

# Chương 7

## TOÁN TỬ SỐ HỌC

- 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

# 1. MỤC TIÊU



# 1. Mục tiêu

- Hiểu được các toán tử số học (arithmetic operators) là gì?
- Hiểu được vai trò của toán tử số học trong C++



## 2. BÀI TOÁN DẪN NHẬP



## 2. Bài toán dẫn nhập

- Bài toán: Viết chương trình nhập vào hai phân số. Tính tổng giữa chúng và xuất kết quả bằng phương pháp lập trình hướng đối tượng.
- Chương trình

```
11.#include <iostream>  
12.using namespace std;
```

## 2. Bài toán dẫn nhập

```
11.class CPhanSo
12.{
13.    private:
14.        int Tu;
15.        int Mau;
16.    public:
17.        void Nhap();
18.        void Xuat();
19.        CPhanSo Tong(CPhanSo);
20.};
```

## 2. Bài toán dẫn nhập

```
23. void main()  
24. {  
25.     CPhanSo a, b, kq;  
26.     a.Nhap();  
27.     b.Nhap();  
28.     kq = a.Tong(b);  
29.     cout << "Tong la:";  
30.     kq.Xuat();  
31. }
```



## 2. Bài toán dẫn nhập

```
32. void CPhanSo::Nhap()  
33. {  
34.     cout << "Nhap tu:";  
35.     cin >> Tu;  
36.     cout << "Nhap mau:";  
37.     cin >> Mau;  
38. }
```



## 2. Bài toán dẫn nhập

```
39. void CPhanSo::Xuat()  
40. {  
41.     cout << "\nTu: " << Tu;  
42.     cout << "\nMau: " << Mau;  
43. }
```

## 2. Bài toán dẫn nhập

```
44. CPhanSo CPhanSo::Tong(CPhanSo x)
45. {
46.     CPhanSo temp;
47.     temp.Tu = Tu * x.Mau + Mau * x.Tu;
48.     temp.Mau = Mau * x.Mau;
49.     return temp;
50. }
```

## 3. KHÁI NIỆM



## 3. Khái niệm

- Trong ngôn ngữ lập trình C/C++ có các toán tử số học (**arithmetic operators**) như sau:
  - + Toán tử cộng – **addition operator** (operator +)
  - + Toán tử trừ – **subtraction operator** (operator -)
  - + Toán tử nhân – **multiplication operator** (operator \*)
  - + Toán tử chia – **division operator** (operator /)
  - + Toán tử mod – **modulo operator** (operator %).

## 3. Khái niệm

- Trong ngôn ngữ lập trình C còn có các toán tử số học gán kết hợp (**compound assignment operators**) như sau:
  - + Toán tử cộng bằng (operator +=)
  - + Toán tử trừ bằng (operator -=)
  - + Toán tử nhân bằng (operator \*=)
  - + Toán tử chia bằng (operator /=)
  - + Toán tử mod bằng (operator %=)

## 3. Khái niệm

- Trong ngôn ngữ lập trình C còn có các toán tử thao tác bit kết hợp toán tử gán (**compound assignment bitwise operators**) như sau:
  - + Toán tử dịch phải bit – **shift right** (operator  $\gg=$ ).
  - + Toán tử dịch trái bit – **shift left** (operator  $\ll=$ ).
  - + Toán tử and trên bit – **bitwise AND** (operator  $\&=$ ).
  - + Toán tử xor trên bit – **bitwise XOR** (operator  $\wedge=$ ).
  - + Toán tử or trên bit – **bitwise OR** (operator  $\|=$ ).

