

BÀI TOÁN CHI TIẾT MÁY

- 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

BÀI TOÁN

Bài toán

- Bài toán: Một cái máy có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
 - + Chi tiết đơn: là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: mã số chi tiết, giá tiền.
 - + Chi tiết phức: là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.

Bài toán

Yêu cầu: Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:

- Nhập các chi tiết cho máy.
- **Xuất các chi tiết máy.**
- Tính trị giá của máy.
- **Tìm kiếm một chi tiết máy theo mã số.**
- **Đếm số lượng chi tiết đơn có trong cái máy (Bài tập về nhà).**

THIẾT KẾ LỚP SƠ BỘ

Bài toán

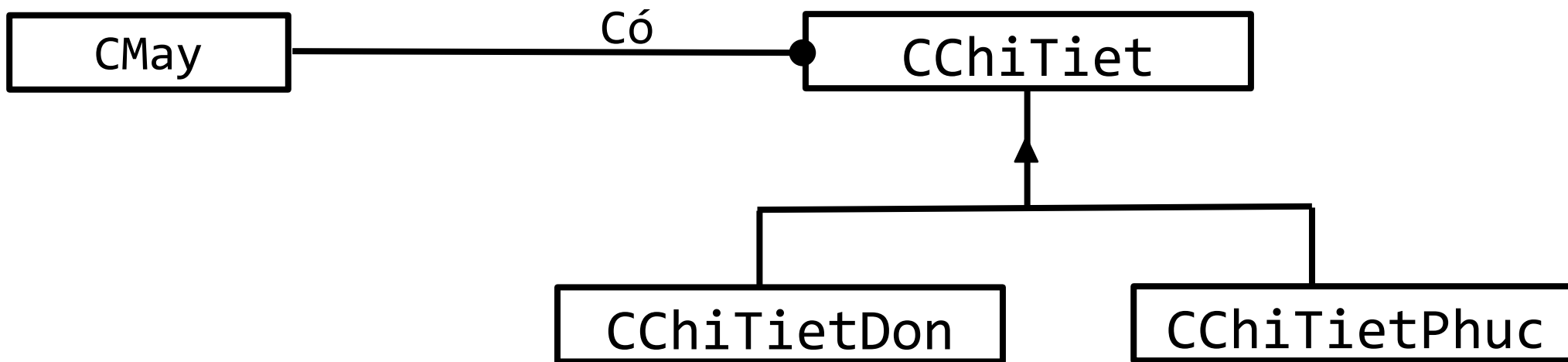
- Bài toán: Một cái máy có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
 - + Chi tiết đơn: là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: mã số chi tiết, giá tiền.
 - + Chi tiết phức: là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.

Bài toán

- Bài toán: Một cái máy có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
- + **Chi tiết đơn**: là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: mã số chi tiết, giá tiền.
- + **Chi tiết phức**: là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.

Bài toán

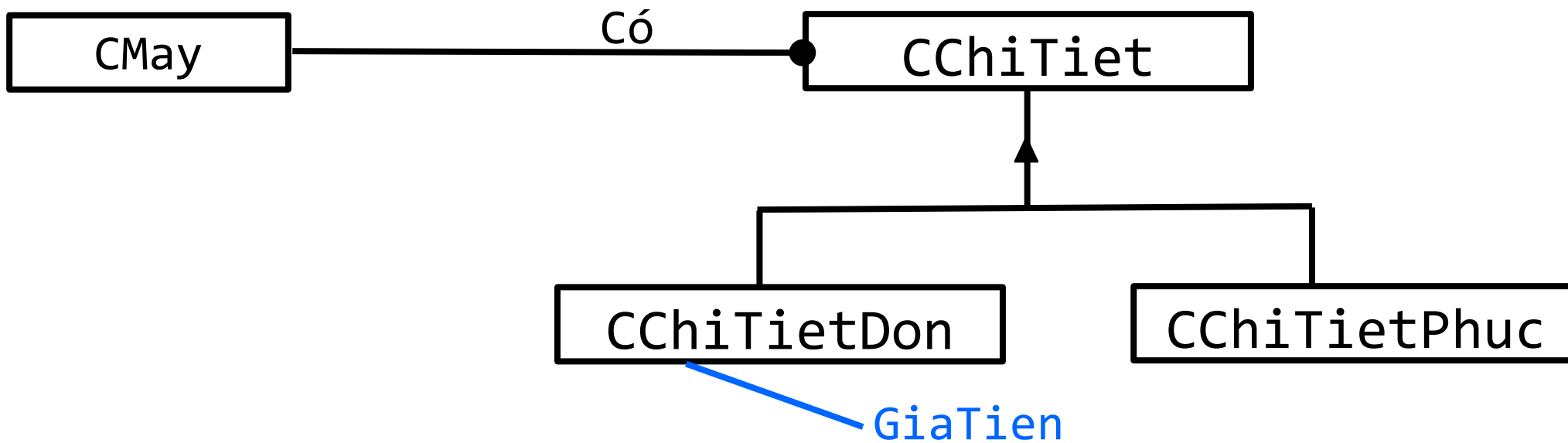
- Bài toán: **Một cái máy** có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
- + **Chi tiết đơn**: là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: mã số chi tiết, giá tiền.
- + **Chi tiết phức**: là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.



THIẾT KẾ LỚP CHI TIẾT

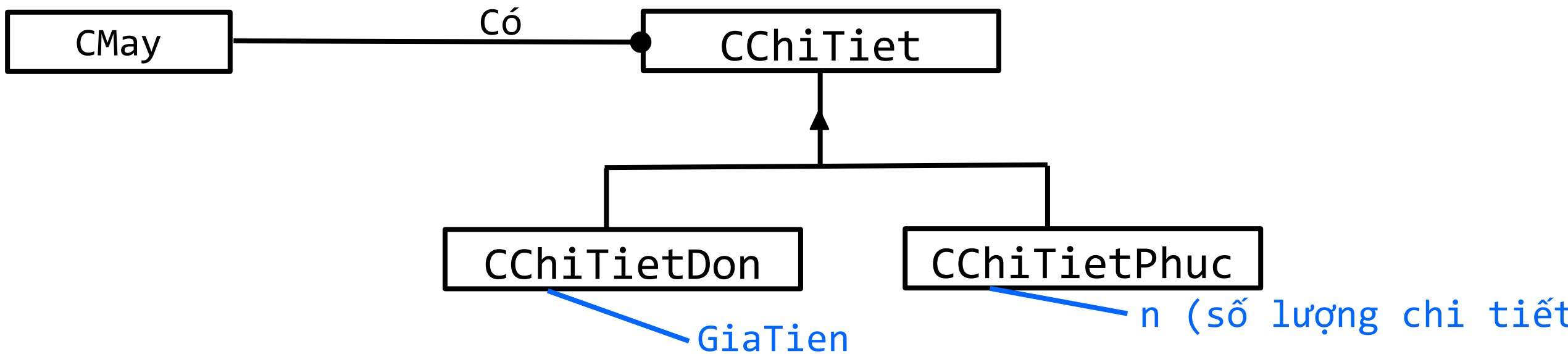
Bài toán

- Bài toán: Một cái máy có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
 - + Chi tiết đơn: là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: mã số chi tiết, giá tiền.
 - + Chi tiết phức: là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.



