

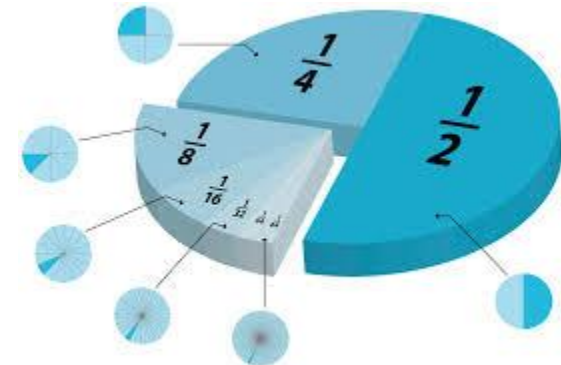
# Chương 9 – THIẾT KẾ LỚP PHÂN SỐ

- 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 PHÂN SỐ

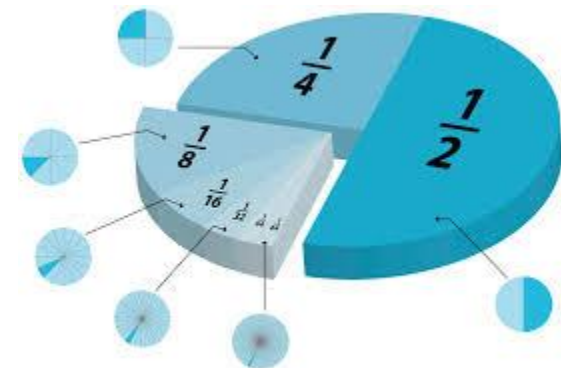
# Thiết kế lớp CPhanSo

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



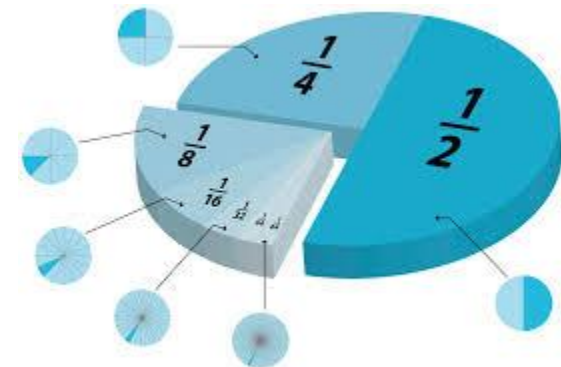
# Thiết kế lớp CPhanSo

- 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 CPhanSo

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



# Thiết kế lớp CPhanSo

## – Nhóm phương thức khởi tạo (tiếp tục)

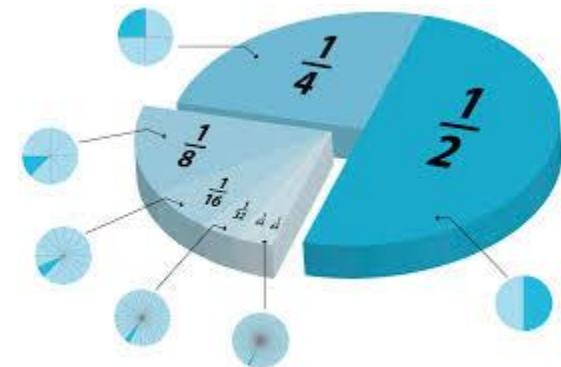
+ ...

+ Phương thức thiết lập mặc định.

+ Phương thức thiết lập sao chép.

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

+ ...



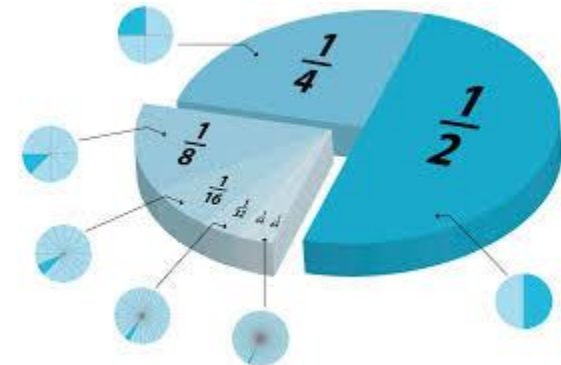
# Thiết kế lớp CPhanSo

– Nhóm phương thức khởi tạo (tiếp tục)

+ ...

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

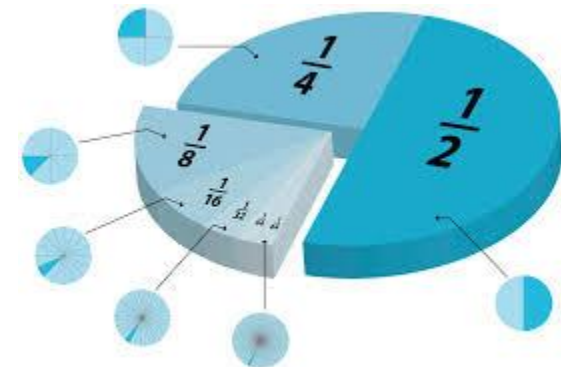
+ ...





# Thiết kế lớp CPhanSo

- 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.
  - + ...



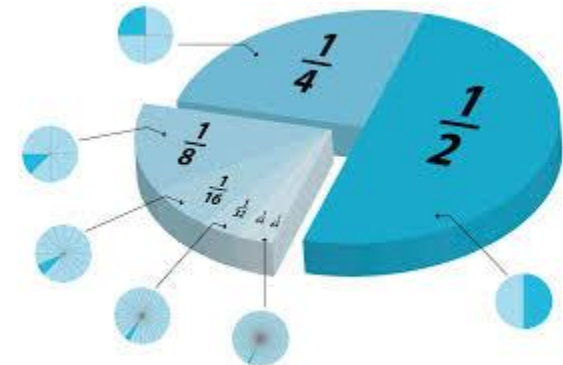


# Thiết kế lớp CPhanSo

– Nhóm phương thức khởi tạo (tiếp tục)

+ ...

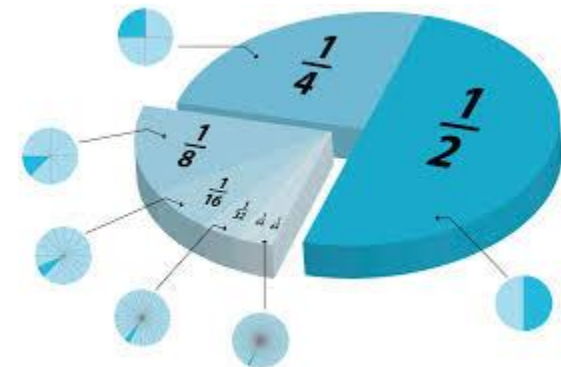
+ Phương thức khởi tạo khi biết tử, mẫu lấy mặc định.



+ Kết thúc nhóm phương thức khởi tạo

# Thiết kế lớp CPhanSo

- Nhóm phương thức cung cấp thông tin
  - + Phương thức Xuất.
  - + Toán tử ra (operator <<).
  - + ...
- + Lớp CPhanSo có bao nhiêu thuộc tính.
- + Trả lời: Lớp CPhanSo có 2 thuộc tính.



# Thiết kế lớp CPhanSo

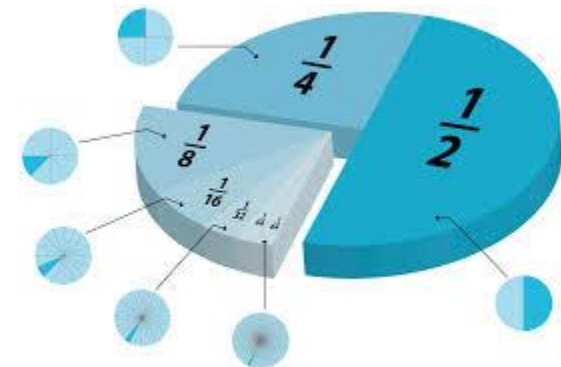
– Nhóm phương thức cung cấp thông tin (tiếp tục)

+ ...

+ Phương thức cung cấp tử.

+ Phương thức cung cấp mẫu.

+ ...



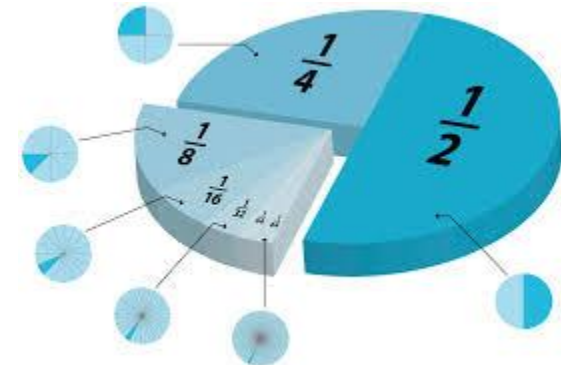
# Thiết kế lớp CPhanSo

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

+ ...

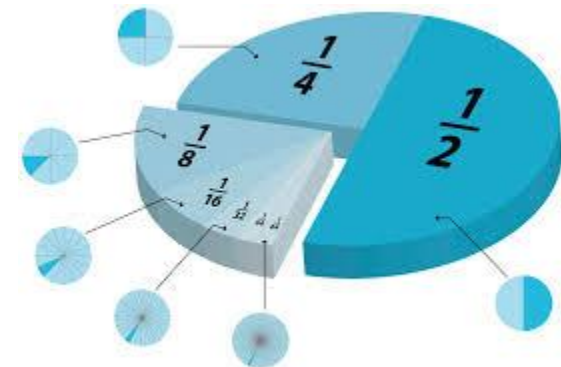
+ Phương thức cung cấp giá trị thực của phân số.

