

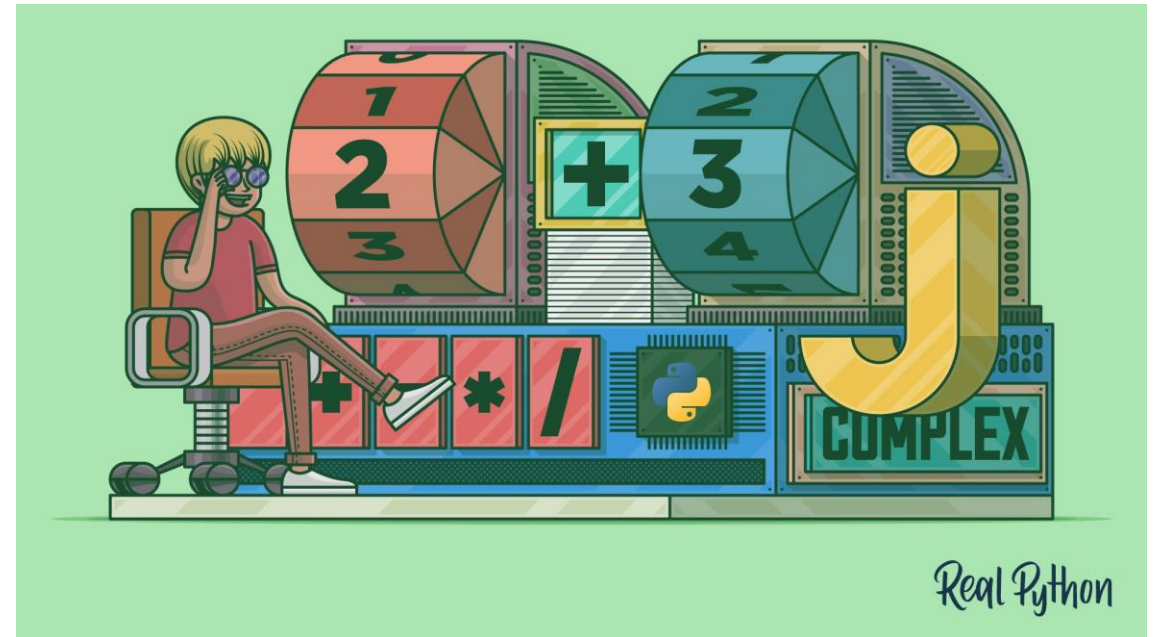
# Chương 9 – THIẾT KẾ LỚP SỐ PHỨC

- 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 SỐ PHỨC

# Thiết kế lớp CSoPhuc

- Thuộc tính
  - + Phần thực.
  - + Phần ảo.



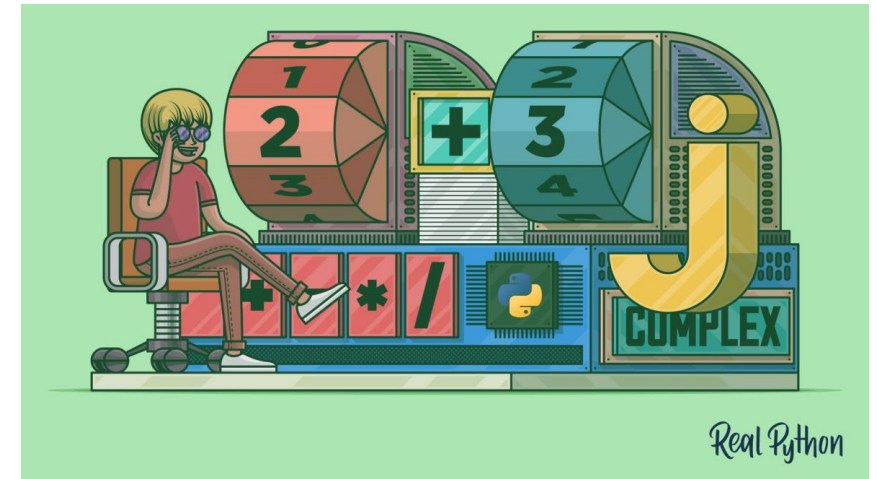
# Thiết kế lớp CSoPhuc

- 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.



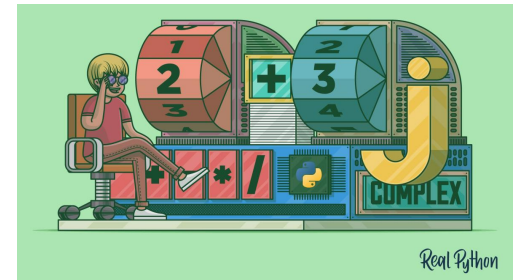
# Thiết kế lớp CSoPhuc

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



# Thiết kế lớp CSoPhuc

- 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.
  - + ...





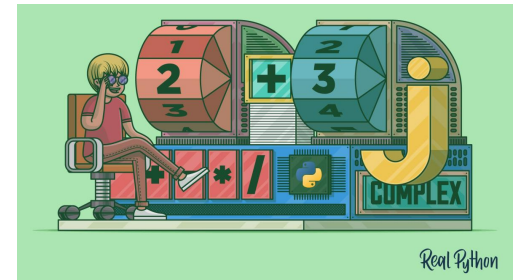
# Thiết kế lớp CSoPhuc

– Nhóm phương thức khởi tạo (tiếp tục)

+ ...

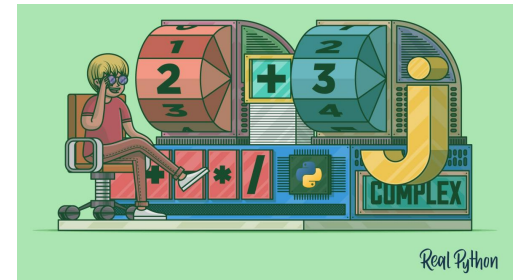
+ Phương thức thiết lập khi biết phần thực, phần ảo lấy mặc định.

+ ...



# Thiết kế lớp CSoPhuc

- 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.
  - + ...



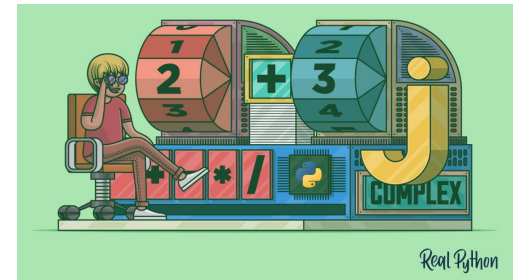


# Thiết kế lớp CSoPhuc

– Nhóm phương thức khởi tạo (tiếp tục)

+ ...

+ Phương thức khởi tạo khi biết phần thực, phần ảo lấy mặc định.



+ Kết thúc nhóm phương thức khởi tạo

# Thiết kế lớp CSoPhuc

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



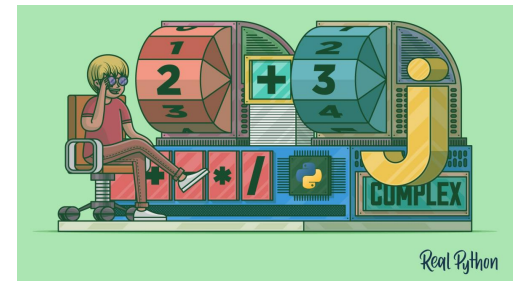
# Thiết kế lớp CSoPhuc

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

+ ...