## 3. Khái niệm

- Trong ngôn ngữ lập trình C có các toán tử tăng giảm một (**increment and decrement operators**) như sau:
  - + Toán tử tăng một – **increment operator** (operator ++)
  - + Toán tử giảm một – **decrement operator** (operator --)



## 4. ĐẶT VẤN ĐỀ



### 3. Đặt vấn đề

— Hãy khai báo và định nghĩa các phương thức và toán tử cần thiết để các câu lệnh sau có thể thực hiện.

```
11.CPhanSo a,b,kq;
```

```
12.cin >> a >> b;
```

```
13.kq = a + b;
```

```
14.cout << "Tong:" << kq;
```

```
15.kq = a - b;
```

```
16.cout << "Hieu:" << kq;
```



### 3. Đặt vấn đề

— Hãy khai báo và định nghĩa các phương thức và toán tử cần thiết để các câu lệnh sau có thể thực hiện.

17. `kq = a * b;`

18. `cout << "Tich:" << kq;`

19. `kq = a / b;`

20. `cout << "Thuong:" << kq;`



## 5. GIẢI QUYẾT VẤN ĐỀ



## 5. Giải quyết vấn đề

— Để giải quyết vấn đề trên ta phải khai báo và định nghĩa các toán tử số học cho lớp đối tượng CPhanSo.

— Khai báo lớp

```
11.class CPhanSo
12.{
13.    private:
14.        int Tu;
15.        int Mau;
16.    public:
17.        CPhanSo Tong(CPhanSo);
18.        CPhanSo operator+(CPhanSo);
19.        ...
20.};
```





## 5. Giải quyết vấn đề

— Để giải quyết vấn đề trên ta phải khai báo và định nghĩa các toán tử số học cho lớp đối tượng CPhanSo.

— Khai báo lớp

```
11.class CPhanSo
12.{
13.    private:
14.        int Tu;
15.        int Mau;
16.    public:
17.        CPhanSo Tong(CPhanSo);
18.        CPhanSo operator+(CPhanSo);
19.        CPhanSo operator-(CPhanSo);
20.        CPhanSo operator*(CPhanSo);
21.        CPhanSo operator/(CPhanSo);
22.};
```



## 5. Giải quyết vấn đề

```

23. CPhanSo CPhanSo::Tong(CPhanSo x)
24. {
25.     CPhanSo temp;
26.     temp.Tu = Tu * x.Mau + Mau * x.Tu;
27.     temp.Mau = Mau * x.Mau;
28.     return temp;
29. }
    
```





## 5. Giải quyết vấn đề

```

23. CPhanSo CPhanSo::operator+(CPhanSo x)
24. {
25.     CPhanSo temp;
26.     temp.Tu = Tu * x.Mau + Mau * x.Tu;
27.     temp.Mau = Mau * x.Mau;
28.     return temp;
29. }

```



## 5. Giải quyết vấn đề

```

30. CPhanSo CPhanSo::operator-(CPhanSo x)
31. {
32.     CPhanSo temp;
33.     temp.Tu = Tu * x.Mau - Mau * x.Tu;
34.     temp.Mau = Mau * x.Mau;
35.     return temp;
36. }
  
```



## 5. Giải quyết vấn đề

```
37. CPhanSo CPhanSo::operator*(CPhanSo x)
38. {
39.     CPhanSo temp;
40.     temp.Tu = Tu * x.Tu;
41.     temp.Mau = Mau * x.Mau;
42.     return temp;
43. }
```



## 5. Giải quyết vấn đề

```

44. CPhanSo CPhanSo::operator/(CPhanSo x)
45. {
46.     CPhanSo temp;
47.     temp.Tu = Tu * x.Mau;
48.     temp.Mau = Mau * x.Tu;
49.     return temp;
50. }
  
```



## 6. BÀI TẬP VỀ NHÀ



## 6. Bài tập về nhà

- Bài 01: Hãy định nghĩa các toán tử  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $+=$ ,  $-=$ ,  $*=$ ,  $/=$ ,  $++$ ,  $--$  cho lớp đối tượng **CPhanSo**.
- Bài 02: Hãy định nghĩa các toán tử  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $+=$ ,  $-=$ ,  $*=$ ,  $/=$  cho lớp đối tượng **CSoPhuc**.
- Bài 03: Hãy định nghĩa các toán tử  $*$ ,  $/$ ,  $*=$ ,  $/=$  cho lớp đối tượng **CDonThuc**.
- Bài 04: Hãy định nghĩa các toán tử  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $+=$ ,  $-=$ ,  $*=$ ,  $/=$  cho lớp đối tượng **CDaThuc**.