+ Kết thúc nhóm phương thức cung cấp thông tin.



# Thiết kế lớp CPhanSo

- Nhóm phương thức cập nhật thông tin
  - + Toán tử gán (operator `=`).
  - + ...
- + Lớp **CPhanSo** có bao nhiêu thuộc tính.
- + Trả lời: Lớp **CPhanSo** có 2 thuộc tính.



# Thiết kế lớp CPhanSo

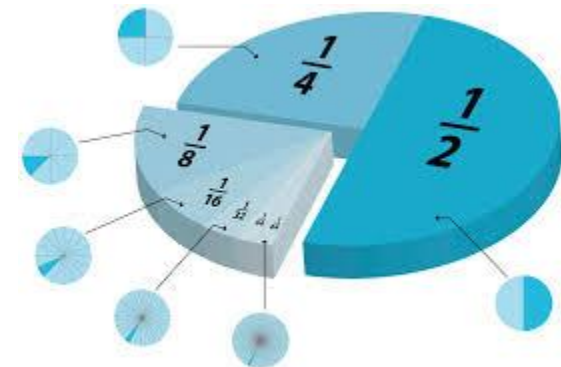
– Nhóm phương thức cập nhật thông tin (tiếp tục)

+ ...

+ Phương thức cập nhật tử.

+ Phương thức cập nhật mẫu.

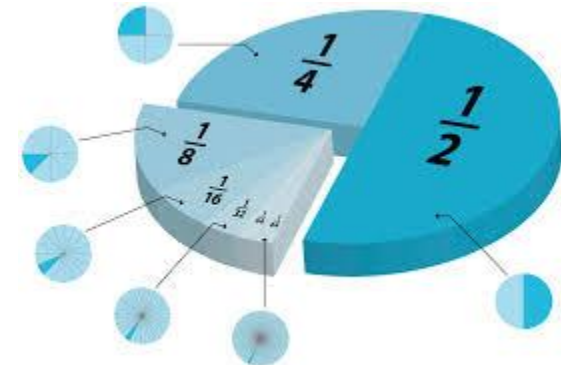
+ Kết thúc nhóm phương thức cung cấp thông tin.





# Thiết kế lớp CPhanSo

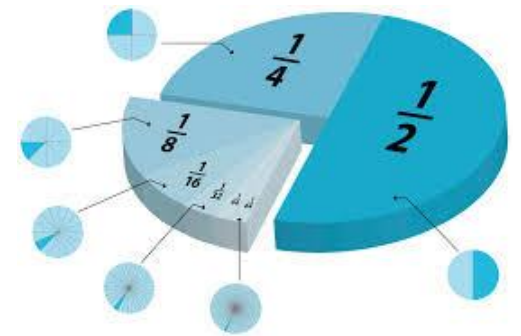
- Nhóm phương thức kiểm tra
  - + Toán tử so sánh bằng.
  - + Toán tử so sánh khác.
  - + Toán tử so sánh lớn hơn.
  - + Toán tử so sánh nhỏ hơn.
  - + Toán tử so sánh lớn hơn bằng.
  - + Toán tử so sánh nhỏ hơn bằng.
  - + ...





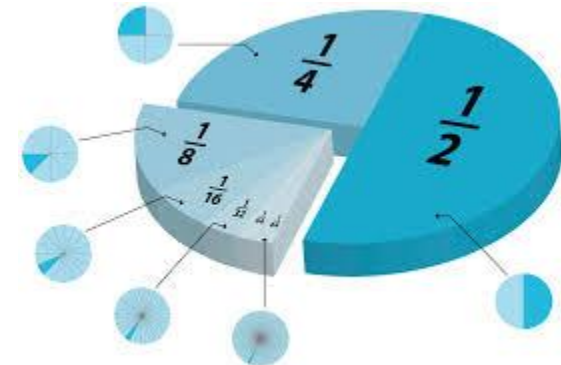
# Thiết kế lớp CPhanSo

- Nhóm phương thức kiểm tra (tiếp tục)
  - + ...
  - + Kiểm tra phân số có nghĩa.
  - + Kiểm tra phân số tối giản.
  - + Kiểm tra phân số dương.
  - + Kiểm tra phân số âm.
  - + Kiểm tra phân số bằng không.
  - + Kết thúc nhóm phương thức kiểm tra.



# Thiết kế lớp CPhanSo

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



# Thiết kế lớp CPhanSo

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

+ ...

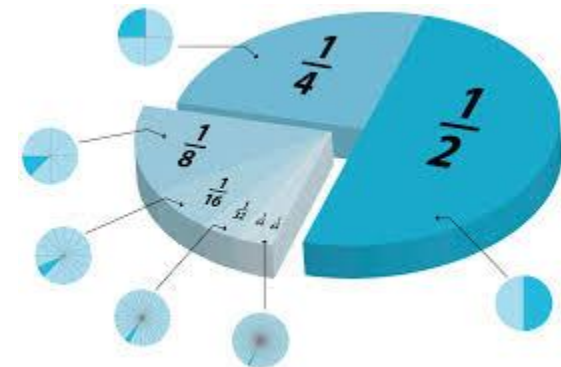
+ Toán tử cộng hai phân số.

+ Toán tử trừ hai phân số.

+ Toán tử nhân hai phân số.

+ Toán tử chia hai phân số.

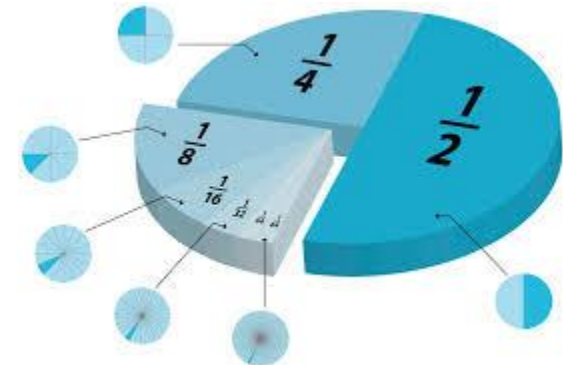
+ ...



# Thiết kế lớp CPhanSo

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

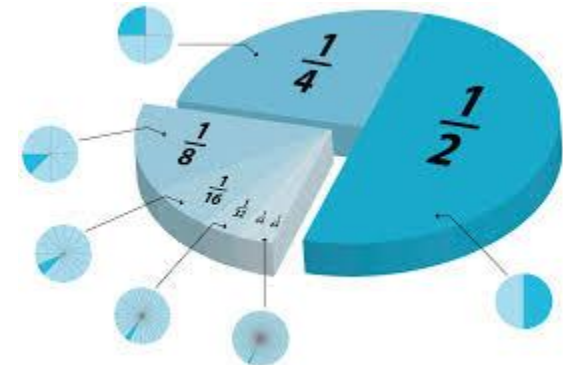
- + ...
- + Toán tử cộng bằng hai phân số.
- + Toán tử trừ bằng hai phân số.
- + Toán tử nhân bằng hai phân số.
- + Toán tử chia bằng hai phân số.
- + ...



# Thiết kế lớp CPhanSo

## – 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 CPhanSo

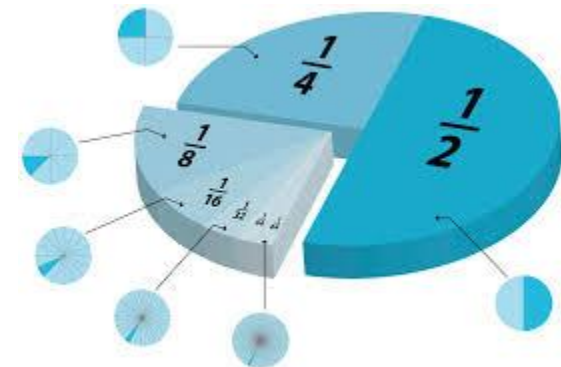
– Nhóm phương thức xử lý (tiếp tục)

+ ...

+ Toán tử lũy thừa phân số.

+ Toán tử lũy thừa bằng phân số.

+ ...

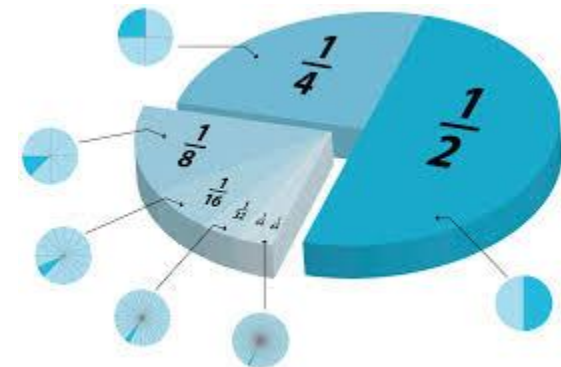




# Thiết kế lớp CPhanSo

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

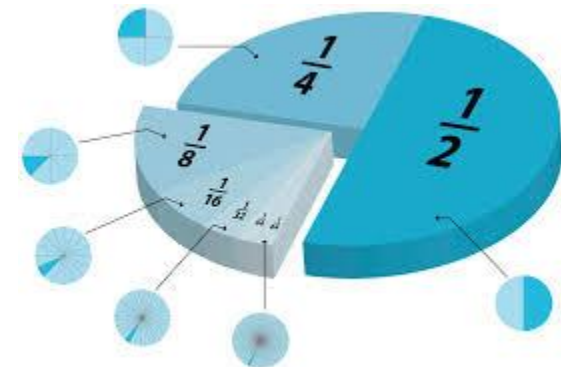
- + ...
- + Rút gọn phân số.
- + Ước chung lớn nhất tử và mẫu.
- + Xét dấu phân số.
- + ...