+ Phương thức cung cấp phần thực.

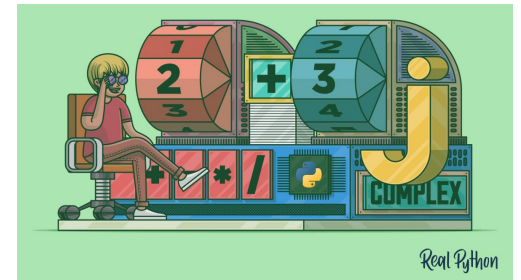
+ Phương thức cung cấp phần ảo.



+ Kết thúc nhóm phương thức cung cấp thông tin.

# Thiết kế lớp CSoPhuc

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



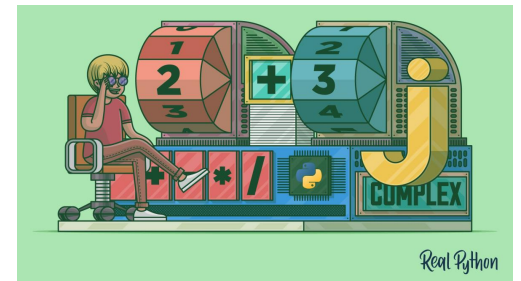
# Thiết kế lớp CSoPhuc

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

+ ...

+ Phương thức cập nhật phần thực.

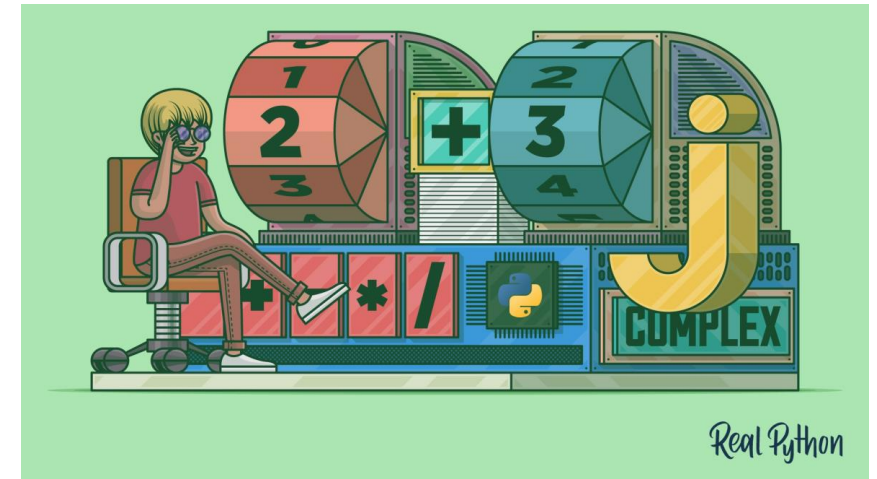
+ Phương thức cập nhật phần ảo.



+ Kết thúc nhóm phương thức cung cấp thông tin.

# Thiết kế lớp CSoPhuc

- 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 CSoPhuc

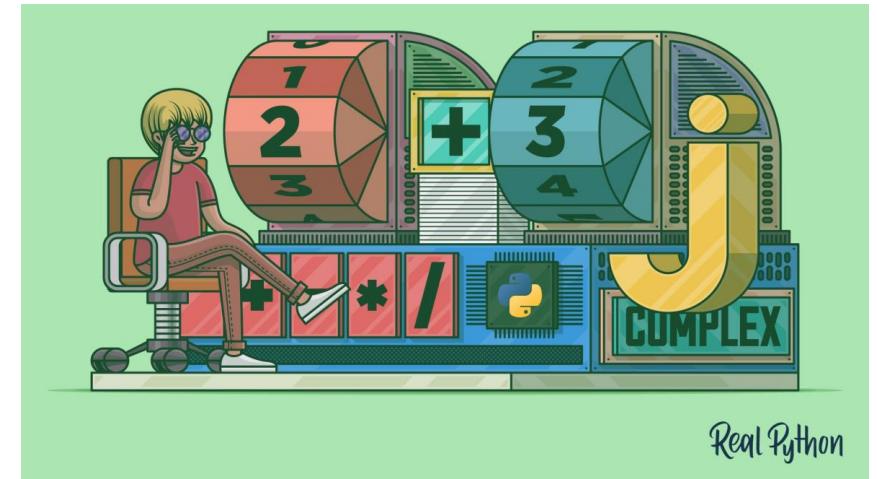
– Nhóm phương thức kiểm tra (tiếp tục)

+ ...

+ Kiểm tra thuần ảo.

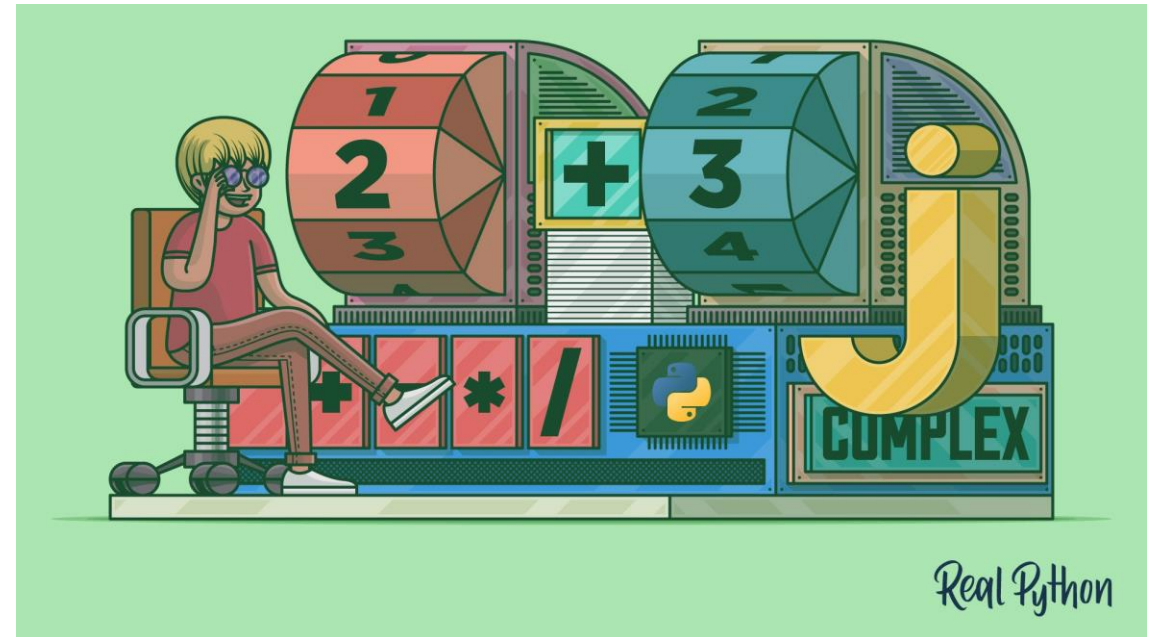
+ Kiểm tra thuần thực.

+ Kết thúc nhóm phương thức kiểm tra.