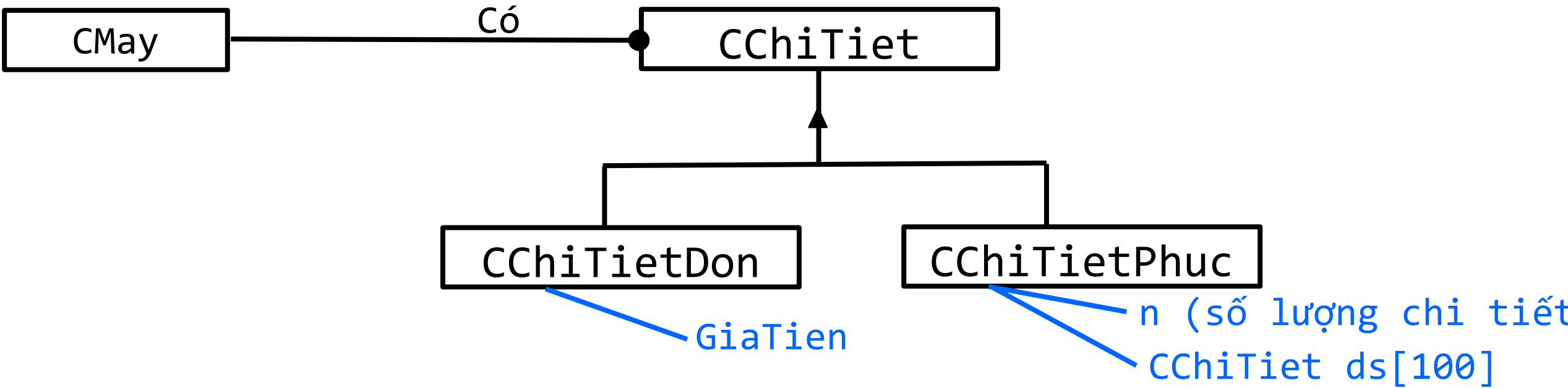
Bài toán

- Bài toán: Một cái máy có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
 - + Chi tiết đơn: là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: mã số chi tiết, giá tiền.
 - + Chi tiết phức: là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.



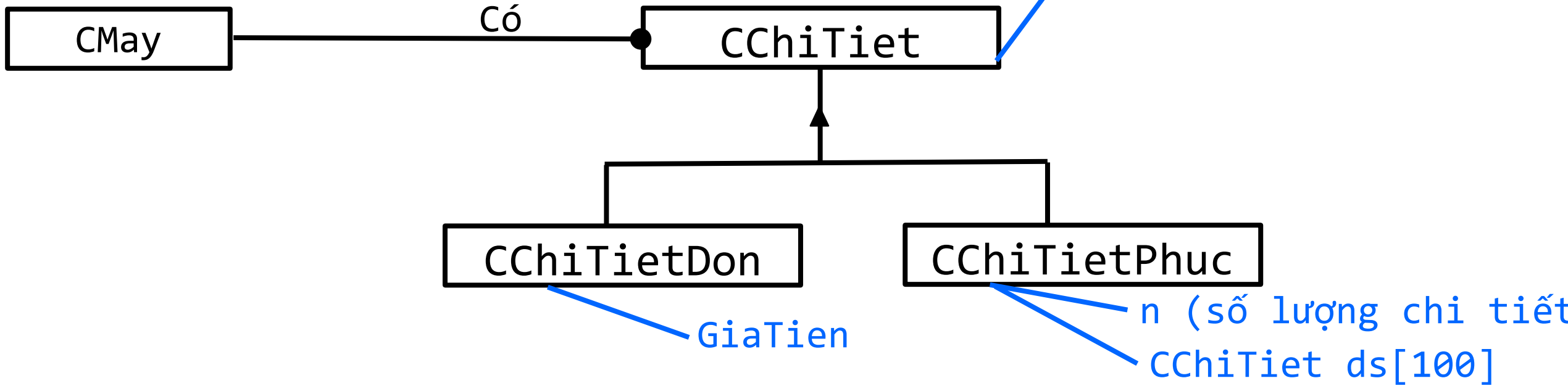
Bài toán

- Bài toán: Một cái máy có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
 - + Chi tiết đơn: là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: mã số chi tiết, giá tiền.
 - + Chi tiết phức: là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.



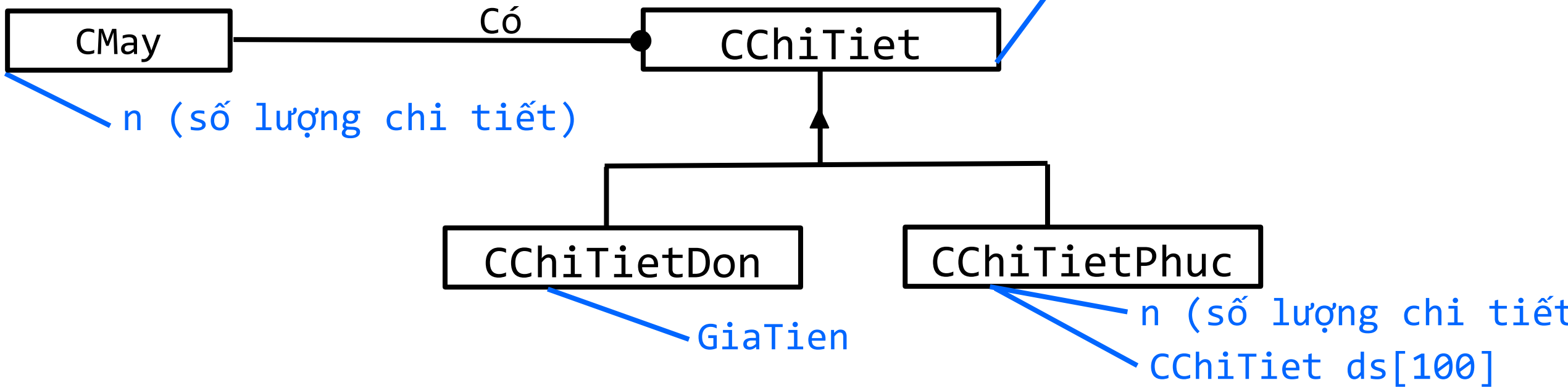
Bài toán

- Bài toán: Một cái máy có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
 - + Chi tiết đơn: là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: mã số chi tiết, giá tiền.
 - + Chi tiết phức: là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.



