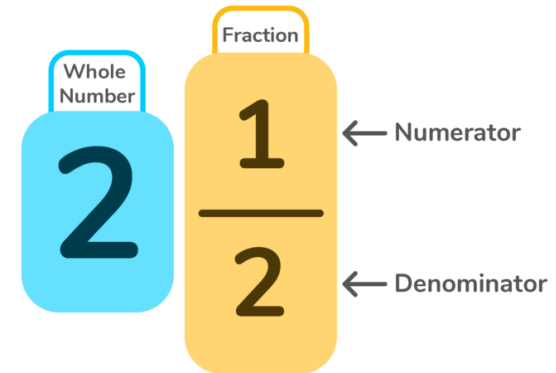
+ Toán tử cộng cộng (trái).

+ Toán tử trừ trừ (trái).

+ Toán tử cộng cộng (phải).

+ Toán tử trừ trừ (phải).

+ ...



# Khai báo lớp CHonSo

```
11.  //... Nhóm phương thức xử lý
12.  //... Chương 07 - Toán tử số học
13.  ...
14.  CHonSo& operator ++();
15.  CHonSo& operator --();
16.  CHonSo& operator ++(int);
17.  CHonSo& operator --(int);
18.  ...
```



# Thiết kế lớp đối tượng hỗn số CHonSo

## – Nhóm phương thức xử lý (tiếp tục)

+ ...

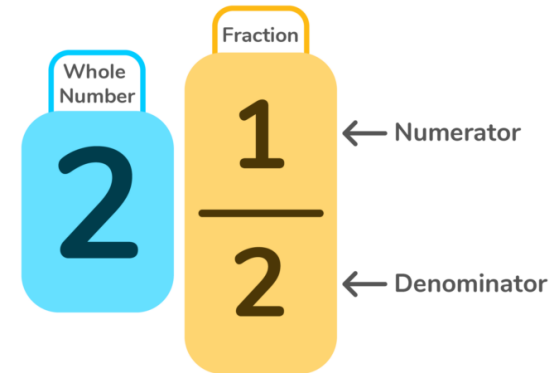
+ Ước chung lớn nhất tử và mẫu.

+ Rút gọn phân số.

+ Rút gọn hỗn số.

+ Xét dấu hỗn số.

+ Kết thúc nhóm phương thức xử lý.





# Khai báo lớp CHonSo

```
11.      ///// Nhóm phương thức xử lý
12.      ///// Chương 09 - Thiết kế lớp
13.      ...
14.      int UCLN();
15.      void RutGonPhanSo();
16.      void RutGon();
17.      int XetDau();
18. };
```

Định nghĩa các phương thức

# **NHÓM PHƯƠNG THỨC KHỞI TẠO**

# Định nghĩa phương thức

```
11. void CHonSo::Nhap()  
12. {  
13.     cout << "Nhap phan nguyen: ";  
14.     cin >> Nguyen;  
15.     cout << "Nhap tu: ";  
16.     cin >> Tu;  
17.     cout << "Nhap mau: ";  
18.     cin >> Mau;  
19. }
```

# Định nghĩa phương thức

```
11.istream& operator>> (istream& is, CHonSo& x)
12.{
13.    cout << "\nNhap phan nguyen: ";
14.    is >> x.Nguyen;
15.    cout << "Nhap tu: ";
16.    is >> x.Tu;
17.    cout << "Nhap mau: ";
18.    is >> x.Mau;
19.    return is;
20.}
```

# Định nghĩa phương thức

```
11. CHonSo :: CHonSo()
```

```
12. {
```

```
13. |   Nguyen = 0;
```

```
14. |   Tu = 0;
```

```
15. |   Mau = 1;
```

```
16. }
```

# Định nghĩa phương thức

```
11. CHonSo::CHonSo(const CHonSo& x)
12. {
13.     Nguyen = x.Nguyen;
14.     Tu = x.Tu;
15.     Mau = x.Mau;
16. }
```

# Định nghĩa phương thức

```
11. CHonSo::CHonSo(int NguyenNguyen, int TuTu,  
12.                int MauMau)  
13. {  
14.     Nguyen = NguyenNguyen;  
15.     Tu = TuTu;  
16.     Mau = MauMau;  
17. }
```