# Thiết kế lớp CSoPhuc

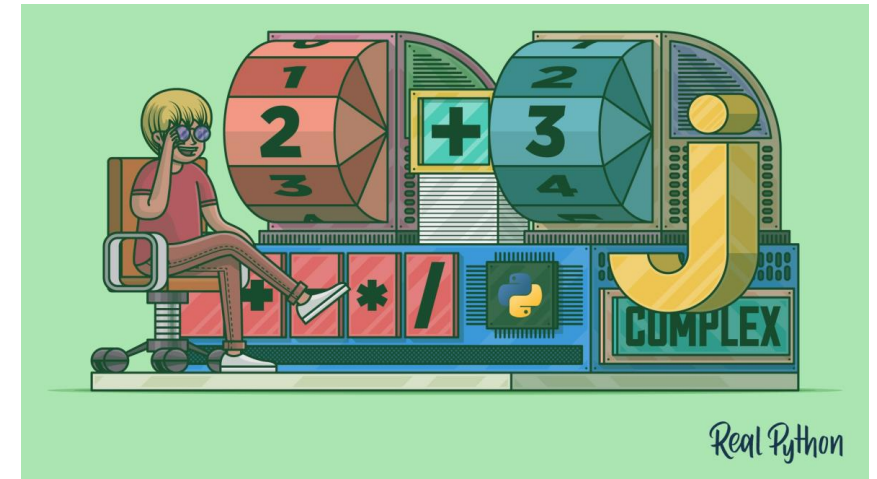
- Nhóm phương thức xử lý
  - + Phương thức phá hủy.
  - + ...



# Thiết kế lớp CSoPhuc

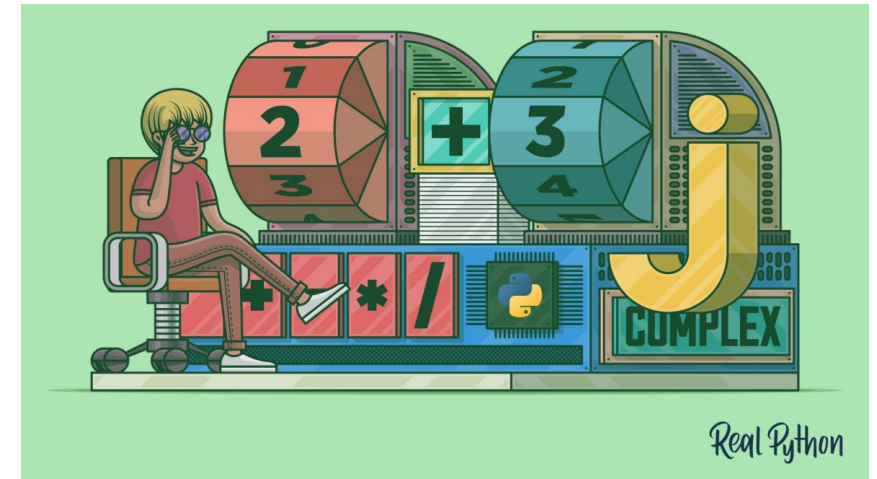
## – Nhóm phương thức xử lý (tiếp tục)

- + ...
- + Toán tử cộng hai số phức.
- + Toán tử trừ hai số phức.
- + Toán tử nhân hai số phức.
- + Toán tử chia hai số phức.
- + ...



# Thiết kế lớp CSoPhuc

- Nhóm phương thức xử lý (tiếp tục)
  - + ...
  - + Toán tử cộng bằng hai số phức.
  - + Toán tử trừ bằng hai số phức.
  - + Toán tử nhân bằng hai số phức.
  - + Toán tử chia bằng hai số phức.
  - + ...





# Thiết kế lớp CSoPhuc

## – 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 CSoPhuc

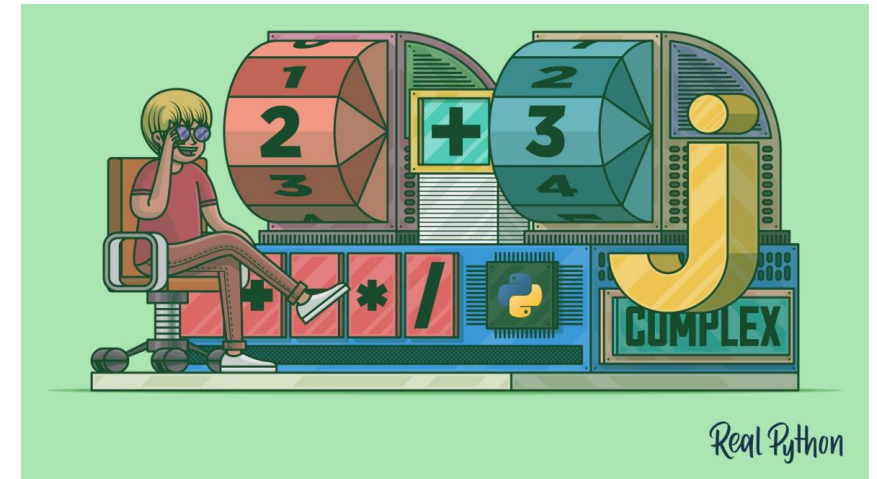
– Nhóm phương thức xử lý (tiếp tục)

+ ...

+ Toán tử lũy thừa số phức.

+ Toán tử lũy thừa bằng số phức.

+ ...





# Thiết kế lớp CSoPhuc

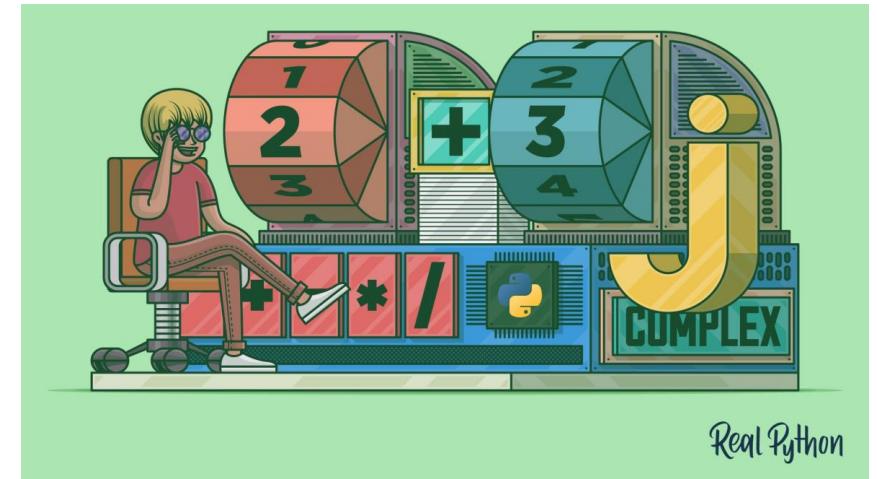
– Nhóm phương thức xử lý (tiếp tục)

+ ...

+ Tính modun số phức.

+ Tìm số phức liên hợp.

+ ...



# Thiết kế lớp CSoPhuc

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

+ ...