# Thiết kế lớp CPhanSo

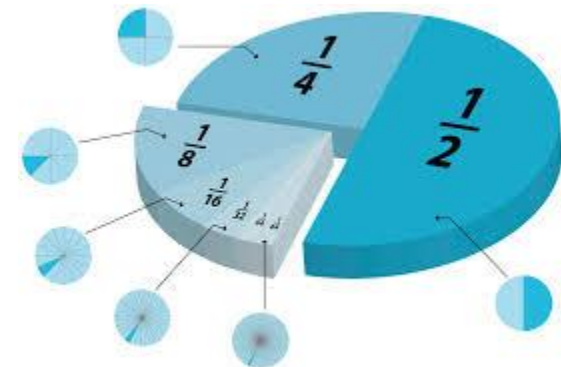
- Nhóm phương thức xử lý (tiếp tục)
  - + ...
  - + Tính tổng hai phân số.
  - + Tính hiệu hai phân số.
  - + Tính tích hai phân số.
  - + Tính thương hai phân số.
  - + Kết thúc nhóm phương thức xử lý.



# KHAI BÁO LỚP CPHANSO

# Thiết kế lớp CPhanSo

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

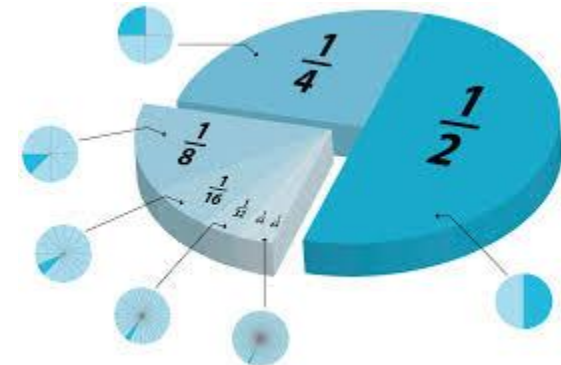


# Khai báo lớp CPhanSo

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

# Thiết kế lớp CPhanSo

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



# Khai báo lớp CPhanSo

```
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.                                     CPhanSo&);  
19.     ...
```

# Thiết kế lớp CPhanSo

## – Nhóm phương thức khởi tạo (tiếp tục)

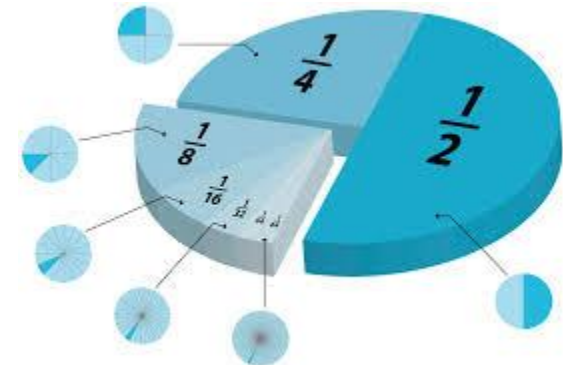
+ ...

+ Phương thức thiết lập mặc định.

+ Phương thức thiết lập sao chép.

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

+ ...





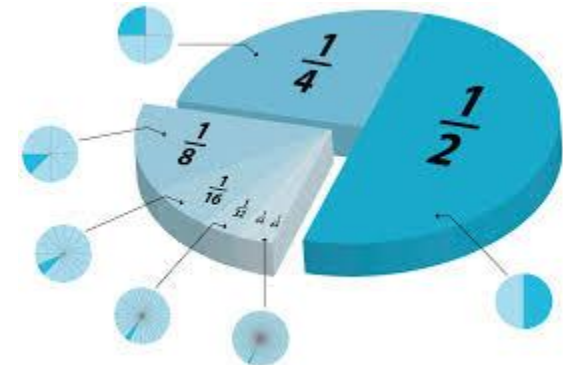
# Thiết kế lớp CPhanSo

– Nhóm phương thức khởi tạo (tiếp tục)

+ ...

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

+ ...

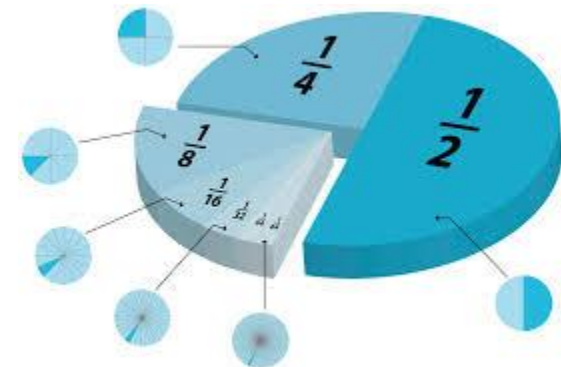


# Khai báo lớp CPhanSo

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

# Thiết kế lớp CPhanSo

- 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.
  - + ...

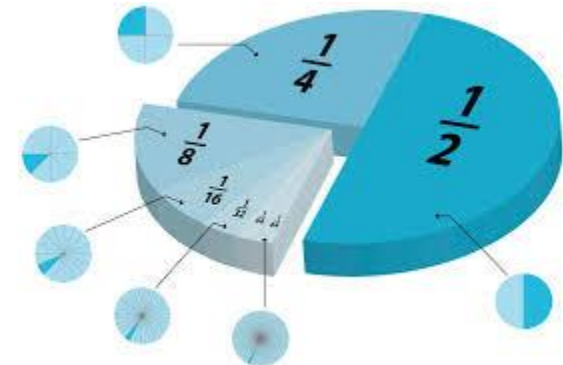


# Thiết kế lớp CPhanSo

– Nhóm phương thức khởi tạo (tiếp tục)

+ ...

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



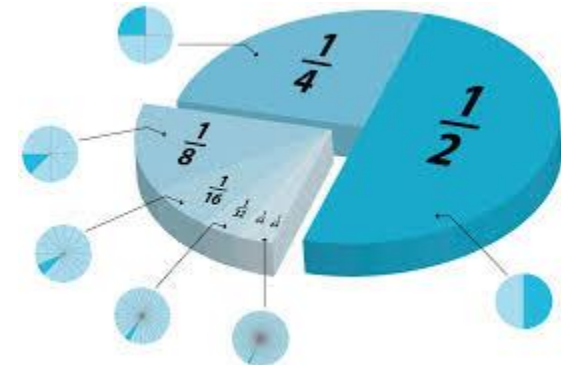
+ Kết thúc nhóm phương thức khởi tạo

# Khai báo lớp CPhanSo

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

# Thiết kế lớp CPhanSo

- Nhóm phương thức cung cấp thông tin
  - + Phương thức Xuất.
  - + Toán tử ra (operator <<).
  - + ...
- + Lớp CPhanSo có bao nhiêu thuộc tính.
- + Trả lời: Lớp CPhanSo có 2 thuộc tính.





# Khai báo lớp CPhanSo

```
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.                                CPhanSo&);
18. ...
```



# Thiết kế lớp CPhanSo

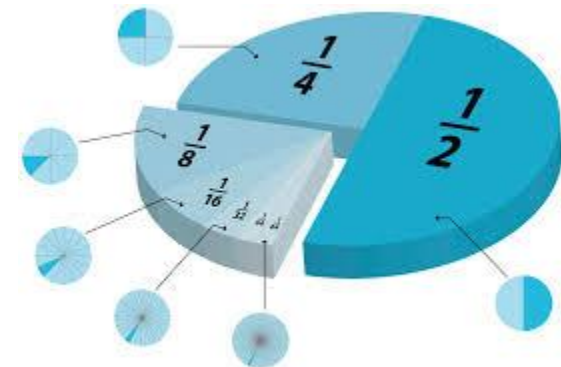
– Nhóm phương thức cung cấp thông tin  
(tiếp tục)

+ ...

+ Phương thức cung cấp tử.

+ Phương thức cung cấp mẫu.

+ ...



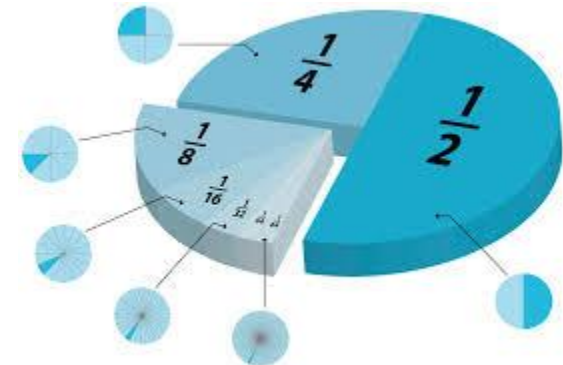
# Thiết kế lớp CPhanSo

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

+ ...

+ Phương thức cung cấp giá trị thực của phân số.

+ Kết thúc nhóm phương thức cung cấp thông tin.