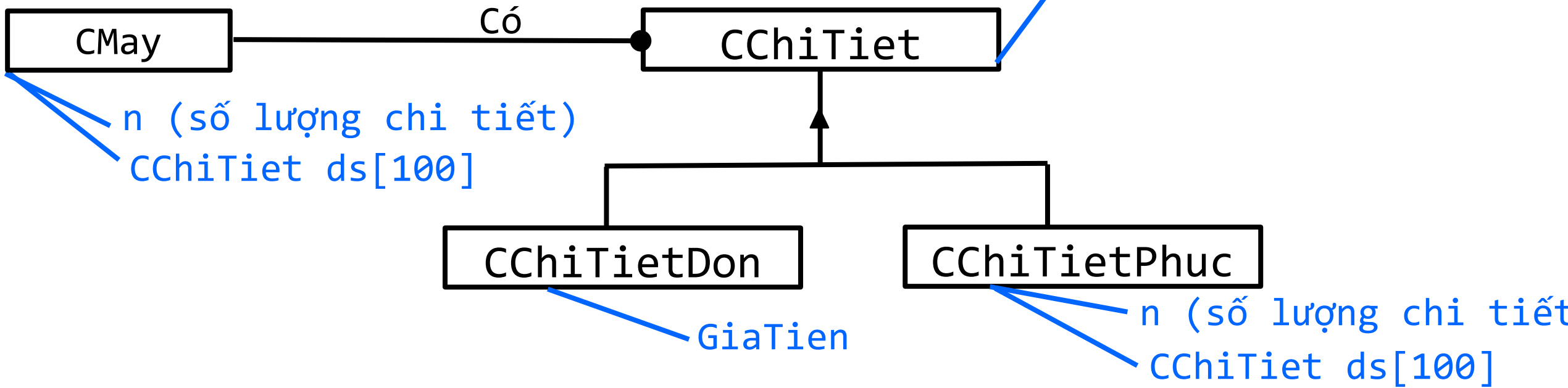
Bài toán

- Bài toán: Một cái máy có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
 - + Chi tiết đơn: là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: mã số chi tiết, giá tiền.
 - + Chi tiết phức: là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.



Bài toán

- Bài toán: Một cái máy có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
 - + Chi tiết đơn: là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: mã số chi tiết, giá tiền.
 - + Chi tiết phức: là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.



Bài toán

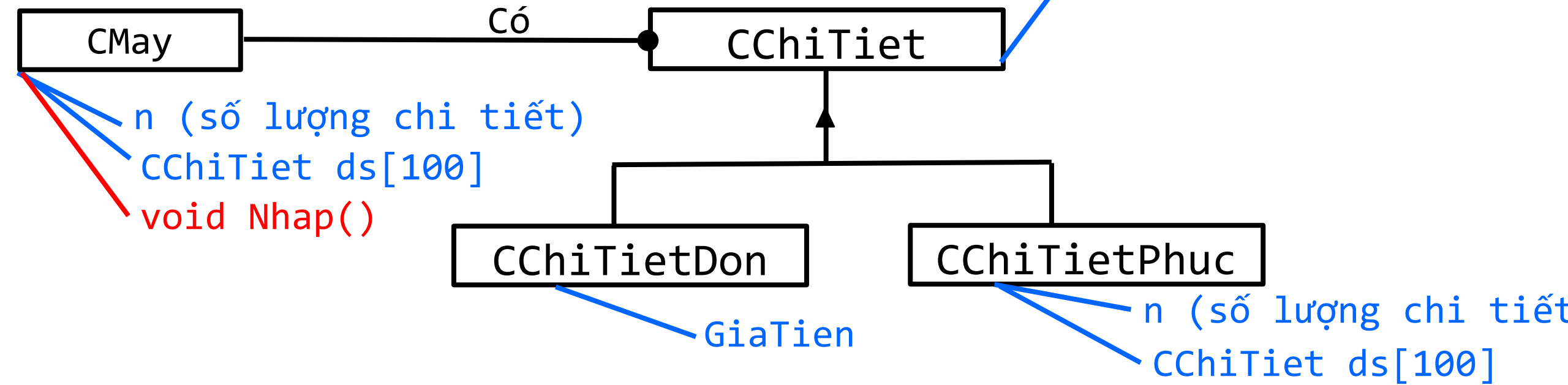
Yêu cầu: Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:

- Nhập các chi tiết cho máy.
- **Xuất các chi tiết máy.**
- Tính trị giá của máy.
- **Tìm kiếm một chi tiết máy theo mã số.**
- **Đếm số lượng chi tiết đơn có trong cái máy (Bài tập về nhà).**

Bài toán

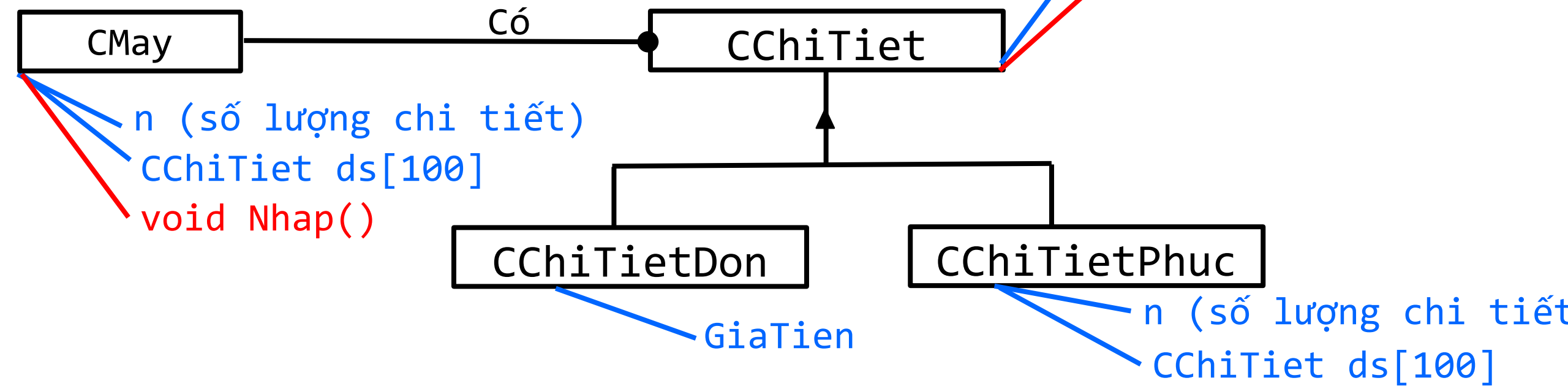
Yêu cầu: Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:

- Nhập các chi tiết cho máy.
- Xuất các chi tiết máy.
- Tính trị giá của máy.
- Tìm kiếm một chi tiết máy theo mã số.
- Đếm số lượng chi tiết đơn có trong cái máy (Bài tập về nhà).