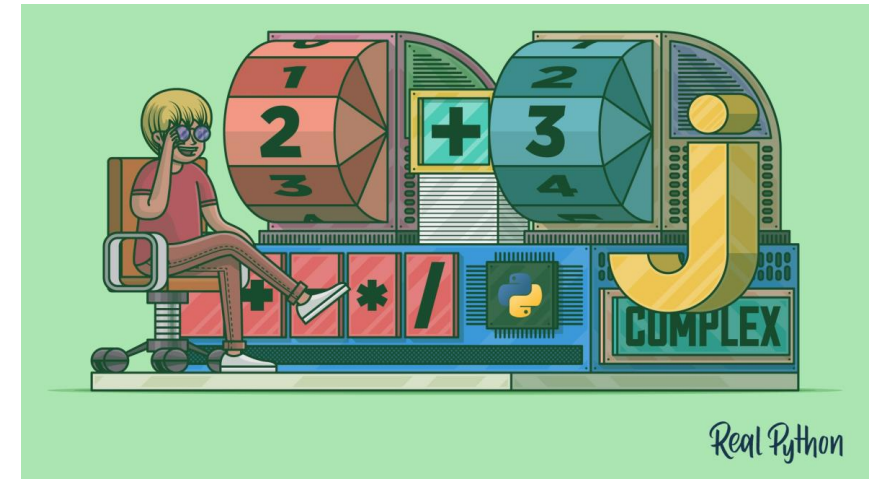
+ Tính tổng hai số phức.

+ Tính hiệu hai số phức.

+ Tính tích hai số phức.

+ Tính thương hai số phức.

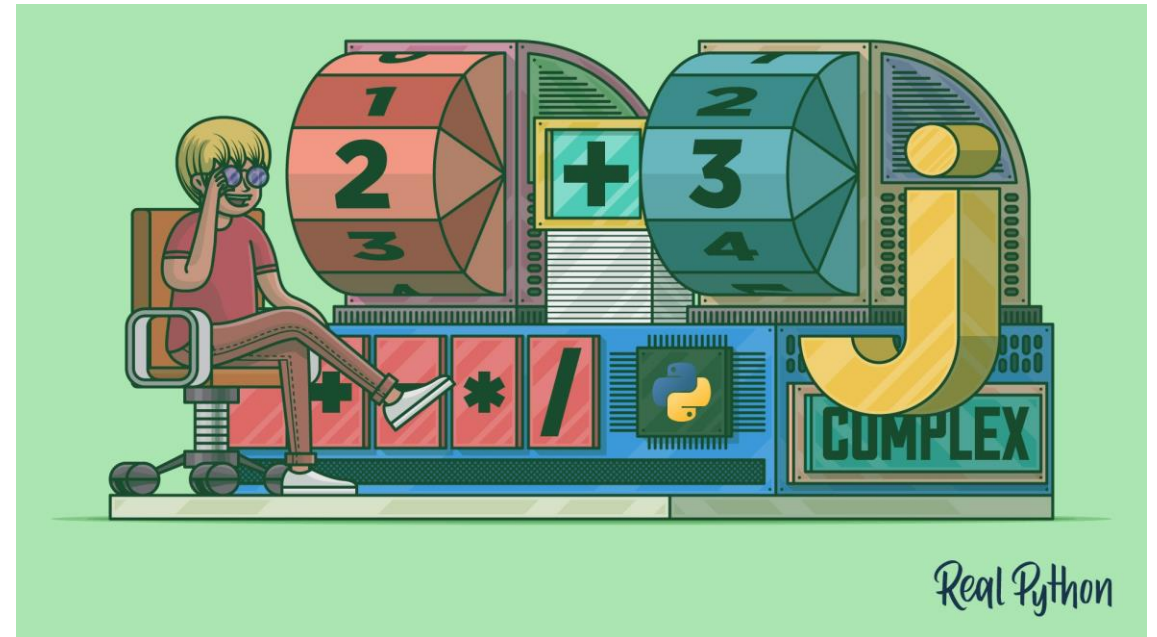
+ Kết thúc nhóm phương thức xử lý.



# KHAI BÁO LỚP

# Thiết kế lớp CSoPhuc

- Thuộc tính
  - + Phần thực.
  - + Phần ảo.



# Khai báo lớp CSoPhuc

```
11.class CSoPhuc
12.{
13.    private:
14.        float Thuc;
15.        float Ao;
16.    public:
17.        ///// Nhóm phương thức khởi tạo
18.        ...
```



# Thiết kế lớp CSoPhuc

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



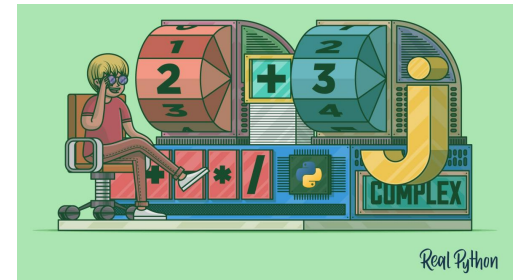


# Khai báo lớp CSoPhuc

```
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.                                         CSoPhuc&);
19.      ...
```

# Thiết kế lớp CSoPhuc

- 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.
  - + ...



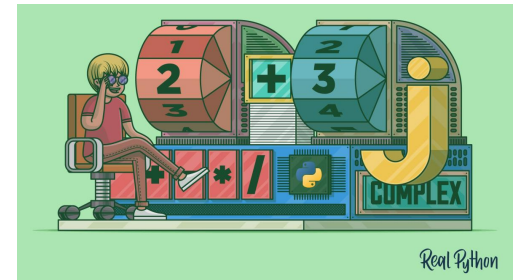
# Thiết kế lớp CSoPhuc

– Nhóm phương thức khởi tạo (tiếp tục)

+ ...

+ Phương thức thiết lập khi biết phần thực, phần ảo lấy mặc định.

+ ...

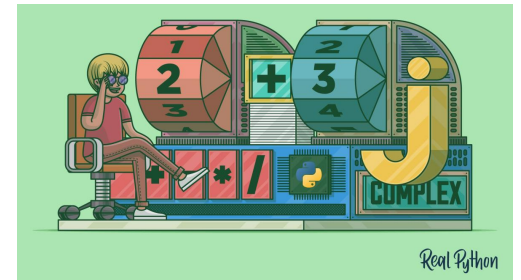


# Khai báo lớp CSoPhuc

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

# Thiết kế lớp CSoPhuc

- 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.
  - + ...

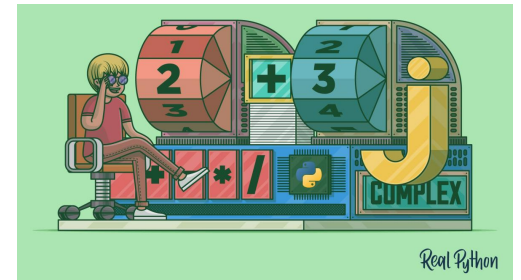


# Thiết kế lớp CSoPhuc

– Nhóm phương thức khởi tạo (tiếp tục)

+ ...

+ Phương thức khởi tạo khi biết phần thực, phần ảo lấy mặc định.