# Định nghĩa phương thức

```
11. CHonSo::CHonSo(int NguyenNguyen, int TuTu)
12. {
13.     Nguyen = NguyenNguyen;
14.     Tu = TuTu;
15.     Mau = 1;
16. }
```



# Định nghĩa phương thức

```
11. CHonSo::CHonSo(int NguyenNguyen)
12. {
13.     Nguyen = NguyenNguyen;
14.     Tu = 0;
15.     Mau = 1;
16. }
```

# Định nghĩa phương thức

```
11. void CHonSo::KhoiTao()  
12. {  
13.     |   Nguyen = 0;  
14.     |   Tu = 0;  
15.     |   Mau = 1;  
16. }
```

# Định nghĩa phương thức

```
11. void CHonSo::KhoiTao(const CHonSo& x)
12. {
13.     Nguyen = x.Nguyen;
14.     Tu = x.Mau;
15.     Mau = x.Mau;
16. }
```

# Định nghĩa phương thức

```
11. void CHonSo::KhoiTao(int NguyenNguyen, int TuTu,  
12.                      int MauMau)  
13. {  
14.     Nguyen = NguyenNguyen;  
15.     Tu = TuTu;  
16.     Mau = MauMau;  
17. }
```

# Định nghĩa phương thức

```
11. void CHonSo::KhoiTao(int NguyenNguyen, int TuTu)
12. {
13.     Nguyen = NguyenNguyen;
14.     Tu = TuTu;
15.     Mau = 1;
16. }
```

# Định nghĩa phương thức

```
11. void CHonSo::KhoiTao(int NguyenNguyen)
12. {
13.     Nguyen = NguyenNguyen;
14.     Tu = 0;
15.     Mau = 1;
16. }
```

Định nghĩa các phương thức  
**NHÓM PHƯƠNG THỨC  
CUNG CẤP THÔNG TIN**

# Định nghĩa phương thức

```
11. void CHonSo::Xuat()  
12. {  
13.     if (Nguyen == 0)  
14.         cout << Tu << "/" << Mau;  
15.     else  
16.         cout << Nguyen << " " << Tu << "/" << Mau;  
17. }
```



# Định nghĩa phương thức

```
11 ostream& operator<< (ostream& os, CHonSo& x)
12 {
13     if (x.Nguyen == 0)
14         os << x.Tu << "/" << x.Mau;
15     else
16         os << x.Nguyen << " " << x.Tu << "/" << x.Mau;
17     return os;
18 }
```

# Định nghĩa phương thức

```
11.int CHonSo::getNguyen()  
12.{  
13.|    return Nguyen;  
14.}
```

# Định nghĩa phương thức

```
11.int CHonSo::getTu()  
12.{  
13.|    return Tu;  
14.}
```

# Định nghĩa phương thức

```
11.int CHonSo::getMau()  
12.{  
13.|    return Mau;  
14.}
```

# Định nghĩa phương thức

```
11.float CHonSo::getGiaTri()  
12.{  
13.|    return Nguyen + (float)Tu / Mau;  
14.}
```

Định nghĩa các phương thức

# **NHÓM PHƯƠNG THỨC CẬP NHẬT THÔNG TIN**

# Định nghĩa phương thức

```
11. CHonSo& CHonSo::operator = (const CHonSo& x)
12. {
13.     Nguyen = x.Nguyen;
14.     Tu = x.Mau;
15.     Mau = x.Mau;
16.     return *this;
17. }
```

# Định nghĩa phương thức

```
11. void CHonSo::setNguyen(int NguyenNguyen)
12. {
13. |   Nguyen = NguyenNguyen;
14. }
```



# Định nghĩa phương thức

```
11. void CHonSo::setTu(int TuTu)
12. {
13. |   Tu = TuTu;
14. }
```

# Định nghĩa phương thức

```
11. void CHonSo::setMau(int MauMau)
12. {
13. |     Mau = MauMau;
14. }
```

Định nghĩa các phương thức

# **NHÓM PHƯƠNG THỨC KIỂM TRA**

# Định nghĩa phương thức

```
11. bool CHonSo::operator == (const CHonSo& x)
12. {
13.     CHonSo temp(x);
14.     float a = getGiaTri();
15.     float b = temp.getGiaTri();
16.     if (a == b)
17.         return true;
18.     return false;
19. }
```