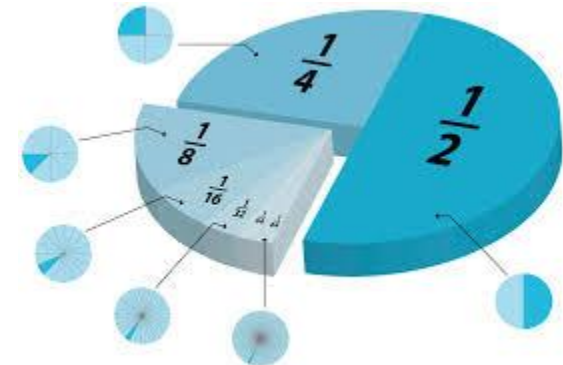


# Khai báo lớp CPhanSo

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

# Thiết kế lớp CPhanSo

- Nhóm phương thức cập nhật thông tin
  - + Toán tử gán (operator `=`).
  - + ...
- + Lớp CPhanSo có bao nhiêu thuộc tính.
- + Trả lời: Lớp CPhanSo có 2 thuộc tính.



# Thiết kế lớp CPhanSo

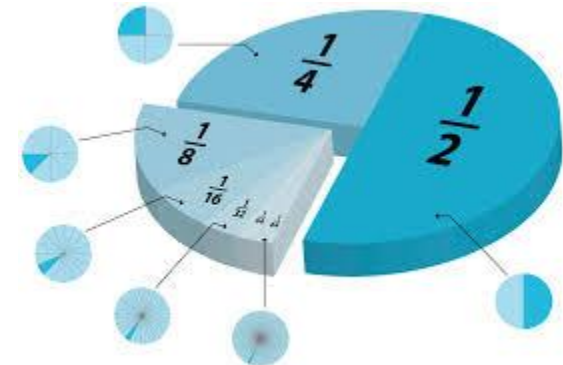
– Nhóm phương thức cập nhật thông tin (tiếp tục)

+ ...

+ Phương thức cập nhật tử.

+ Phương thức cập nhật mẫu.

+ Kết thúc nhóm phương thức cung cấp thông tin.



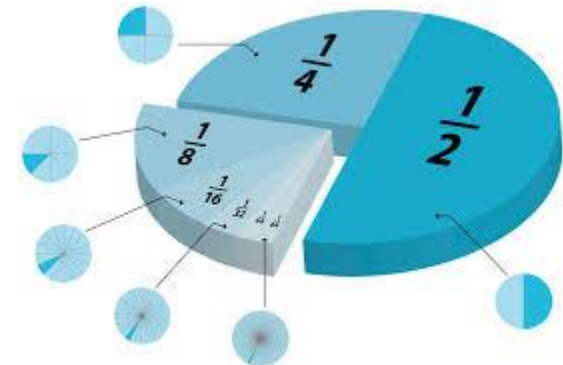
# Khai báo lớp CPhanSo

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



# Thiết kế lớp CPhanSo

- 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 CPhanSo

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

# Thiết kế lớp CPhanSo

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

+ ...

+ Kiểm tra phân số có nghĩa.

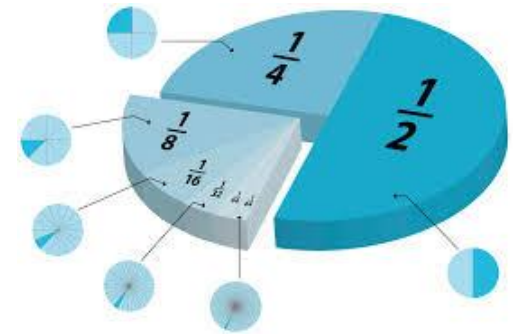
+ Kiểm tra phân số tối giản.

+ Kiểm tra phân số dương.

+ Kiểm tra phân số âm.

+ Kiểm tra phân số bằng không.

+ Kết thúc nhóm phương thức kiểm tra.

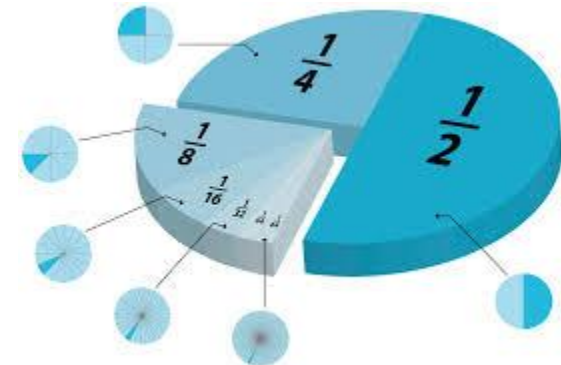


# Khai báo lớp CPhanSo

```
11. ...
12. ///// Nhóm phương thức kiểm tra
13. ///// Chương 09 - Thiết kế lớp
14. bool ktCoNghia();
15. bool ktToiGian();
16. bool ktDuong();
17. bool ktAm();
18. bool ktKhong();
19. ///// Nhóm phương thức xử lý
```

# Thiết kế lớp CPhanSo

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



# Xây dựng lớp CPhanSo

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

# Thiết kế lớp CPhanSo

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

+ ...

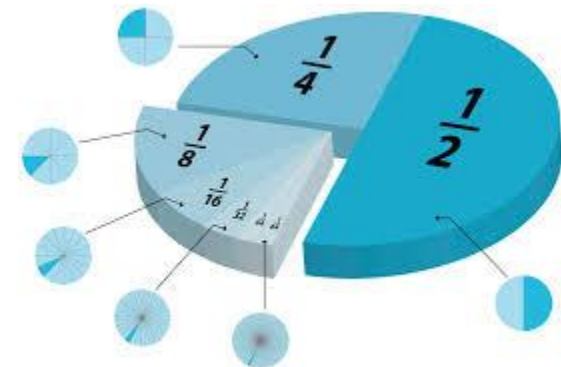
+ Toán tử cộng hai phân số.

+ Toán tử trừ hai phân số.

+ Toán tử nhân hai phân số.

+ Toán tử chia hai phân số.

+ ...

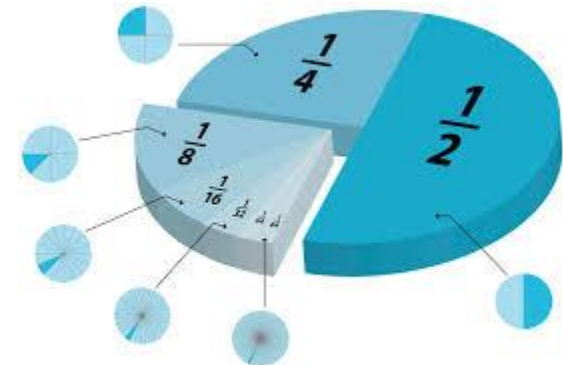




# Thiết kế lớp CPhanSo

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

- + ...
- + Toán tử cộng bằng hai phân số.
- + Toán tử trừ bằng hai phân số.
- + Toán tử nhân bằng hai phân số.
- + Toán tử chia bằng hai phân số.
- + ...





# Khai báo lớp CPhanSo

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

# Khai báo lớp CPhanSo

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

# Thiết kế lớp CPhanSo

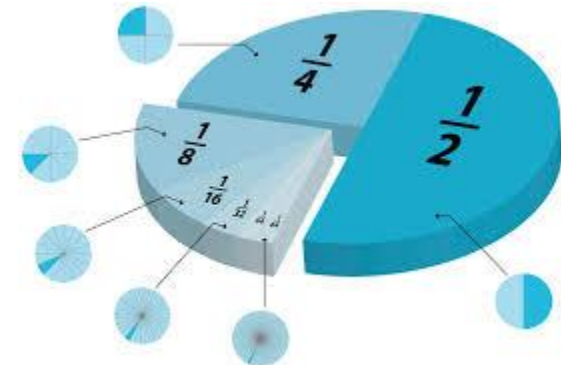
– Nhóm phương thức xử lý (tiếp tục)

+ ...

+ Toán tử lũy thừa phân số.

+ Toán tử lũy thừa bằng phân số.

+ ...



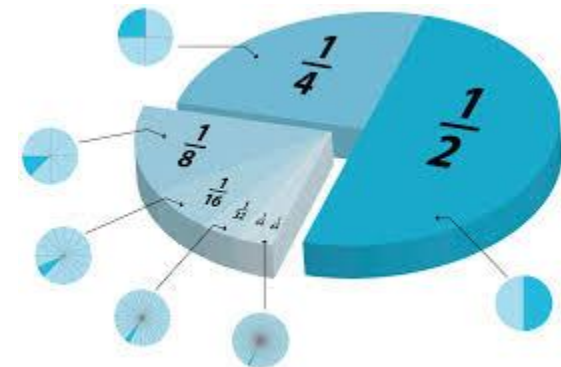
# Khai báo lớp CPhanSo

```
11.  //... Nhóm phương thức xử lý
12.  //... Chương 07 - Toán tử số học
13.  ...
14.  CPhanSo operator^(int);
15.  CPhanSo& operator^=(int);
16.  ...
```

# Thiết kế lớp CPhanSo

## – 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 CPhanSo

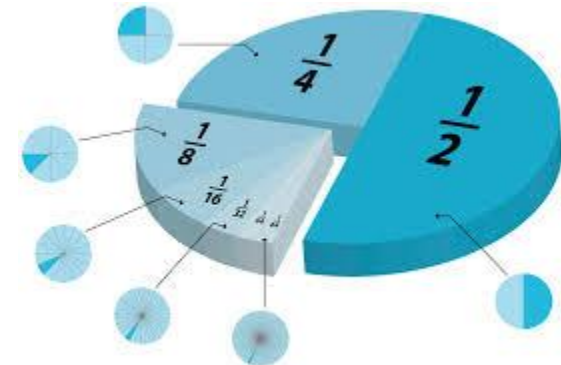
```
11.  //... Nhóm phương thức xử lý
12.  //... Chương 07 - Toán tử số học
13.  ...
14.  CPhanSo& operator++();
15.  CPhanSo& operator--();
16.  CPhanSo& operator++(int);
17.  CPhanSo& operator--(int);
18.  ...
```



# Thiết kế lớp CPhanSo

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

- + ...
- + Rút gọn phân số.
- + Ước chung lớn nhất tử và mẫu.
- + Xét dấu phân số.
- + ...