+ Kết thúc nhóm phương thức khởi tạo

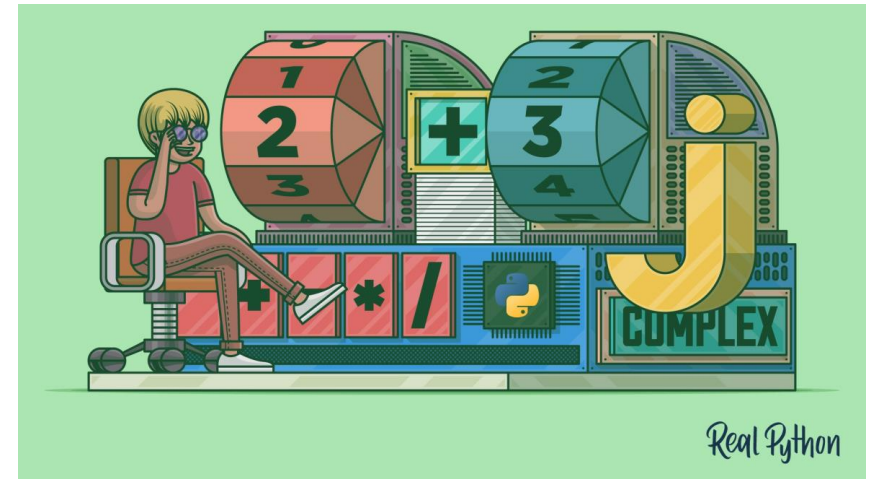


# Khai báo lớp CSoPhuc

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

# Thiết kế lớp CSoPhuc

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



# Khai báo lớp CSoPhuc

```
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.                                CSoPhuc&);
18. ...
```

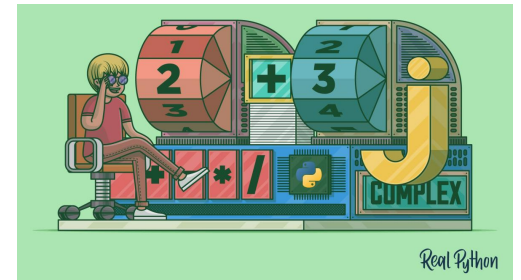
# Thiết kế lớp CSoPhuc

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

+ ...

+ Phương thức cung cấp phần thực.

+ Phương thức cung cấp phần ảo.



+ Kết thúc nhóm phương thức cung cấp thông tin.

# Khai báo lớp CSoPhuc

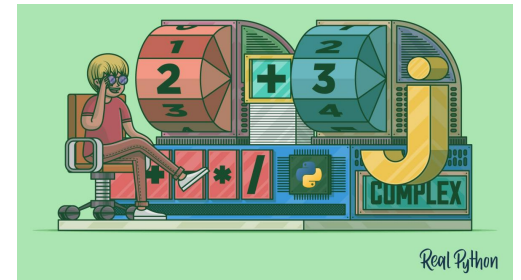
```
11.      ///// Nhóm phương thức cung cấp thông tin
12.      ///// Chương 09 - Thiết kế lớp
13.      float GetThuc();
14.      float GetAo();

15.      ///// Nhóm phương thức cập nhật thông tin
```



# Thiết kế lớp CSoPhuc

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



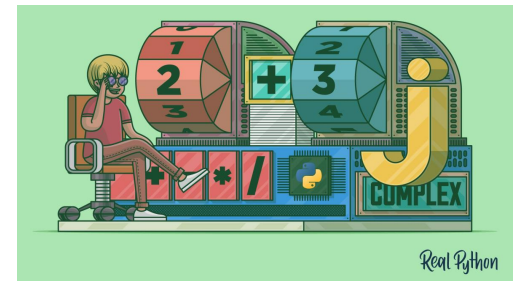
# Thiết kế lớp CSoPhuc

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

+ ...

+ Phương thức cập nhật phần thực.

+ Phương thức cập nhật phần ảo.



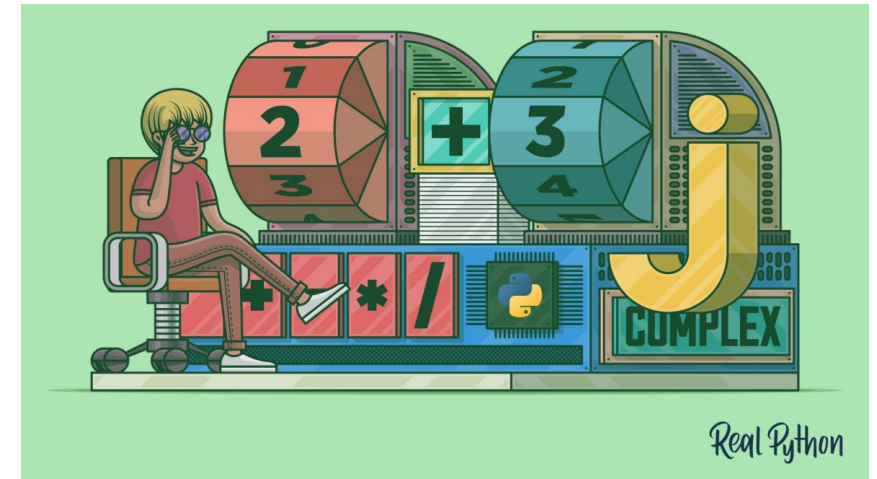
+ Kết thúc nhóm phương thức cung cấp thông tin.

# Khai báo lớp CSoPhuc

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

# Thiết kế lớp CSoPhuc

- 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 CSoPhuc

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

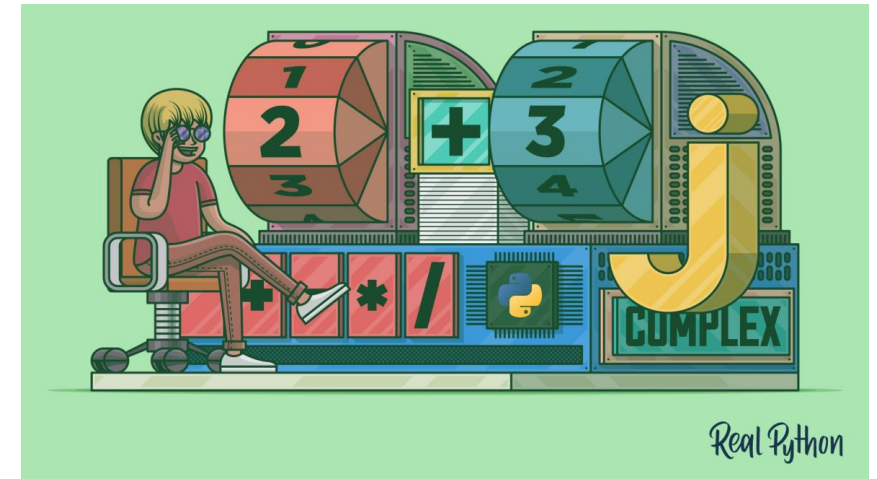
# Thiết kế lớp CSoPhuc

– Nhóm phương thức kiểm tra (tiếp tục)

+ ...

+ Kiểm tra thuần ảo.

+ Kiểm tra thuần thực.



+ Kết thúc nhóm phương thức kiểm tra.