# Định nghĩa phương thức

```
11. bool CHonSo::operator != (const CHonSo& x)
12. {
13.     CHonSo temp(x);
14.     float a = getGiaTri();
15.     float b = temp.getGiaTri();
16.     if (a != b)
17.         return true;
18.     return false;
19. }
```

# Định nghĩa phương thức

```
11. bool CHonSo::operator > (const CHonSo& x)
12. {
13.     CHonSo temp(x);
14.     float a = getGiaTri();
15.     float b = temp.getGiaTri();
16.     if (a > b)
17.         return true;
18.     return false;
19. }
```

# Định nghĩa phương thức

```
11. bool CHonSo::operator < (const CHonSo & x)
12. {
13.     CHonSo temp(x);
14.     float a = getGiaTri();
15.     float b = temp.getGiaTri();
16.     if (a < b)
17.         return true;
18.     return false;
19. }
```

# Định nghĩa phương thức

```
11. bool CHonSo::operator >= (const CHonSo& x)
12. {
13.     CHonSo temp(x);
14.     float a = getGiaTri();
15.     float b = temp.getGiaTri();
16.     if (a >= b)
17.         return true;
18.     return false;
19. }
```



# Định nghĩa phương thức

```
11. bool CHonSo::operator <= (const CHonSo& x)
12. {
13.     CHonSo temp(x);
14.     float a = getGiaTri();
15.     float b = temp.getGiaTri();
16.     if (a <= b)
17.         return true;
18.     return false;
19. }
```

# Định nghĩa phương thức

```
11.bool CHonSo::isCoNghia()  
12.{  
13.    if (Mau != 0)  
14.        return true;  
15.    return false;  
16.}
```

# Định nghĩa phương thức

```
11.bool CHonSo::isToiGian()  
12.{  
13.    if (UCLN() != 1)  
14.        return false;  
15.    if (Tu / Mau >= 1)  
16.        return false;  
17.    return true;  
18.}
```

# Định nghĩa phương thức

```
11. bool CHonSo::isKhong()  
12. {  
13.     if (getGiaTri() == 0)  
14.         return true;  
15.     return false;  
16. }
```

# Định nghĩa phương thức

```
11. bool CHonSo::isDuong()  
12. {  
13.     if (getGiaTri() > 0)  
14.         return true;  
15.     return false;  
16. }
```

# Định nghĩa phương thức

```
11. bool CHonSo::isAm()  
12. {  
13.     if (getGiaTri() < 0)  
14.         return true;  
15.     return false;  
16. }
```

Định nghĩa các phương thức

# **NHÓM PHƯƠNG THỨC XỬ LÝ**

# Định nghĩa phương thức

```
11. CHonSo::~CHonSo()  
12. {  
13. |    return;  
14. }
```



# Định nghĩa phương thức

```
11. CHonSo CHonSo::operator +(const CHonSo& x)
12. {
13.     CHonSo temp;
14.     temp.Nguyen = Nguyen + x.Nguyen;
15.     temp.Tu = Tu * x.Mau + Mau * x.Tu;
16.     temp.Mau = Mau * x.Mau;
17.     temp.RutGon();
18.     return temp;
19. }
```

# Định nghĩa phương thức

```
11. CHonSo CHonSo::operator - (const CHonSo& x)
12. {
13.     CHonSo temp;
14.     temp.Nguyen = Nguyen - x.Nguyen;
15.     temp.Tu = Tu * x.Mau - Mau * x.Tu;
16.     temp.Mau = Mau * x.Mau;
17.     temp.RutGon();
18.     return temp;
19. }
```

# Định nghĩa phương thức

```
11. CHonSo CHonSo::operator* (const CHonSo& x)
12. {
13.     CHonSo temp;
14.     temp.Tu = (Nguyen * Mau + Tu) *
15.         (x.Nguyen * x.Mau + x.Tu);
16.     temp.Mau = Mau * x.Mau;
17.     temp.RutGon();
18.     return temp;
19. }
```