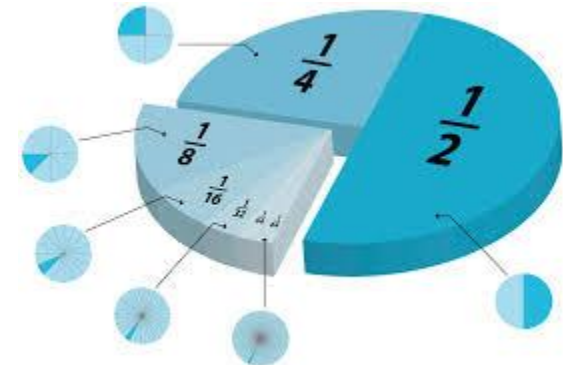


# Khai báo lớp CPhanSo

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

# Thiết kế lớp CPhanSo

- Nhóm phương thức xử lý (tiếp tục)
  - + ...
  - + Tính tổng hai phân số.
  - + Tính hiệu hai phân số.
  - + Tính tích hai phân số.
  - + Tính thương hai phân số.
  - + Kết thúc nhóm phương thức xử lý.



# Khai báo lớp CPhanSo

```
11.  //... Nhóm phương thức xử lý
12.  //... Chương 09 - Thiết kế lớp
13.  ...
14.  CPhanSo Tong(const CPhanSo&);
15.  CPhanSo Hieu(const CPhanSo&);
16.  CPhanSo Tich(const CPhanSo&);
17.  CPhanSo Thuong(const CPhanSo&);
18. };
```

Định nghĩa các phương thức

# **NHÓM PHƯƠNG THỨC KHỞI TẠO**

# Xây dựng lớp CPhanSo

```
11. void CPhanSo::Nhap()  
12. {  
13.     cout << "Nhap tu: ";  
14.     cin >> Tu;  
15.     cout << "Nhap mau: ";  
16.     cin >> Mau;  
17. }
```

18. Slide này làm gì?



# Xây dựng lớp CPhanSo

```
11. void CPhanSo::Nhap()  
12. {  
13.     cout << "Nhap tu: ";  
14.     cin >> Tu;  
15.     cout << "Nhap mau: ";  
16.     cin >> Mau;  
17. }
```

18. Slide này làm gì?

19. Định nghĩa phương thức nhập.

# Xây dựng lớp CPhanSo

```
11.istream& operator>> (istream& is, CPhanSo& x)
12.{
13.    cout << "Nhap tu: ";
14.    is >> x.Tu;
15.    cout << "Nhap mau: ";
16.    is >> x.Mau;
17.    return is;
18.}
19.Slide này làm gì?
```

# Xây dựng lớp CPhanSo

```
11.istream& operator>> (istream& is, CPhanSo& x)
12.{
13.    cout << "Nhap tu: ";
14.    is >> x.Tu;
15.    cout << "Nhap mau: ";
16.    is >> x.Mau;
17.    return is;
18.}
```

19.Định nghĩa hàm toán tử vào.

# Xây dựng lớp CPhanSo

— Hướng dẫn sử dụng phương thức nhập và toán tử vào

```
11. CPhanSo a;  
12. cout << "Nhap phan so thu nhat: " << endl;  
13. a.Nhap();  
14. CPhanSo b;  
15. cout << "Nhap phan so thu hai: " << endl;  
16. cin >> b;  
17. cout << "Phan so thu nhat: " << a << endl;  
18. cout << "Phan so thu hai: " << b << endl;
```

# Xây dựng lớp CPhanSo

11. CPhanSo::CPhanSo()

12. {

13. | Tu = 0;

14. | Mau = 1;

15. }

16. Slide này làm gì?

# Xây dựng lớp CPhanSo

11. CPhanSo::CPhanSo()

12. {

13. | Tu = 0;

14. | Mau = 1;

15. }

16. Slide này làm gì?

17. Định nghĩa phương thức thiết lập mặc định.

# Xây dựng lớp CPhanSo

```
11.CPhanSo::CPhanSo(const CPhanSo& x)
```

```
12.{
```

```
13. |    Tu = x.Tu;
```

```
14. |    Mau = x.Mau;
```

```
15.}
```

```
16.Slide này làm gì?
```



# Xây dựng lớp CPhanSo

11. `CPhanSo::CPhanSo(const CPhanSo& x)`

12. {

13. |     `Tu = x.Tu;`

14. |     `Mau = x.Mau;`

15. }

16. Slide này làm gì?

17. Định nghĩa phương thức thiết lập sao chép.

# Xây dựng lớp CPhanSo

```
11.CPhanSo::CPhanSo(int TuTu, int MauMau)
```

```
12.{
```

```
13. |    Tu = TuTu;
```

```
14. |    Mau = MauMau;
```

```
15.}
```

```
16.Slide này làm gì?
```

# Xây dựng lớp CPhanSo

11. `CPhanSo::CPhanSo(int TuTu, int MauMau)`

12. {

13. |     Tu = TuTu;

14. |     Mau = MauMau;

15. }

16. Slide này làm gì?

17. Định nghĩa phương thức thiết lập khi biết đầy đủ thông tin.

# Xây dựng lớp CPhanSo

```
11.CPhanSo::CPhanSo(int TuTu)
```

```
12.{
```

```
13. |    Tu = TuTu;
```

```
14. |    Mau = 1;
```

```
15.}
```

```
16.Slide này làm gì?
```

# Xây dựng lớp CPhanSo

11. CPhanSo::CPhanSo(int TuTu)

12. {

13. | Tu = TuTu;

14. | Mau = 1;

15. }

16. Slide này làm gì?

17. Định nghĩa phương thức thiết lập khi biết tử, mẫu lấy mặc định.

# Xây dựng lớp CPhanSo

— Hướng dẫn sử dụng phương thức thiết lập

```
11. CPhanSo u;  
12. cout << "Phan so u: " << u << endl;  
13. CPhanSo v(1, 2);  
14. cout << "Phan so v: " << v << endl;  
15. CPhanSo w(v);  
16. cout << "Phan so w: " << w << endl;  
17. CPhanSo x(7);  
18. cout << "Phan so x: " << x << endl;
```

# Xây dựng lớp CPhanSo

```
11. void CPhanSo::KhoiTao()
```

```
12. {
```

```
13. |     Tu = 0;
```

```
14. |     Mau = 1;
```

```
15. }
```

```
16. Slide này làm gì?
```



# Xây dựng lớp CPhanSo

11. `void CPhanSo::KhoiTao()`

12. {

13. |     `Tu = 0;`

14. |     `Mau = 1;`

15. }

16. Slide này làm gì?

17. Định nghĩa phương thức khởi tạo mặc định.

# Xây dựng lớp CPhanSo

```
11. void CPhanSo::KhoiTao(const CPhanSo& x)
```

```
12. {
```

```
13. |     Tu = x.Tu;
```

```
14. |     Mau = x.Mau;
```

```
15. }
```

```
16. Slide này làm gì?
```

# Xây dựng lớp CPhanSo

11. `void CPhanSo::KhoiTao(const CPhanSo& x)`

12. {

13. |     `Tu = x.Tu;`

14. |     `Mau = x.Mau;`

15. }

16. Slide này làm gì?

17. Định nghĩa phương thức khởi tạo sao chép.

# Xây dựng lớp CPhanSo

```
11. void CPhanSo::KhoiTao(int TuTu, int MauMau)
12. {
13.     Tu = TuTu;
14.     Mau = MauMau;
15. }
16. Slide này làm gì?
```

# Xây dựng lớp CPhanSo

```
11. void CPhanSo::KhoiTao(int TuTu, int MauMau)
```

```
12. {
```

```
13. |     Tu = TuTu;
```

```
14. |     Mau = MauMau;
```

```
15. }
```

16. Slide này làm gì?

17. Định nghĩa phương thức khởi tạo khi biết đầy đủ thông tin.

# Xây dựng lớp CPhanSo

```
11. void CPhanSo::KhoiTao(int TuTu)
```

```
12. {
```

```
13. |     Tu = TuTu;
```

```
14. |     Mau = 1;
```

```
15. }
```

```
16. Slide này làm gì?
```

# Xây dựng lớp CPhanSo

```
11. void CPhanSo::KhoiTao(int TuTu)
```

```
12. {
```

```
13. |     Tu = TuTu;
```

```
14. |     Mau = 1;
```

```
15. }
```

16. Slide này làm gì?

17. Định nghĩa phương thức khởi tạo khi biết tử, mẫu lấy mặc định.



# Xây dựng lớp CPhanSo

— Hướng dẫn sử dụng phương thức khởi tạo.

```
11.   CPhanSo u, v, w, x;  
12.   u.KhoiTao();  
13.   v.KhoiTao(1, 2);  
14.   w.KhoiTao(7);  
15.   x.KhoiTao(v);  
  
16.   cout << u <<endl<< v <<endl<< w <<endl<< x;
```

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

# Xây dựng lớp CPhanSo

```
11. void CPhanSo::Xuat()  
12. {  
13. |     cout << Tu << "/" << Mau;  
14. }  
15. Slide này làm gì?
```

# Xây dựng lớp CPhanSo

11. `void CPhanSo::Xuat()`

12. `{`

13. `|     cout << Tu << "/" << Mau;`

14. `}`

15. Slide này làm gì?

16. Định nghĩa phương thức xuất.