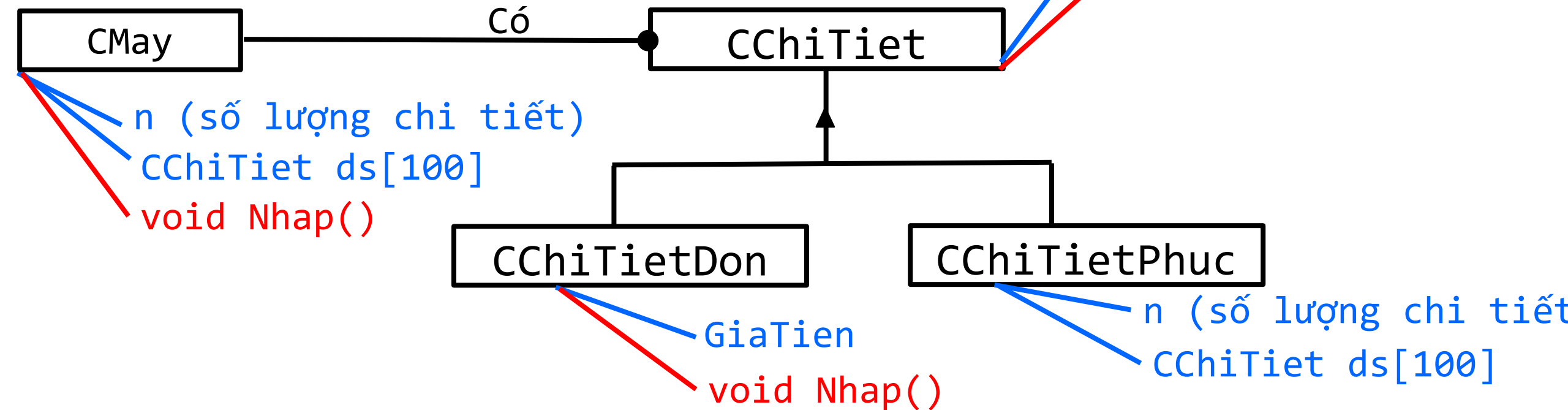
MaSo (long)

void Nhap()



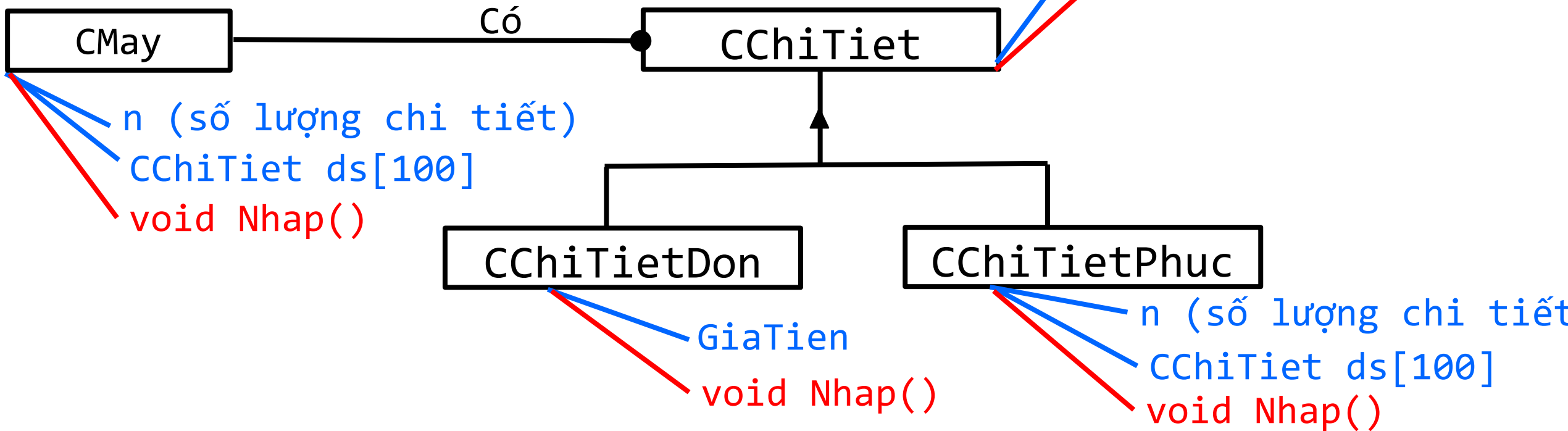
MaSo (long)

void Nhap()



MaSo (long)

void Nhap()