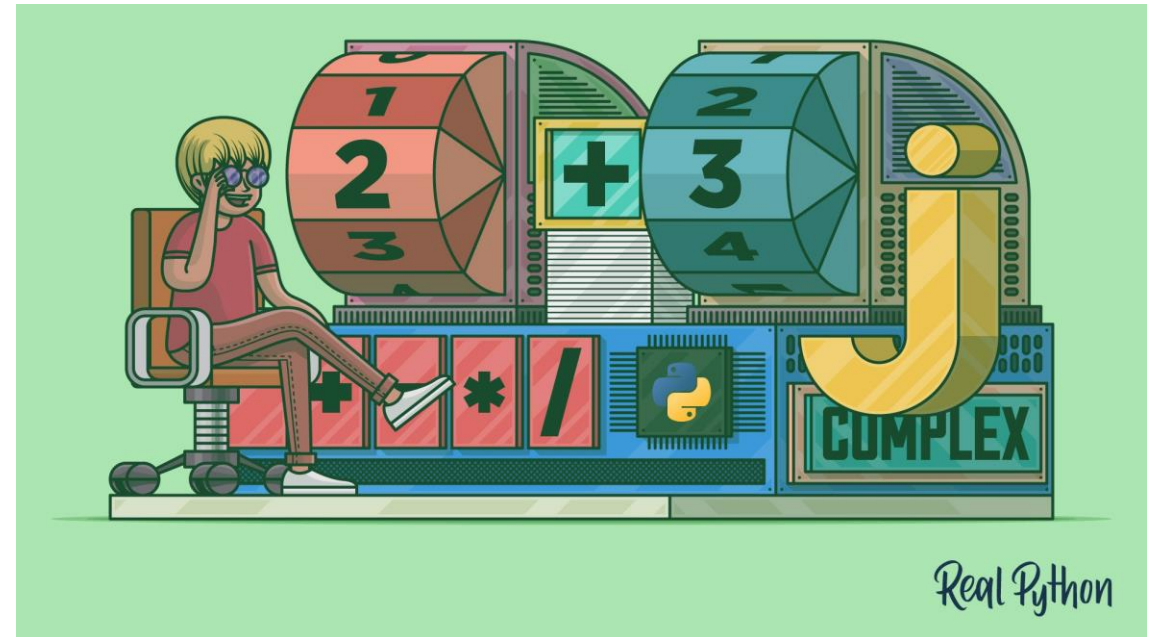
# Khai báo lớp CSoPhuc

```
11.      ///// Nhóm phương thức kiểm tra
12.      ///// Chương 09 - Thiết kế lớp
13.      ...
14.      bool ktThuanAo();
15.      bool ktThuanThuc();

16.      ///// Nhóm phương thức xử lý
```

# Thiết kế lớp CSoPhuc

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



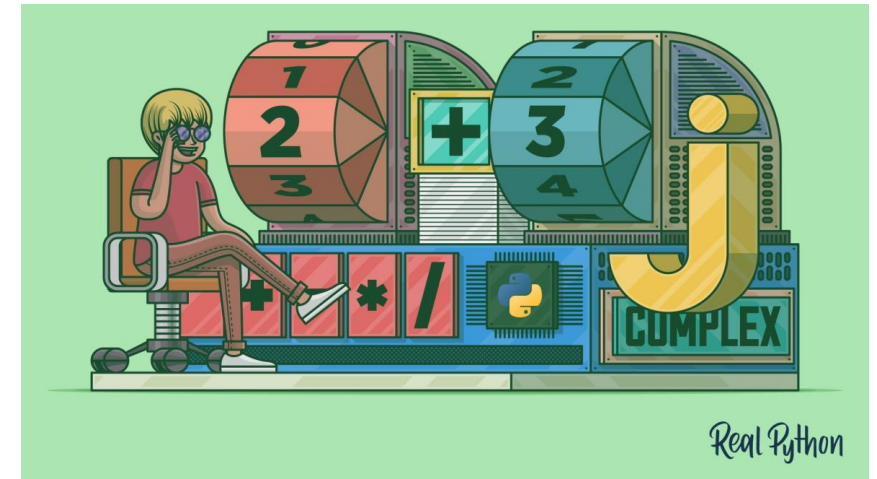
# Khai báo lớp CSoPhuc

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

# Thiết kế lớp CSoPhuc

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

- + ...
- + Toán tử cộng hai số phức.
- + Toán tử trừ hai số phức.
- + Toán tử nhân hai số phức.
- + Toán tử chia hai số phức.
- + ...

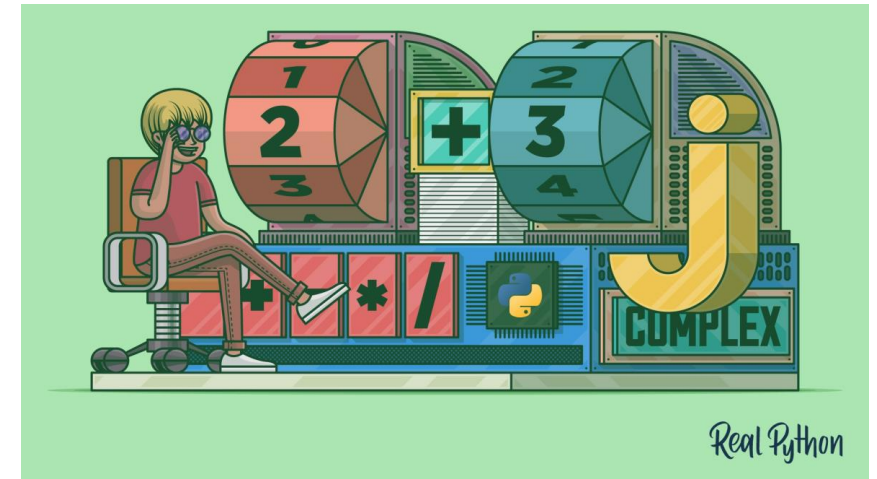


# Khai báo lớp CSoPhuc

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

# Thiết kế lớp CSoPhuc

- Nhóm phương thức xử lý (tiếp tục)
  - + ...
  - + Toán tử cộng bằng hai số phức.
  - + Toán tử trừ bằng hai số phức.
  - + Toán tử nhân bằng hai số phức.
  - + Toán tử chia bằng hai số phức.
  - + ...