# Xây dựng lớp CPhanSo

```
11.ostream& operator<< (ostream& os, CPhanSo& x)
12.{
13.    |   os << x.Tu << "/" << x.Mau;
14.    |   return os;
15.}
16.Slide này làm gì?
```

# Xây dựng lớp CPhanSo

```
11. ostream& operator<< (ostream& os, CPhanSo& x)
```

```
12. {
```

```
13. |   os << x.Tu << "/" << x.Mau;
```

```
14. |   return os;
```

```
15. }
```

16. Slide này làm gì?

17. Định nghĩa hàm toán tử xuất.

# Xây dựng lớp CPhanSo

```
11.int CPhanSo::getTu()
```

```
12.{
```

```
13.|    return Tu;
```

```
14.}
```

```
15.Slide này làm gì?
```



# Xây dựng lớp CPhanSo

11.int CPhanSo::getTu()

12.{

13.|     return Tu;

14.}

15.Slide này làm gì?

16.Định nghĩa phương thức cung cấp thông tin tử.

# Xây dựng lớp CPhanSo

```
11.int CPhanSo::getMau()
```

```
12.{
```

```
13.|    return Mau;
```

```
14.}
```

```
15.Slide này làm gì?
```

# Xây dựng lớp CPhanSo

```
11.int CPhanSo::getMau()
```

```
12.{
```

```
13.|    return Mau;
```

```
14.}
```

15.Slide này làm gì?

16.Định nghĩa phương thức cung cấp thông tin mẫu.

# Xây dựng lớp CPhanSo

```
11.float CPhanSo::getGiaTri()  
12.{  
13.|    return (float)Tu / Mau;  
14.}  
15.Slide này làm gì?
```

# Xây dựng lớp CPhanSo

11. `float CPhanSo::getGiaTri()`

12. {

13. |     `return (float)Tu / Mau;`

14. }

15. Slide này làm gì?

16. Định nghĩa phương thức cung cấp thông tin giá trị thực của phân số.

# Xây dựng lớp CPhanSo

— Hướng dẫn sử dụng phương thức cung cấp thông tin.

```
11. CPhanSo u;  
12. cout << "Phan so thu nhat: ";  
13. cout << u << endl;  
14. u.Xuat();  
  
15. cout << endl << "Tu: " << u.getTu();  
16. cout << endl << "Mau: " << u.getMau();
```

Định nghĩa các phương thức

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



# Xây dựng lớp CPhanSo

```
11.CPhanSo& CPhanSo::operator = (const CPhanSo& x)
```

```
12.{
```

```
13.    Tu = x.Tu;
```

```
14.    Mau = x.Mau;
```

```
15.    return *this;
```

```
16.}
```

```
17.Slide này làm gì?
```

# Xây dựng lớp CPhanSo

11. `CPhanSo& CPhanSo::operator = (const CPhanSo& x)`

12. {

13.     `Tu = x.Tu;`

14.     `Mau = x.Mau;`

15.     `return *this;`

16. }

17. Slide này làm gì?

18. Định nghĩa phương thức toán tử gán.

# Xây dựng lớp CPhanSo

```
11. void CPhanSo::setTu(int TuTu)
```

```
12. {
```

```
13. |     Tu = TuTu;
```

```
14. }
```

```
15. Slide này làm gì?
```

# Xây dựng lớp CPhanSo

11. `void CPhanSo::setTu(int TuTu)`

12. {

13. |     `Tu = TuTu;`

14. }

15. Slide này làm gì?

16. Định nghĩa phương thức cập nhật thông tin tử.

# Xây dựng lớp CPhanSo

11. `void CPhanSo::setMau(int MauMau)`

12. `{`

13. `|     Mau = MauMau;`

14. `}`

15. Slide này làm gì?

# Xây dựng lớp CPhanSo

11. `void CPhanSo::setMau(int MauMau)`

12. {

13. |     Mau = MauMau;

14. }

15. Slide này làm gì?

16. Định nghĩa phương thức cập nhật thông tin mẫu.

# Xây dựng lớp CPhanSo

— Hướng dẫn sử dụng phương thức cập nhật thông tin.

```
11.   CPhanSo u, v;  
12.   cout << "Nhap phan so thu nhat: " << endl;  
13.   cin >> u;  
14.   v = u;  
15.   u.setTu(7);  
16.   u.setMau(6);  
17.   cout << u << endl << v << endl;
```



Định nghĩa các phương thức

# **NHÓM PHƯƠNG THỨC KIỂM TRA**

# Xây dựng lớp CPhanSo

```
11. bool CPhanSo::operator >(const CPhanSo& x)
12. {
13.     CPhanSo temp = *this - x;
14.     return (temp.Tu * temp.Mau > 0);
15. }
16. Slide này làm gì?
```

# Xây dựng lớp CPhanSo

```
11.bool CPhanSo::operator >(const CPhanSo& x)
```

```
12.{
```

```
13.    CPhanSo temp = *this - x;
```

```
14.    return (temp.Tu * temp.Mau > 0);
```

```
15.}
```

16.Slide này làm gì?

17.Định nghĩa phương thức toán tử so sánh lớn hơn.

# Xây dựng lớp CPhanSo

— Hướng dẫn sử dụng phương thức toán tử so sánh lớn hơn.

```
11.     CPhanSo u, v;  
12.     cout << "Nhap phan so thu nhat: " << endl;  
13.     cin >> u;  
14.     cout << "Nhap phan so thu hai: " << endl;  
15.     cin >> v;  
16.     if (u > v)  
17.         cout << u << " lon hon " << v << endl;  
18.     else  
19.         cout << u << " khong lon hon " << v << endl;
```

# Xây dựng lớp CPhanSo

```
11. if (u > v)
12.     cout << u << " lon hon " << v << endl;
13. else
14.     cout << u << " khong lon hon " << v << endl;
15. bool CPhanSo::operator >(const CPhanSo& x)
16. {
17.     CPhanSo temp = *this - x;
18.     return (temp.Tu * temp.Mau > 0);
19. }
```

# Xây dựng lớp CPhanSo

```
11. bool CPhanSo::operator <(const CPhanSo& x)
12. {
13.     CPhanSo temp = *this - x;
14.     return (temp.Tu * temp.Mau < 0);
15. }
16. Slide này làm gì?
```

# Xây dựng lớp CPhanSo

```
11.bool CPhanSo::operator <(const CPhanSo& x)
```

```
12.{
```

```
13.    CPhanSo temp = *this - x;
```

```
14.    return (temp.Tu * temp.Mau < 0);
```

```
15.}
```

16.Slide này làm gì?

17.Định nghĩa phương thức toán tử so sánh nhỏ hơn.



# Xây dựng lớp CPhanSo

```
11.bool CPhanSo::operator >=(const CPhanSo& x)
```

```
12.{
```

```
13.|    return (*this > x || *this == x);
```

```
14.}
```

```
15.Slide này làm gì?
```

# Xây dựng lớp CPhanSo

11. `bool CPhanSo::operator >=(const CPhanSo& x)`

12. `{`

13. `|     return (*this > x || *this == x);`

14. `}`

15. Slide này làm gì?

16. Định nghĩa phương thức toán tử so sánh lớn hơn bằng.

# Xây dựng lớp CPhanSo

11. `bool CPhanSo::operator <=(const CPhanSo& x)`

12. `{`

13. `|     return (*this < x || *this == x);`

14. `}`

15. Slide này làm gì?

# Xây dựng lớp CPhanSo

```
11.bool CPhanSo::operator <=(const CPhanSo& x)
```

```
12.{
```

```
13.|    return (*this < x || *this == x);
```

```
14.}
```

15.Slide này làm gì?

16.Định nghĩa phương thức toán tử so sánh nhỏ hơn bằng.

# Xây dựng lớp CPhanSo

```
11. bool CPhanSo::operator ==(const CPhanSo& x)
12. {
13.     CPhanSo temp = *this - x;
14.     return(temp.Tu == 0);
15. }
16. Slide này làm gì?
```

# Xây dựng lớp CPhanSo

```
11.bool CPhanSo::operator ==(const CPhanSo& x)
```

```
12.{
```

```
13.    CPhanSo temp = *this - x;
```

```
14.    return(temp.Tu == 0);
```

```
15.}
```

16.Slide này làm gì?

17.Định nghĩa phương thức toán tử so sánh bằng.

# Xây dựng lớp CPhanSo

11. `bool CPhanSo::operator !=(const CPhanSo& x)`

12. `{`

13. `|     return !(*this == x);`

14. `}`

15. Slide này làm gì?



# Xây dựng lớp CPhanSo

11. `bool CPhanSo::operator !=(const CPhanSo& x)`

12. `{`

13. `|     return !(*this == x);`

14. `}`

15. Slide này làm gì?

16. Định nghĩa phương thức toán tử so sánh khác.

# Xây dựng lớp CPhanSo

```
11. bool CPhanSo::operator >(const CPhanSo& x)
12. {
13.     CPhanSo temp(x);
14.     if (getGiaTri() > temp.getGiaTri())
15.         return true;
16.     return false;
17. }
```

18. Slide này làm gì?

19. Định nghĩa phương thức toán tử so sánh lớn hơn.