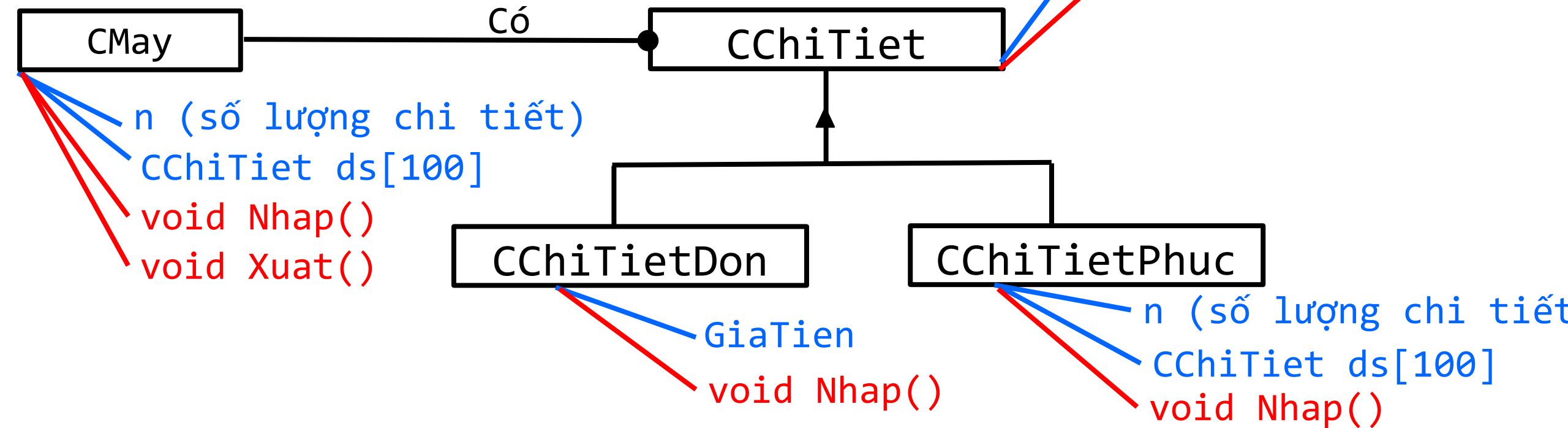
Bài toán

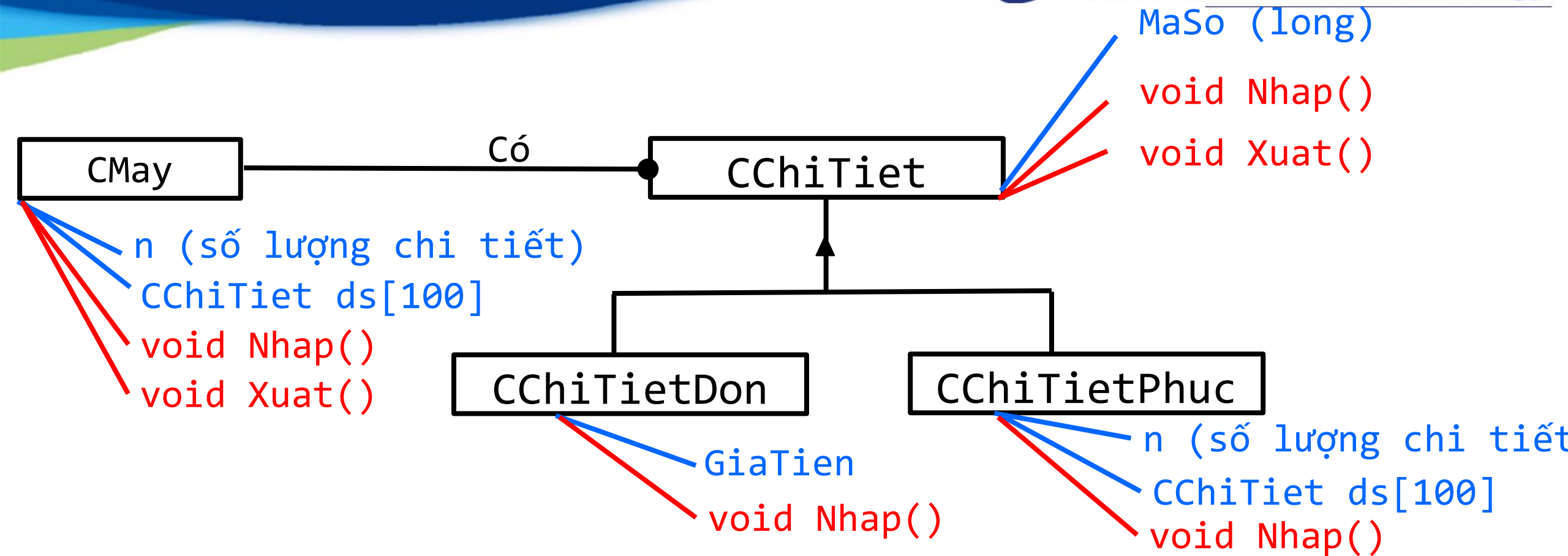
Yêu cầu: Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:

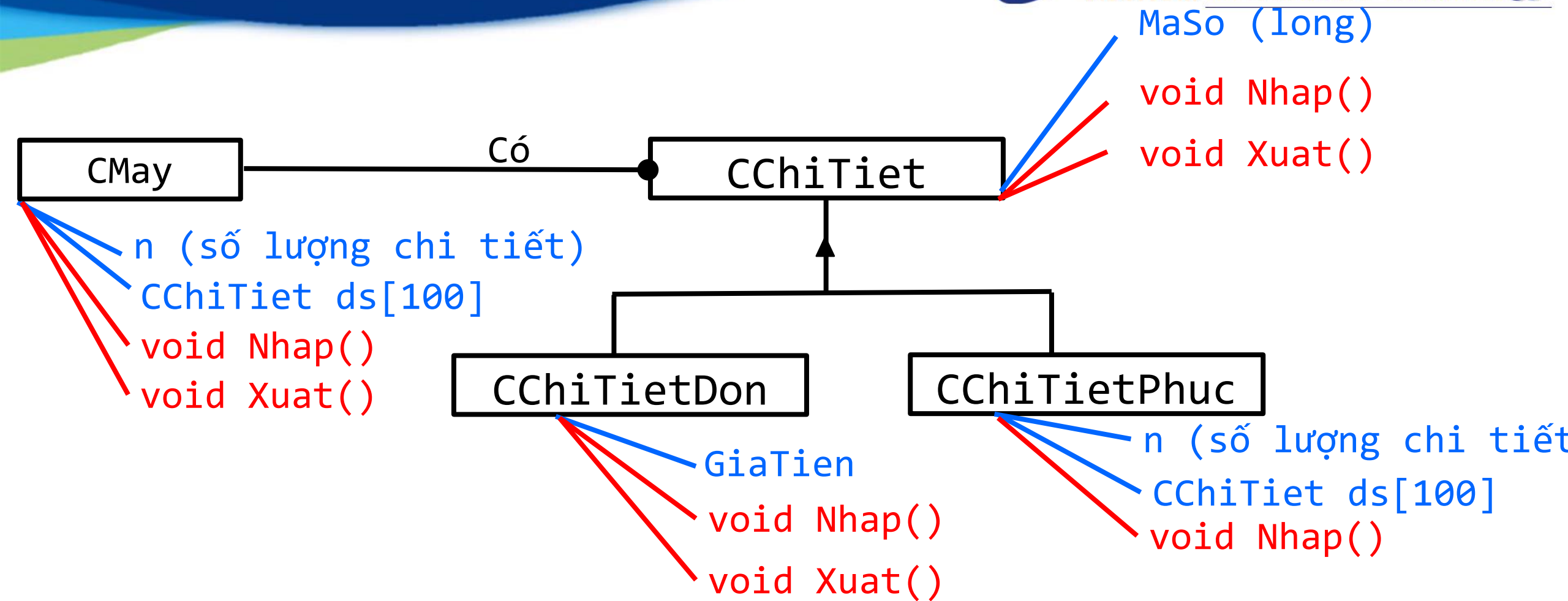
- Nhập các chi tiết cho máy.
- Xuất các chi tiết máy.
- Tính trị giá của máy.
- Tìm kiếm một chi tiết máy theo mã số.
- Đếm số lượng chi tiết đơn có trong cái máy (Bài tập về nhà).