**Cảm ơn quý vị đã lắng nghe**

**Nhóm tác giả**

**TS. Nguyễn Tấn Trần Minh Khang**



# Chương 7

## TOÁN TỬ SỐ HỌC PYTHON

- 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

# 1. BÀI TOÁN DẪN NHẬP

# Bài toán dẫn nhập

— Bài toán: Viết chương trình nhập vào hai phân số. Tính tổng giữa chúng và xuất kết quả bằng phương pháp lập trình hướng đối tượng.

— Chương trình

```
1. class CPhanSo:
2.     def __init__(self):
3.         self.Tu = 0
4.         self.Mau = 1
5.     ...
```

# Bài toán dẫn nhập

```
1. class CPhanSo:
2.     ...
5.     def Nhap(self):
6.         self.Tu = int(input("Nhap tu: "))
7.         self.Mau = int(input("Nhap mau: "))
8.     ...
```

# Bài toán dẫn nhập

```
1. class CPhanSo:
2.     ...
8.     def Xuat(self):
9.         print(self.Tu, "/", self.Mau)
10.    ...
```

# Bài toán dẫn nhập

```
1. class CPhanSo:
2.     ...
3.     def Tong(self, x):
4.         temp = CPhanSo()
5.         temp.Tu = self.Tu * x.Mau + self.Mau * x.Tu
6.         temp.Mau = self.Mau * x.Mau
7.         return temp
```

# Bài toán dẫn nhập

```
1. def main():  
2.     a = CPhanSo()  
3.     b = CPhanSo()  
4.     kq = CPhanSo()  
5.     a.Nhap()  
6.     b.Nhap()  
7.     kq = a.Tong(b)  
8.     print("Tong la:", end = ' ' )  
9.     kq.Xuat()
```



# Bài toán dẫn nhập

```
1. if __name__ == "__main__":  
2.     main()
```

## 2. KHÁI NIỆM

# Khái niệm

- Trong ngôn ngữ lập trình Python có các toán tử số học (**arithmetic operators**) như sau:
  - + Toán tử cộng – **addition operator** (`__add__`)
  - + Toán tử trừ – **subtraction operator** (`__sub__`)
  - + Toán tử nhân – **multiplication operator** (`__mul__`)
  - + Toán tử lũy thừa – **multiplication operator** (`__pow__`)
  - + Toán tử chia – **division operator** (`__truediv__`)
  - + Toán tử chia nguyên – **division operator** (`__floordiv__`)
  - + Toán tử mod – **modulo operator** (`__mod__`)

# Khái niệm

- Trong ngôn ngữ lập trình Python còn có các toán tử thao tác trên bit (**compound assignment bitwise operators**) như sau:
  - + Toán tử dịch phải bit – shift right (**\_\_rshift\_\_**)
  - + Toán tử dịch trái bit – shift left (**\_\_lshift\_\_**)
  - + Toán tử and trên bit – bitwise AND (**\_\_and\_\_**)
  - + Toán tử or trên bit – bitwise OR (**\_\_or\_\_**)
  - + Toán tử xor trên bit – bitwise XOR (**\_\_xor\_\_**)
  - + Toán tử not trên bit – bitwise NOT (**\_\_invert\_\_**)

### 3. ĐẶT VẤN ĐỀ

# Đặt vấn đề

— Hãy khai báo và định nghĩa các phương thức và toán tử cần thiết để các câu lệnh sau có thể thực hiện.

1. `a = CPhanSo()`
2. `b = CPhanSo()`
3. `kq = CPhanSo()`
4. `a.Nhap()`
5. `b.Nhap()`
6. ...

# Đặt vấn đề

— Hãy khai báo và định nghĩa các phương thức và toán tử cần thiết để các câu lệnh sau có thể thực hiện.

5. ...