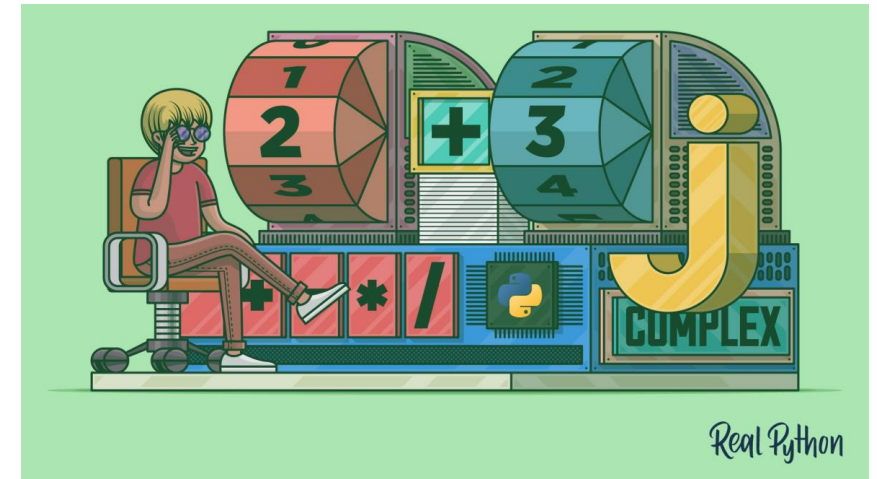
# Khai báo lớp CSoPhuc

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

# Thiết kế lớp CSoPhuc

## – 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 CSoPhuc

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

# Thiết kế lớp CSoPhuc

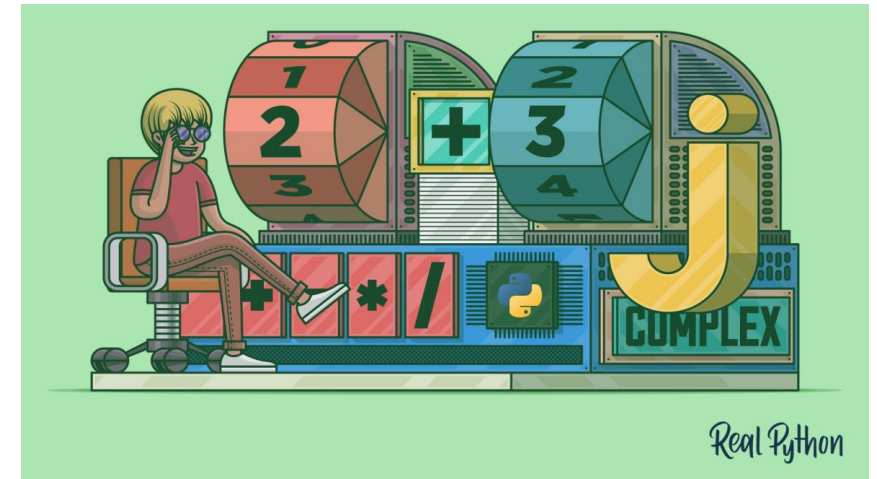
– Nhóm phương thức xử lý (tiếp tục)

+ ...

+ Toán tử lũy thừa số phức.

+ Toán tử lũy thừa bằng số phức.

+ ...



# Khai báo lớp CSoPhuc

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

# Thiết kế lớp CSoPhuc

– Nhóm phương thức xử lý (tiếp tục)

+ ...

+ Tính modun số phức.

+ Tìm số phức liên hợp.

+ ...





# Khai báo lớp CSoPhuc

```
11.      ///// Nhóm phương thức xử lý
12.      ///// Chương 09 - Thiết kế lớp
13.      ...
14.      float Module();
15.      CSoPhuc LienHop();
16.      ...
```

# Thiết kế lớp CSoPhuc

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

+ ...

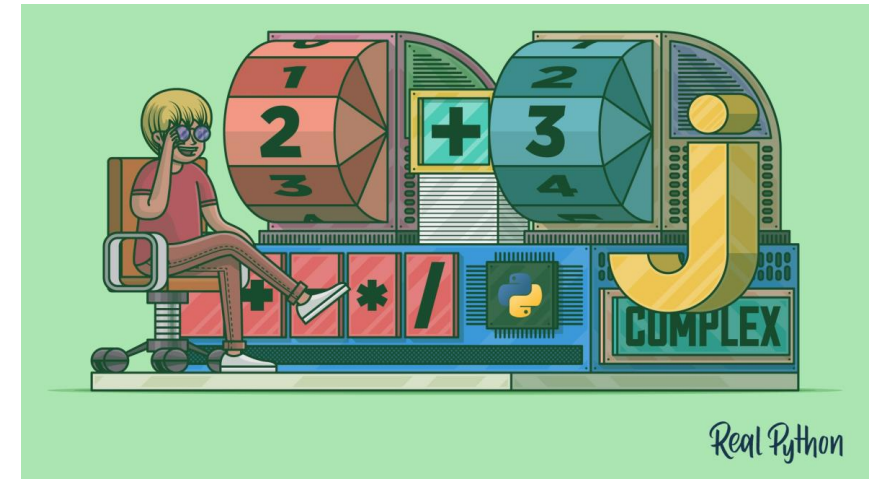
+ Tính tổng hai số phức.

+ Tính hiệu hai số phức.

+ Tính tích hai số phức.

+ Tính thương hai số phức.

+ Kết thúc nhóm phương thức xử lý.



# Khai báo lớp CSoPhuc

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

Định nghĩa các phương thức

# **NHÓM PHƯƠNG THỨC KHỞI TẠO**

# Định nghĩa phương thức

```
11. void CSoPhuc::Nhap()  
12. {  
13.     cout << "Nhap phan thuc:";  
14.     cin >> Thuc;  
15.     cout << "Nhap phan ao:";  
16.     cin >> Ao;  
17. }
```

# Định nghĩa phương thức

```
11.istream& operator >> (istream& is, CSoPhuc& x)
12.{
13.    cout << "Nhap phan thuc:";
14.    is >> x.Thuc;
15.    cout << "Nhap phan ao:";
16.    is >> x.Ao;
17.    return is;
18.}
```



# Định nghĩa phương thức

```
11. CSoPhuc::CSoPhuc()
```

```
12. {
```

```
13. |   Thuc = 0;
```

```
14. |   Ao  = 0;
```

```
15. }
```

# Định nghĩa phương thức

```
11. CSoPhuc::CSoPhuc(const CSoPhuc& x)
12. {
13. |   Thuc = x.Thuc;
14. |   Ao = x.Ao;
15. }
```

# Định nghĩa phương thức

```
11. CSoPhuc::CSoPhuc(float ThucThuc, float AoAo)
12. {
13. |   Thuc = ThucThuc;
14. |   Ao = AoAo;
15. }
```

# Định nghĩa phương thức

```
11. CSoPhuc::CSoPhuc(float ThucThuc)
12. {
13. |   Thuc = ThucThuc;
14. |   Ao = 0;
15. }
```

# Định nghĩa phương thức

```
11. void CSoPhuc::KhoiTao()  
12. {  
13. |   Thuc = 0;  
14. |   Ao = 0;  
15. }
```

# Định nghĩa phương thức

```
11. void CSoPhuc::KhoiTao(const CSoPhuc& x)
12. {
13.     Thuc = x.Thuc;
14.     Ao = x.Ao;
15. }
```



# Định nghĩa phương thức

```
11. void CSoPhuc::KhoiTao(float ThucThuc, float AoAo)
12. {
13.     Thuc = ThucThuc;
14.     Ao = AoAo;
15. }
```

# Định nghĩa phương thức

```
11. void CSoPhuc::KhoiTao(float ThucThuc)
12. {
13.     Thuc = ThucThuc;
14.     Ao = 0;
15. }
```

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

# Định nghĩa phương thức

```
11. void CSoPhuc::Xuat()  
12. {  
13. |   cout << Thuc << "+" << Ao << "i";  
14. }
```

# Định nghĩa phương thức

```
11 ostream& operator << (ostream& os, CSoPhuc& x)
12 {
13     |   os << x.Thuc << "+" << x.Ao << "i";
14     |   return os;
15 }
```

# Định nghĩa phương thức

```
11.float CSoPhuc::GetThuc()  
12.{  
13.|    return Thuc;  
14.}
```



# Định nghĩa phương thức

```
11.float CSoPhuc::GetAo()  
12.{  
13.|    return Ao;  
14.}
```

Định nghĩa các phương thức

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

# Định nghĩa phương thức

```
11.CSoPhuc& CSoPhuc::operator=(const CSoPhuc& x)
12.{
13.    Thuc = x.Thuc;
14.    Ao = x.Ao;
15.    return *this;
16.}
```