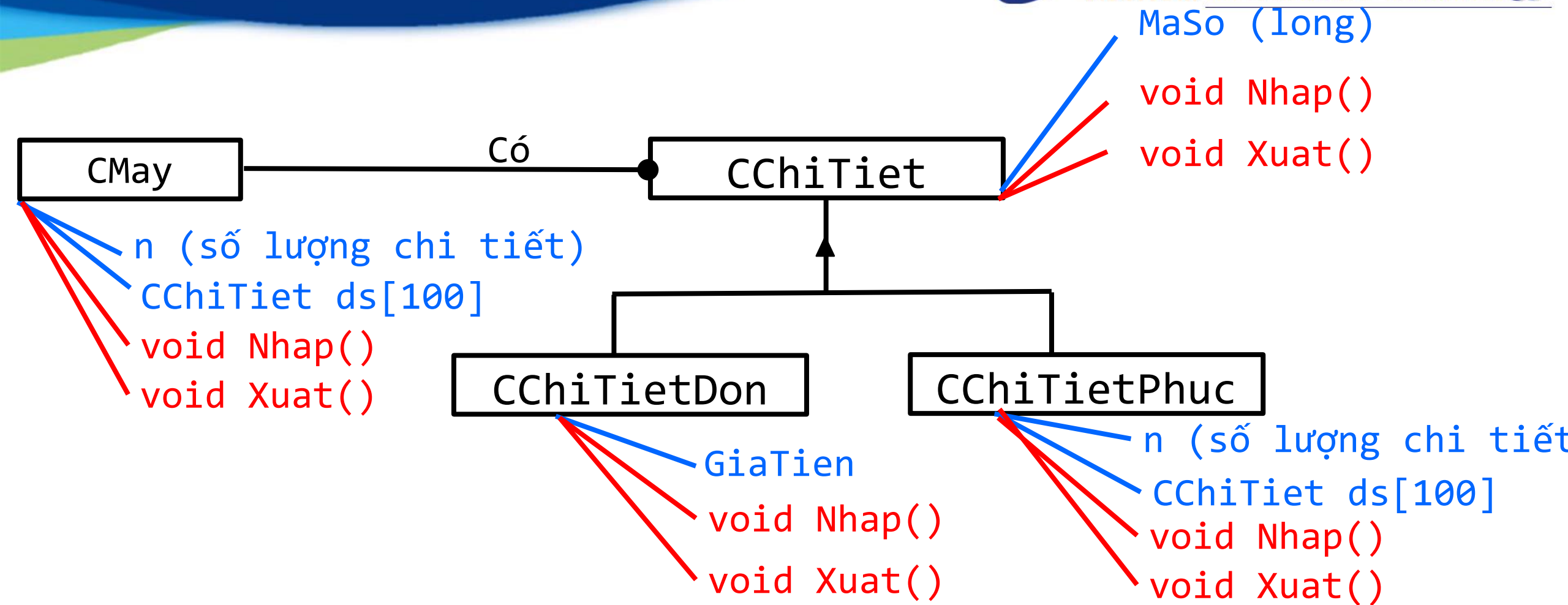
MaSo (long)

void Nhap()