# Xây dựng lớp CPhanSo

```
11. bool CPhanSo::operator <(const CPhanSo& x)
12. {
13.     CPhanSo temp(x);
14.     if (getGiaTri() < temp.getGiaTri())
15.         return true;
16.     return false;
17. }
```

18. Slide này làm gì?

19. Định nghĩa phương thức toán tử so sánh nhỏ hơn.

# Xây dựng lớp CPhanSo

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

18. Slide này làm gì?

19. Định nghĩa phương thức toán tử so sánh lớn hơn bằng.

# Xây dựng lớp CPhanSo

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

18. Slide này làm gì?

19. Định nghĩa phương thức toán tử so sánh nhỏ hơn bằng.

# Xây dựng lớp CPhanSo

```
11. bool CPhanSo::operator ==(const CPhanSo& x)
12. {
13.     CPhanSo temp(x);
14.     if (getGiaTri() == temp.getGiaTri())
15.         return true;
16.     return false;
17. }
```

18. Slide này làm gì?

19. Định nghĩa phương thức toán tử so sánh bằng.

# Xây dựng lớp CPhanSo

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

18. Slide này làm gì?

19. Định nghĩa phương thức toán tử so sánh khác.



# Xây dựng lớp CPhanSo

```
11.bool CPhanSo::ktCoNghia()
```

```
12.{
```

```
13.    if (Mau != 0)
```

```
14.        return true;
```

```
15.    return false;
```

```
16.}
```

```
17.Slide này làm gì?
```

# Xây dựng lớp CPhanSo

```
11.bool CPhanSo::ktCoNghia()
```

```
12.{
```

```
13.    if (Mau != 0)
```

```
14.        return true;
```

```
15.    return false;
```

```
16.}
```

17.Slide này làm gì?

18.Định nghĩa phương thức kiểm tra phân số có nghĩa.

# Xây dựng lớp CPhanSo

```
11.bool CPhanSo::ktToiGian()
```

```
12.{
```

```
13.    if (UCLN() == 1)
```

```
14.        return true;
```

```
15.    return false;
```

```
16.}
```

```
17.Slide này làm gì?
```

# Xây dựng lớp CPhanSo

```
11.bool CPhanSo::ktToiGian()  
12.{  
13.    if (UCLN() == 1)  
14.        return true;  
15.    return false;  
16.}
```

17.Slide này làm gì?

18.Định nghĩa phương thức kiểm tra phân số tối giản.

# Xây dựng lớp CPhanSo

```
11.bool CPhanSo::ktKhong()
```

```
12.{
```

```
13.    if (Tu == 0)
```

```
14.        return true;
```

```
15.        return false;
```

```
16.}
```

```
17.Slide này làm gì?
```

# Xây dựng lớp CPhanSo

11. `bool CPhanSo::ktKhong()`

12. {

13. |     `if (Tu == 0)`

14. |         `return true;`

15. |     `return false;`

16. }

17. Slide này làm gì?

18. Định nghĩa phương thức kiểm tra phân số bằng không.

# Xây dựng lớp CPhanSo

```
11.bool CPhanSo::ktDuong()
```

```
12.{
```

```
13.    if (Tu * Mau > 0)
```

```
14.        return true;
```

```
15.    return false;
```

```
16.}
```

```
17.Slide này làm gì?
```



# Xây dựng lớp CPhanSo

```
11.bool CPhanSo::ktDuong()
```

```
12.{
```

```
13.    if (Tu * Mau > 0)
```

```
14.        return true;
```

```
15.    return false;
```

```
16.}
```

17.Slide này làm gì?

18.Định nghĩa phương thức kiểm tra phân số dương.

# Xây dựng lớp CPhanSo

```
11. bool CPhanSo::ktAm()  
12. {  
13.     if (Tu * Mau < 0)  
14.         return true;  
15.     return false;  
16. }  
17. Slide này làm gì?
```

# Xây dựng lớp CPhanSo

```
11. bool CPhanSo::ktAm()  
12. {  
13.     if (Tu * Mau < 0)  
14.         return true;  
15.     return false;  
16. }
```

17. Slide này làm gì?

18. Định nghĩa phương thức kiểm tra phân số âm.

Định nghĩa các phương thức

# **NHÓM PHƯƠNG THỨC XỬ LÝ**

# Xây dựng lớp CPhanSo

11. CPhanSo::~~CPhanSo()

12. {

13. |     return;

14. }

15. Slide này làm gì?

# Xây dựng lớp CPhanSo

11. CPhanSo::~~CPhanSo()

12. {

13. |     return;

14. }

15. Slide này làm gì?

16. Định nghĩa phương thức phá hủy.

# Xây dựng lớp CPhanSo

```
11.CPhanSo CPhanSo::operator+(const CPhanSo& x)
12.{
13.    CPhanSo temp;
14.    temp.Tu = Tu * x.Mau + Mau * x.Tu;
15.    temp.Mau = Mau * x.Mau;
16.    temp.RutGon();
17.    return temp;
18.}
19.Slide này làm gì?
```



# Xây dựng lớp CPhanSo

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

19.Định nghĩa phương thức toán tử cộng.

# Xây dựng lớp CPhanSo

```
11.CPhanSo CPhanSo::operator-(const CPhanSo& x)
12.{
13.    CPhanSo temp;
14.    temp.Tu = Tu * x.Mau - Mau * x.Tu;
15.    temp.Mau = Mau * x.Mau;
16.    temp.RutGon();
17.    return temp;
18.}
19.Slide này làm gì?
```

# Xây dựng lớp CPhanSo

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

19.Định nghĩa phương thức toán tử trừ.

# Xây dựng lớp CPhanSo

```
11.CPhanSo CPhanSo::operator*(const CPhanSo& x)
12.{
13.    CPhanSo temp;
14.    temp.Tu = Tu * x.Tu;
15.    temp.Mau = Mau * x.Mau;
16.    temp.RutGon();
17.    return temp;
18.}
19.Slide này làm gì?
```

# Xây dựng lớp CPhanSo

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

19.Định nghĩa phương thức toán tử nhân.

# Xây dựng lớp CPhanSo

```
11.CPhanSo CPhanSo::operator/(const CPhanSo& x)
12.{
13.    CPhanSo temp;
14.    temp.Tu = Tu * x.Mau;
15.    temp.Mau = Mau * x.Tu;
16.    temp.RutGon();
17.    return temp;
18.}
19.Slide này làm gì?
```

# Xây dựng lớp CPhanSo

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

19.Định nghĩa phương thức toán tử chia.



# Xây dựng lớp CPhanSo

11. `CPhanSo& CPhanSo::operator+=(const CPhanSo& x)`

12. {

13.     `Tu = Tu * x.Mau + Mau * x.Tu;`

14.     `Mau = Mau * x.Mau;`

15.     `RutGon();`

16.     `return *this;`

17. }

18. Slide này làm gì?

# Xây dựng lớp CPhanSo

11. `CPhanSo& CPhanSo::operator+=(const CPhanSo& x)`

12. {

13.     `Tu = Tu * x.Mau + Mau * x.Tu;`

14.     `Mau = Mau * x.Mau;`

15.     `RutGon();`

16.     `return *this;`

17. }

18. Slide này làm gì?

19. Định nghĩa phương thức toán tử cộng bằng.

# Xây dựng lớp CPhanSo

— Hướng dẫn sử dụng phương thức cập nhật thông tin.

```
11. CPhanSo u, v;  
12. cout << "Nhap phan so thu nhat: " << endl;  
13. cin >> u;  
14. cout << "Nhap phan so thu hai: " << endl;  
15. cin >> v;  
16. CPhanSo w = u += v;  
17. cout << u << endl;  
18. cout << w << endl;
```

# Xây dựng lớp CPhanSo

```
11.CPhanSo w = u += v;
```

```
12.CPhanSo& CPhanSo::operator+=(const CPhanSo& x)
```

```
13.{
```

```
14.    Tu = Tu * x.Mau + Mau * x.Tu;
```

```
15.    Mau = Mau * x.Mau;
```

```
16.    RutGon();
```

```
17.    return *this;
```

```
18.}
```

# Xây dựng lớp CPhanSo

```
11.CPhanSo w = u += v;
```

```
12.CPhanSo::CPhanSo(const CPhanSo& x)
```

```
13.{
```

```
14. | Tu = x.Tu;
```

```
15. | Mau = x.Mau;
```

```
16.}
```

# Xây dựng lớp CPhanSo

11. `CPhanSo& CPhanSo::operator-=(const CPhanSo& x)`

12. {

13.     `Tu = Tu * x.Mau - Mau * x.Tu;`

14.     `Mau = Mau * x.Mau;`

15.     `RutGon();`

16.     `return *this;`

17. }

18. Slide này làm gì?

# Xây dựng lớp CPhanSo

11. `CPhanSo& CPhanSo::operator-=(const CPhanSo& x)`

12. {

13.     `Tu = Tu * x.Mau - Mau * x.Tu;`

14.     `Mau = Mau * x.Mau;`

15.     `RutGon();`

16.     `return *this;`

17. }

18. Slide này làm gì?

19. Định nghĩa phương thức toán tử trừ bằng.



# Xây dựng lớp CPhanSo

11. `CPhanSo& CPhanSo::operator*=(const CPhanSo& x)`

12. {

13.     `Tu = Tu * x.Tu;`

14.     `Mau = Mau * x.Mau;`

15.     `RutGon();`

16.     `return *this;`

17. }

18. Slide này làm gì?

# Xây dựng lớp CPhanSo

11. `CPhanSo& CPhanSo::operator*=(const CPhanSo& x)`

12. {

13.     `Tu = Tu * x.Tu;`

14.     `Mau = Mau * x.Mau;`

15.     `RutGon();`

16.     `return *this;`

17. }

18. Slide này làm gì?

19. Định nghĩa phương thức toán tử nhân bằng.