# Định nghĩa phương thức

```
11. CHonSo CHonSo::operator/ (const CHonSo& x)
12. {
13.     CHonSo temp;
14.     temp.Tu = (Nguyen * Mau + Tu) * x.Mau;
15.     temp.Mau = Mau * (x.Nguyen * x.Mau + x.Tu);
16.     temp.RutGon();
17.     return temp;
18. }
```

# Định nghĩa phương thức

```
11. CHonSo& CHonSo::operator+= (const CHonSo& x)
12. {
13.     CHonSo temp;
14.     temp.Nguyen = Nguyen + x.Nguyen;
15.     temp.Tu = Tu * x.Mau + Mau * x.Tu;
16.     temp.Mau = Mau * x.Mau;
17.     temp.RutGon();
18.     *this = temp;
19.     return *this;
20. }
```

# Định nghĩa phương thức

```
11. CHonSo& CHonSo::operator-= (const CHonSo& x)
12. {
13.     CHonSo temp;
14.     temp.Nguyen = Nguyen - x.Nguyen;
15.     temp.Tu = Tu * x.Mau - Mau * x.Tu;
16.     temp.Mau = Mau * x.Mau;
17.     temp.RutGon();
18.     *this = temp;
19.     return *this;
20. }
```

# Định nghĩa phương thức

```
11. CHonSo& CHonSo::operator*= (const CHonSo& x)
12. {
13.     CHonSo temp;
14.     temp.Tu = (Nguyen * Mau + Tu) *
15.             (x.Nguyen * x.Mau + x.Tu);
16.     temp.Mau = Mau * x.Mau;
17.     temp.RutGon();
18.     *this = temp;
19.     return *this;
20. }
```



# Định nghĩa phương thức

```
11. CHonSo& CHonSo::operator/= (const CHonSo& x)
12. {
13.     CHonSo temp;
14.     temp.Tu = (Nguyen * Mau + Tu) * x.Mau;
15.     temp.Mau = Mau * (x.Nguyen * x.Mau + x.Tu);
16.     temp.RutGon();
17.     *this = temp;
18.     return *this;
19. }
```



# Định nghĩa phương thức

```
11. CHonSo& CHonSo::operator++()
```

```
12. {
```

```
13. |   Nguyen++;
```

```
14. |   return *this;
```

```
15. }
```

# Định nghĩa phương thức

```
11. CHonSo& CHonSo::operator--()
```

```
12. {
```

```
13. |   Nguyen--;
```

```
14. |   return *this;
```

```
15. }
```

# Định nghĩa phương thức

```
11. CHonSo& CHonSo::operator++(int)
12. {
13. |   Nguyen++;
14. |   return *this;
15. }
```

# Định nghĩa phương thức

```
11. CHonSo& CHonSo::operator--(int)
12. {
13.     |   Nguyen--;
14.     |   return *this;
15. }
```

# Định nghĩa phương thức

```
11.int CHonSo::UCLN()  
12.{  
13.    int a = abs(Tu);  
14.    int b = abs(Mau);  
15.    while (a * b != 0)  
16.    {  
17.        if (a > b)  
18.            a = a % b;  
19.        else  
20.            b = b % a;  
21.    }  
22.    return a + b;  
23.}
```

# Định nghĩa phương thức

```
11. void CHonSo::RutGonPhanSo()  
12. {  
13.     int temp = UCLN();  
14.     Tu /= temp;  
15.     Mau /= temp;  
16. }
```

# Định nghĩa phương thức

```
11. void CHonSo::RutGon()  
12. {  
13. |   RutGonPhanSo();  
14. |   Nguyen += Tu / Mau;  
15. |   Tu %= Mau;  
16. }
```

# Định nghĩa phương thức

```
11.int CHonSo::XetDau()  
12.{  
13.    float temp = Nguyen + (float)Tu / Mau;  
14.    if (temp > 0)  
15.        return 1;  
16.    if (temp < 0)  
17.        return -1;  
18.    return 0;  
19.}
```



**Cảm ơn quý vị đã lắng nghe**

**Nhóm tác giả**

**TS. Nguyễn Tấn Trần Minh Khang**