# Định nghĩa phương thức

```
11. void CSoPhuc::SetThuc(float ThucThuc)
12. {
13. |   Thuc = ThucThuc;
14. }
```

# Định nghĩa phương thức

```
11. void CSoPhuc::SetAo(float AoAo)
12. {
13. |   Ao = AoAo;
14. }
```

Định nghĩa các phương thức

# **NHÓM PHƯƠNG THỨC KIỂM TRA**



# Định nghĩa phương thức

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

# Định nghĩa phương thức

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

# Định nghĩa phương thức

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

# Định nghĩa phương thức

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

# Định nghĩa phương thức

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

# Định nghĩa phương thức

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



# Định nghĩa phương thức

```
11. bool CSoPhuc::ktThuanAo()  
12. {  
13.     if (Thuc == 0)  
14.         return true;  
15.     return false;  
16. }
```

# Định nghĩa phương thức

```
11. bool CSoPhuc::ktThuanThuc()  
12. {  
13.     if (Ao == 0)  
14.         return true;  
15.     return false;  
16. }
```

Định nghĩa các phương thức

# **NHÓM PHƯƠNG THỨC XỬ LÝ**

# Định nghĩa phương thức

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

# Định nghĩa phương thức

```
11. CSoPhuc CSoPhuc::operator+(const CSoPhuc& x)
12. {
13.     CSoPhuc temp;
14.     temp.Thuc = Thuc + x.Thuc;
15.     temp.Ao = Ao + x.Ao;
16.     return temp;
17. }
```

# Định nghĩa phương thức

```
11. CSoPhuc CSoPhuc::operator-(const CSoPhuc& x)
12. {
13.     CSoPhuc temp;
14.     temp.Thuc = Thuc - x.Thuc;
15.     temp.Ao = Ao - x.Ao;
16.     return temp;
17. }
```



# Định nghĩa phương thức

```
11. CSoPhuc CSoPhuc::operator*(const CSoPhuc& x)
12. {
13.     CSoPhuc temp;
14.     temp.Thuc = Thuc * x.Thuc - Ao * x.Ao;
15.     temp.Ao = Thuc * x.Ao + Ao * x.Thuc;
16.     return temp;
17. }
```

# Định nghĩa phương thức

```
11. CSoPhuc CSoPhuc::operator/(const CSoPhuc& x)
12. {
13.     CSoPhuc temp;
14.     temp.Thuc = (Thuc * x.Thuc + Ao * x.Ao) /
15.                 (x.Thuc * x.Thuc + x.Ao * x.Ao);
16.     temp.Ao = (Ao * x.Thuc - Thuc * x.Ao) /
17.               (x.Thuc * x.Thuc + x.Ao * x.Ao);
18.     return temp;
19. }
```

# Định nghĩa phương thức

```
11.CSoPhuc& CSoPhuc::operator+=(const CSoPhuc& x)
12.{
13.    Thuc = Thuc + x.Thuc;
14.    Ao = Ao + x.Ao;
15.    return *this;
16.}
```

# Định nghĩa phương thức

```
11.CSoPhuc& CSoPhuc::operator-=(const CSoPhuc& x)
12.{
13.    Thuc = Thuc - x.Thuc;
14.    Ao = Ao - x.Ao;
15.    return *this;
16.}
```

# Định nghĩa phương thức

```
11.CSoPhuc& CSoPhuc::operator*=(const CSoPhuc& x)
12.{
13.    float ThucThuc = Thuc * x.Thuc - Ao * x.Ao;
14.    float AoAo = Thuc * x.Ao + Ao * x.Thuc;
15.    Thuc = ThucThuc;
16.    Ao = AoAo;
17.    return *this;
18.}
```

# Định nghĩa phương thức

```
11.CSoPhuc& CSoPhuc::operator/=(const CSoPhuc& x)
12.{
13.    float ThucThuc = (Thuc * x.Thuc + Ao * x.Ao) /
14.        (x.Thuc * x.Thuc + x.Ao * x.Ao);
15.    float AoAo = (Ao * x.Thuc - Thuc * x.Ao) /
16.        (x.Thuc * x.Thuc + x.Ao * x.Ao);
17.    Thuc = ThucThuc;
18.    Ao = AoAo;
19.    return *this;
20.}
```



# Định nghĩa phương thức

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

# Định nghĩa phương thức

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

# Định nghĩa phương thức

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

# Định nghĩa phương thức

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

# Định nghĩa phương thức

```
11.CSoPhuc CSoPhuc::operator^(int n)
12.{
13.    CSoPhuc temp(1, 0);
14.    for (int i = 1; i <= n; i++)
15.        temp = temp * (*this);
16.    return temp;
17.}
```

# Định nghĩa phương thức

```
11.CSoPhuc& CSoPhuc::operator^=(int n)
12.{
13.    CSoPhuc temp(1, 0);
14.    for (int i = 1; i <= n; i++)
15.        temp = temp * (*this);
16.    *this = temp;
17.    return *this;
18.}
```



# Định nghĩa phương thức

```
11.float CSoPhuc::Module()  
12.{  
13.|    return (float)sqrt(Thuc * Thuc + Ao * Ao);  
14.}
```

# Định nghĩa phương thức

```
11.CSoPhuc CSoPhuc::LienHop()  
12.{  
13.    CSoPhuc temp;  
14.    temp.Thuc = Thuc;  
15.    temp.Ao = -Ao;  
16.    return temp;  
17.}
```

# Định nghĩa phương thức

```
11. CSoPhuc CSoPhuc::Tong(const CSoPhuc& x)
12. {
13.     CSoPhuc temp;
14.     temp.Thuc = Thuc + x.Thuc;
15.     temp.Ao = Ao + x.Ao;
16.     return temp;
17. }
```

# Định nghĩa phương thức

```
11. CSoPhuc CSoPhuc::Hieu(const CSoPhuc& x)
12. {
13.     CSoPhuc temp;
14.     temp.Thuc = Thuc - x.Thuc;
15.     temp.Ao = Ao - x.Ao;
16.     return temp;
17. }
```

# Định nghĩa phương thức

```
11. CSoPhuc CSoPhuc::Tich(const CSoPhuc& x)
12. {
13.     CSoPhuc temp;
14.     temp.Thuc = Thuc * x.Thuc - Ao * x.Ao;
15.     temp.Ao = Thuc * x.Ao + Ao * x.Thuc;
16.     return temp;
17. }
```

# Định nghĩa phương thức

```
11. CSoPhuc CSoPhuc::Thuong(const CSoPhuc& x)
12. {
13.     CSoPhuc temp;
14.     temp.Thuc = ((Thuc * x.Thuc) + (Ao * x.Ao)) /
15.                 (x.Thuc * x.Thuc + x.Ao * x.Ao);
16.     temp.Ao = ((Ao * x.Thuc) - (Thuc * x.Ao)) /
17.                (x.Thuc * x.Thuc + x.Ao * x.Ao);
18.     return temp;
19. }
```

**Cảm ơn quý vị đã lắng nghe**

**Nhóm tác giả**

**TS. Nguyễn Tấn Trần Minh Khang**