# Xây dựng lớp CPhanSo

11. `CPhanSo& CPhanSo::operator/=(const CPhanSo& x)`

12. {

13.     `Tu = Tu * x.Mau;`

14.     `Mau = Mau * x.Tu;`

15.     `RutGon();`

16.     `return *this;`

17. }

18. Slide này làm gì?

# Xây dựng lớp CPhanSo

11. `CPhanSo& CPhanSo::operator/=(const CPhanSo& x)`

12. {

13.     `Tu = Tu * x.Mau;`

14.     `Mau = Mau * x.Tu;`

15.     `RutGon();`

16.     `return *this;`

17. }

18. Slide này làm gì?

19. Định nghĩa phương thức toán tử chia bằng.

# Xây dựng lớp CPhanSo

```
11.CPhanSo& CPhanSo::operator^(int n)
12.{
13.    CPhanSo temp(1, 1);
14.    for (int i = 1; i <= n; i++)
15.        temp *= *this;
16.    temp.RutGon();
17.    return temp;
18.}
19.Slide này làm gì?
```

# Xây dựng lớp CPhanSo

```
11.CPhanSo& CPhanSo::operator^(int n)
12.{
13.    CPhanSo temp(1, 1);
14.    for (int i = 1; i <= n; i++)
15.        temp *= *this;
16.    temp.RutGon();
17.    return temp;
18.}
```

19.Định nghĩa phương thức toán tử lũy thừa.

# Xây dựng lớp CPhanSo

```
11.CPhanSo& CPhanSo::operator^=(int n)
```

```
12.{
```

```
13.    CPhanSo temp(1, 1);
```

```
14.    for (int i = 1; i <= n; i++)
```

```
15.        temp *= *this;
```

```
16.    temp.RutGon();
```

```
17.    *this = temp;
```

```
18.    return *this;
```

```
19.}
```

11.Slide  
này làm  
gì?



# Xây dựng lớp CPhanSo

```
11.CPhanSo& CPhanSo::operator^=(int n)
12.{
13.    CPhanSo temp(1, 1);
14.    for (int i = 1; i <= n; i++)
15.        temp *= *this;
16.    temp.RutGon();
17.    *this = temp;
18.    return *this;
19.}
```

11.Slide  
này làm  
gì?

12.Định  
nghĩa  
phương  
thức toán  
tử lũy  
thừa  
bằng.

# Xây dựng lớp CPhanSo

— Hướng dẫn sử dụng phương thức cập nhật thông tin.

```
11.   CPhanSo u, v;  
12.   cout << "Nhap phan so thu nhat: " << endl;  
13.   cin >> u;  
14.   cout << "Nhap phan so thu hai: " << endl;  
15.   cin >> v;  
16.   CPhanSo w = u ^= 5;  
17.   cout << u << endl;  
18.   cout << w << endl;
```

# Xây dựng lớp CPhanSo

```
11.CPhanSo w = u ^= 5;  
12.CPhanSo& CPhanSo::operator^=(int n)  
13.{  
14.    CPhanSo temp(1, 1);  
15.    ...  
16.    *this = temp;  
17.    return *this;  
18.}
```

# Xây dựng lớp CPhanSo

11. CPhanSo w = u ^ = 5;

12. CPhanSo::CPhanSo(const CPhanSo& x)

13. {

14. | Tu = x.Tu;

15. | Mau = x.Mau;

16. }

# Xây dựng lớp CPhanSo

11. CPhanSo& CPhanSo::operator++()

12. {

13. | Tu = Tu + Mau;

14. | return \*this;

15. }

16. Slide này làm gì?

# Xây dựng lớp CPhanSo

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

12. {

13. |     `Tu = Tu + Mau;`

14. |     `return *this;`

15. }

16. Slide này làm gì?

17. Định nghĩa phương thức toán tử cộng cộng trái.

# Xây dựng lớp CPhanSo

11. CPhanSo& CPhanSo::operator--()

12. {

13. | Tu = Tu - Mau;

14. | return \*this;

15. }

16. Slide này làm gì?



# Xây dựng lớp CPhanSo

11. CPhanSo& CPhanSo::operator--()

12. {

13. | Tu = Tu - Mau;

14. | return \*this;

15. }

16. Slide này làm gì?

17. Định nghĩa phương thức toán tử trừ trừ trái.

# Xây dựng lớp CPhanSo

11. CPhanSo& CPhanSo::operator++(int)

12. {

13. | Tu = Tu + Mau;

14. | return \*this;

15. }

16. Slide này làm gì?

# Xây dựng lớp CPhanSo

11. `CPhanSo& CPhanSo::operator++(int)`

12. {

13. |     `Tu = Tu + Mau;`

14. |     `return *this;`

15. }

16. Slide này làm gì?

17. Định nghĩa phương thức toán tử cộng cộng phải.

# Xây dựng lớp CPhanSo

11. CPhanSo& CPhanSo::operator--(int)

12. {

13. | Tu = Tu - Mau;

14. | return \*this;

15. }

16. Slide này làm gì?

# Xây dựng lớp CPhanSo

11. `CPhanSo& CPhanSo::operator--(int)`

12. {

13. |     `Tu = Tu - Mau;`

14. |     `return *this;`

15. }

16. Slide này làm gì?

17. Định nghĩa phương thức toán tử trừ trừ phải.

# Xây dựng lớp CPhanSo

— Hướng dẫn sử dụng phương thức toán tử một ngôi.

```
11. CPhanSo u;  
12. cout << "Nhap phan so thu nhat: " << endl;  
13. cin >> u;  
14. u++; // operator++(int)  
15. cout << u << endl;  
16. ++u; // operator++()  
17. cout << u << endl;
```

# Xây dựng lớp CPhanSo

```
11. void CPhanSo::RutGon()  
12. {  
13.     int ucln = UCLN();  
14.     Tu = Tu / ucln;  
15.     Mau = Mau / ucln;  
16. }
```

17. Slide này làm gì?



# Xây dựng lớp CPhanSo

```
11. void CPhanSo::RutGon()  
12. {  
13.     int ucln = UCLN();  
14.     Tu = Tu / ucln;  
15.     Mau = Mau / ucln;  
16. }
```

17. Slide này làm gì?

18. Định nghĩa phương thức rút gọn phân số.

# Xây dựng lớp CPhanSo

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

11.Slide này làm gì?

# Xây dựng lớp CPhanSo

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

11.Slide này làm gì?

12.Định nghĩa phương thức tìm ước số chung lớn nhất của tử và mẫu.

# Xây dựng lớp CPhanSo

```
11.CPhanSo CPhanSo::Tong(const CPhanSo& x)
12.{
13.    CPhanSo temp;
14.    temp.Tu = Tu * x.Mau + Mau * x.Tu;
15.    temp.Mau = Mau * x.Mau;
16.    temp.RutGon();
17.    return temp;
18.}
19.Slide này làm gì?
```

# Xây dựng lớp CPhanSo

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

19.Định nghĩa phương thức tính tổng hai phân số.

# Xây dựng lớp CPhanSo

```
11.CPhanSo CPhanSo::Hieu(const CPhanSo& x)
12.{
13.    CPhanSo temp;
14.    temp.Tu = Tu * x.Mau - Mau * x.Tu;
15.    temp.Mau = Mau * x.Mau;
16.    temp.RutGon();
17.    return temp;
18.}
19.Slide này làm gì?
```

# Xây dựng lớp CPhanSo

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

19.Định nghĩa phương thức tính hiệu hai phân số.



# Xây dựng lớp CPhanSo

```
11.CPhanSo CPhanSo::Tich(const CPhanSo& x)
12.{
13.    CPhanSo temp;
14.    temp.Tu = Tu * x.Tu;
15.    temp.Mau = Mau * x.Mau;
16.    temp.RutGon();
17.    return temp;
18.}
19.Slide này làm gì?
```

# Xây dựng lớp CPhanSo

```
11.CPhanSo CPhanSo::Tich(const CPhanSo& x)
12.{
13.    CPhanSo temp;
14.    temp.Tu = Tu * x.Tu;
15.    temp.Mau = Mau * x.Mau;
16.    temp.RutGon();
17.    return temp;
18.}
```

19.Định nghĩa phương thức tính tích hai phân số.

# Xây dựng lớp CPhanSo

```
11.CPhanSo CPhanSo::Thuong(const CPhanSo& x)
```

```
12.{
```

```
13.    CPhanSo temp;
```

```
14.    temp.Tu = Tu * x.Mau;
```

```
15.    temp.Mau = Mau * x.Tu;
```

```
16.    temp.RutGon();
```

```
17.    return temp;
```

```
18.}
```

```
19.Slide này làm gì?
```

# Xây dựng lớp CPhanSo

```
11.CPhanSo CPhanSo::Thuong(const CPhanSo& x)
```

```
12.{
```

```
13.    CPhanSo temp;
```

```
14.    temp.Tu = Tu * x.Mau;
```

```
15.    temp.Mau = Mau * x.Tu;
```

```
16.    temp.RutGon();
```

```
17.    return temp;
```

```
18.}
```

19.Định nghĩa phương thức tính thương hai phân số.

**Cảm ơn quý vị đã lắng nghe**

**Nhóm tác giả**

**TS. Nguyễn Tấn Trần Minh Khang**