6.  $kq = a + b$

7. `print("Tong:", kq)`

8.  $kq = a - b$

9. `print("Hieu:", kq)`

10....



# Đặt vấn đề

- Hãy khai báo và định nghĩa các phương thức và toán tử cần thiết để các câu lệnh sau có thể thực hiện.

10.....

11.kq = a \* b

12.print("Tich:", kq)

13.kq = a / b

14.print("Thuong:", kq)

## 4. GIẢI QUYẾT VẤN ĐỀ

# Giải quyết vấn đề

- Để giải quyết vấn đề trên ta phải khai báo và định nghĩa các toán tử số học cho lớp đối tượng CPhanSo.
- Khai báo và định nghĩa lớp

```
1. class CPhanSo:  
2.     def __init__(self):  
3.         self.Tu = 0  
4.         self.Mau = 1  
5.         ...
```

# Giải quyết vấn đề

```
1. class CPhanSo:
2.     ...
5.     def Nhap(self):
6.         self.Tu = int(input("Nhap tu: "))
7.         self.Mau = int(input("Nhap mau: "))
8.         ...
```

# Giải quyết vấn đề

```
1. class CPhanSo:
2.     ...
8.     def Xuat(self):
9.         print(self.Tu, "/", self.Mau)
10.    ...
```

# Giải quyết vấn đề

```
1. class CPhanSo:
2.     ...
9.     def __str__(self):
10.         return f"{self.Tu} / {self.Mau}"
11.     ...
```

# Giải quyết vấn đề

```
1. class CPhanSo:
2.     ...
3.     def Tong(self, x):
4.         temp = CPhanSo()
5.         temp.Tu = self.Tu * x.Mau + self.Mau * x.Tu
6.         temp.Mau = self.Mau * x.Mau
7.         return temp
8.     ...
```



# Giải quyết vấn đề

```
1. class CPhanSo:
2.     ...
3.     def __add__(self, x):
4.         temp = CPhanSo()
5.         temp.Tu = self.Tu * x.Mau + self.Mau * x.Tu
6.         temp.Mau = self.Mau * x.Mau
7.         return temp
8.     ...
```

# Giải quyết vấn đề

```
1. class CPhanSo:
2.     ...
3.     def __sub__(self, x):
4.         temp = CPhanSo()
5.         temp.Tu = self.Tu * x.Mau - self.Mau * x.Tu
6.         temp.Mau = self.Mau * x.Mau
7.         return temp
8.     ...
```

# Giải quyết vấn đề

```
1. class CPhanSo:
2.     ...
3.     def __mul__(self, x):
4.         temp = CPhanSo()
5.         temp.Tu = self.Tu * x.Tu
6.         temp.Mau = self.Mau * x.Mau
7.         return temp
8.     ...
```

# Giải quyết vấn đề

```
1. class CPhanSo:
2.     ...
3.     def __truediv__(self, x):
4.         temp = CPhanSo()
5.         temp.Tu = self.Tu * x.Mau
6.         temp.Mau = self.Mau * x.Tu
7.         return temp
```

## 5. BÀI TẬP VỀ NHÀ

# Bài tập về nhà

- Bài 01: Hãy định nghĩa các toán tử  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $+=$ ,  $-=$ ,  $*=$ ,  $/=$ ,  $++$ ,  $--$  cho lớp đối tượng CPhanSo.
- Bài 02: Hãy định nghĩa các toán tử  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $+=$ ,  $-=$ ,  $*=$ ,  $/=$  cho lớp đối tượng CSoPhuc.
- Bài 03: Hãy định nghĩa các toán tử  $*$ ,  $/$ ,  $*=$ ,  $/=$  cho lớp đối tượng CDonThuc.
- Bài 04: Hãy định nghĩa các toán tử  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $+=$ ,  $-=$ ,  $*=$ ,  $/=$  cho lớp đối tượng CDaThuc.

**Cảm ơn quý vị đã lắng nghe**

**Nhóm tác giả**

**TS. Nguyễn Tấn Trần Minh Khang**