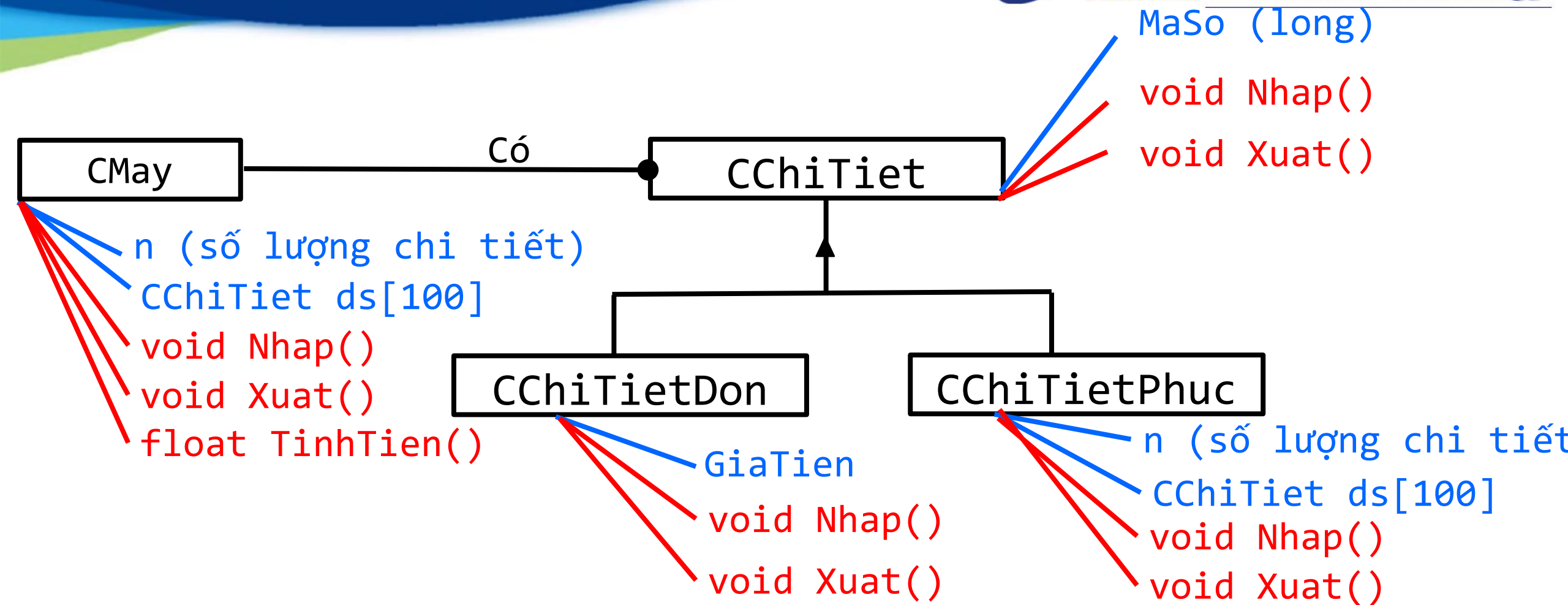


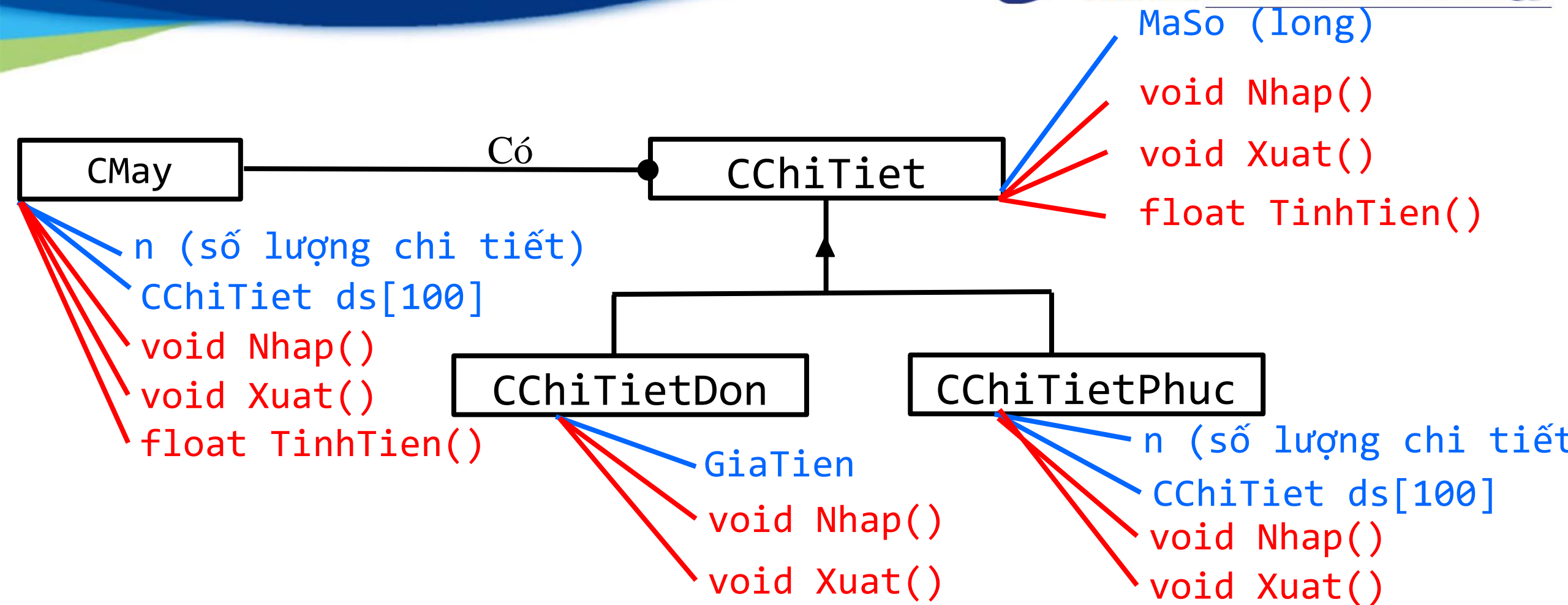


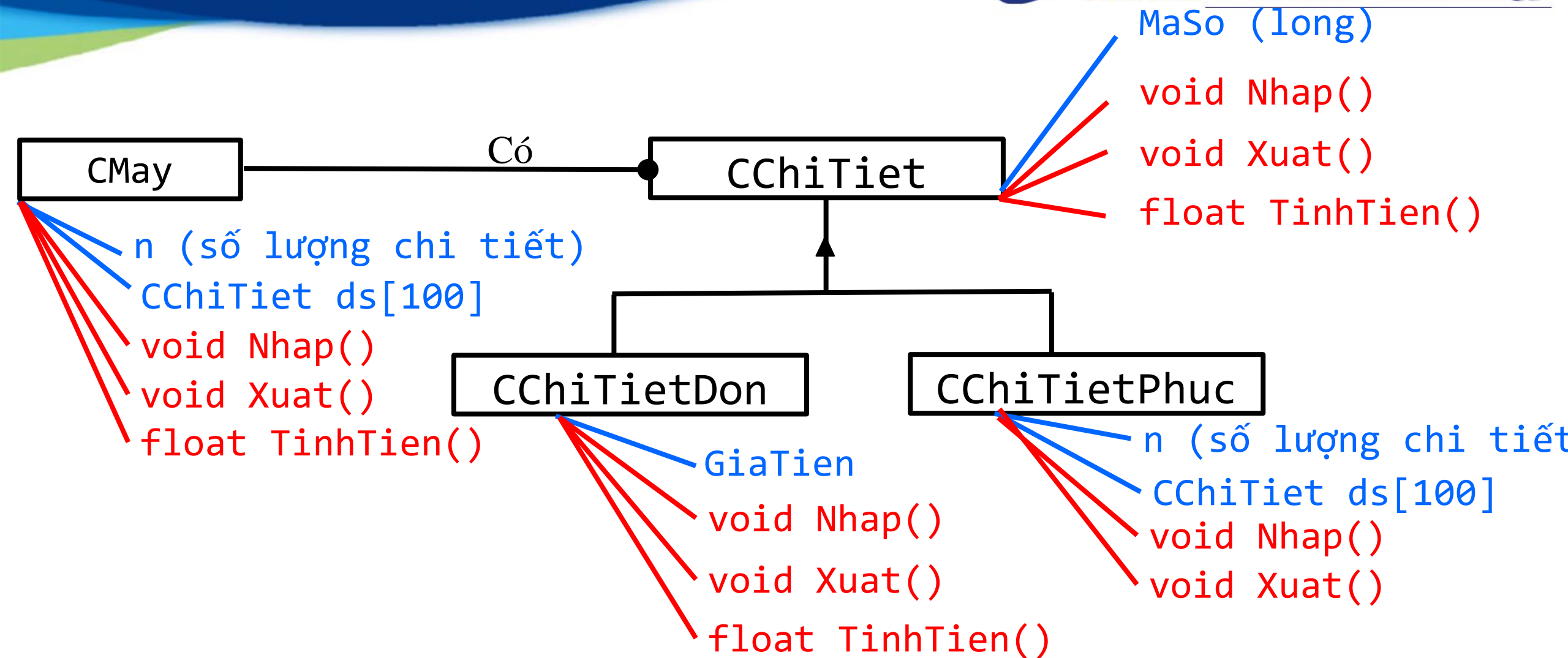
Bài toán

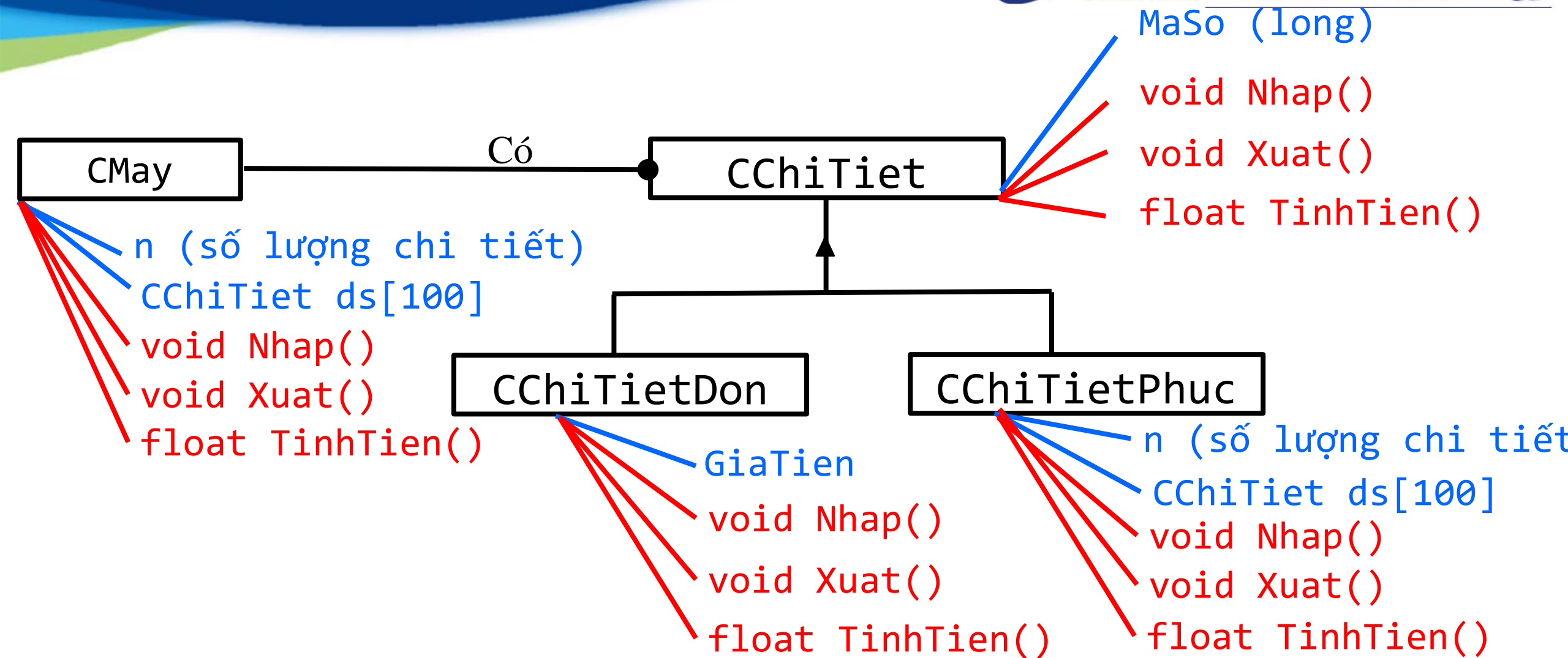
Yêu cầu: Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:

- Nhập các chi tiết cho máy.
- Xuất các chi tiết máy.
- Tính trị giá của máy.
- Tìm kiếm một chi tiết máy theo mã số.
- Đếm số lượng chi tiết đơn có trong cái máy (Bài tập về nhà).





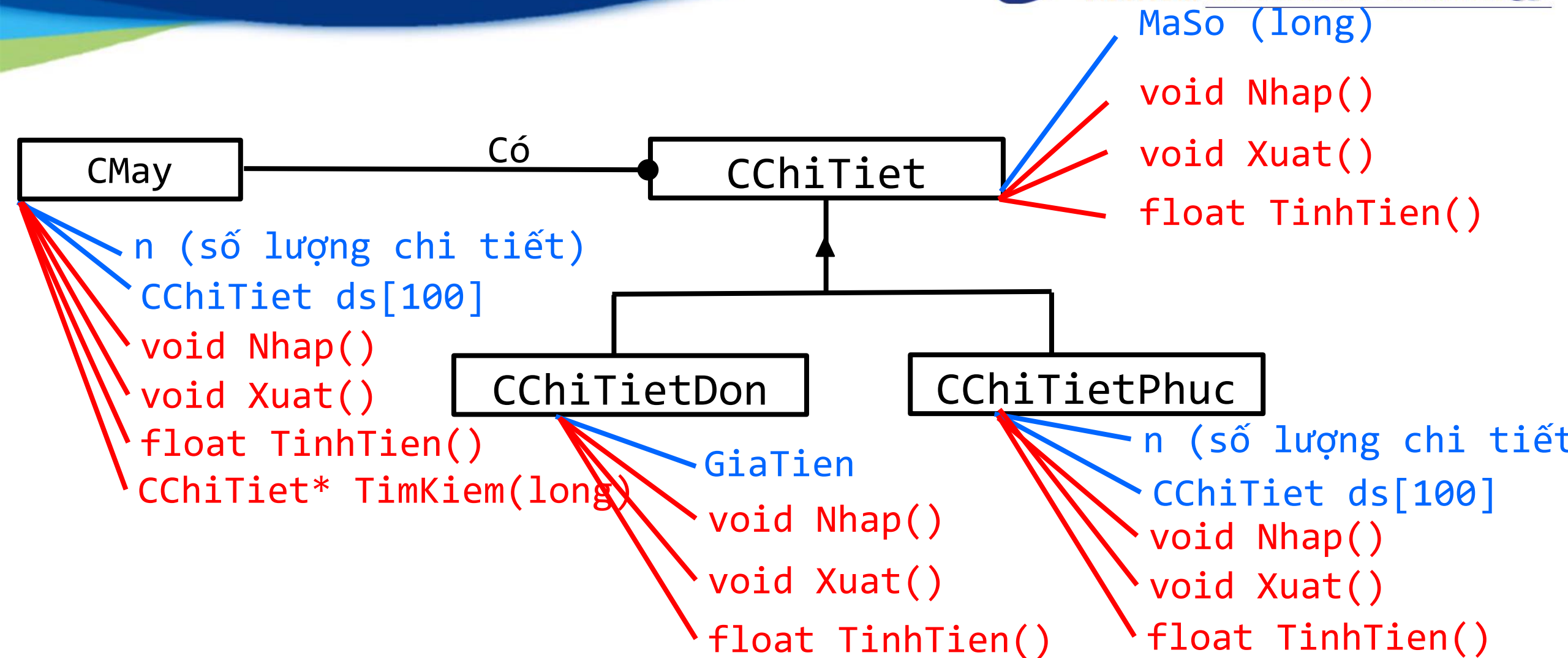




Bài toán

Yêu cầu: Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:

- Nhập các chi tiết cho máy.
- Xuất các chi tiết máy.
- Tính trị giá của máy.
- Tìm kiếm một chi tiết máy theo mã số.
- Đếm số lượng chi tiết đơn có trong cái máy (Bài tập về nhà).



MaSo (long)

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

CMay

Có

CChiTiet

n (số lượng chi tiết)

CChiTiet ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

CChiTietDon

GiaTien

void Nhap()

void Xuat()

float TinhTien()

CChiTietPhuc

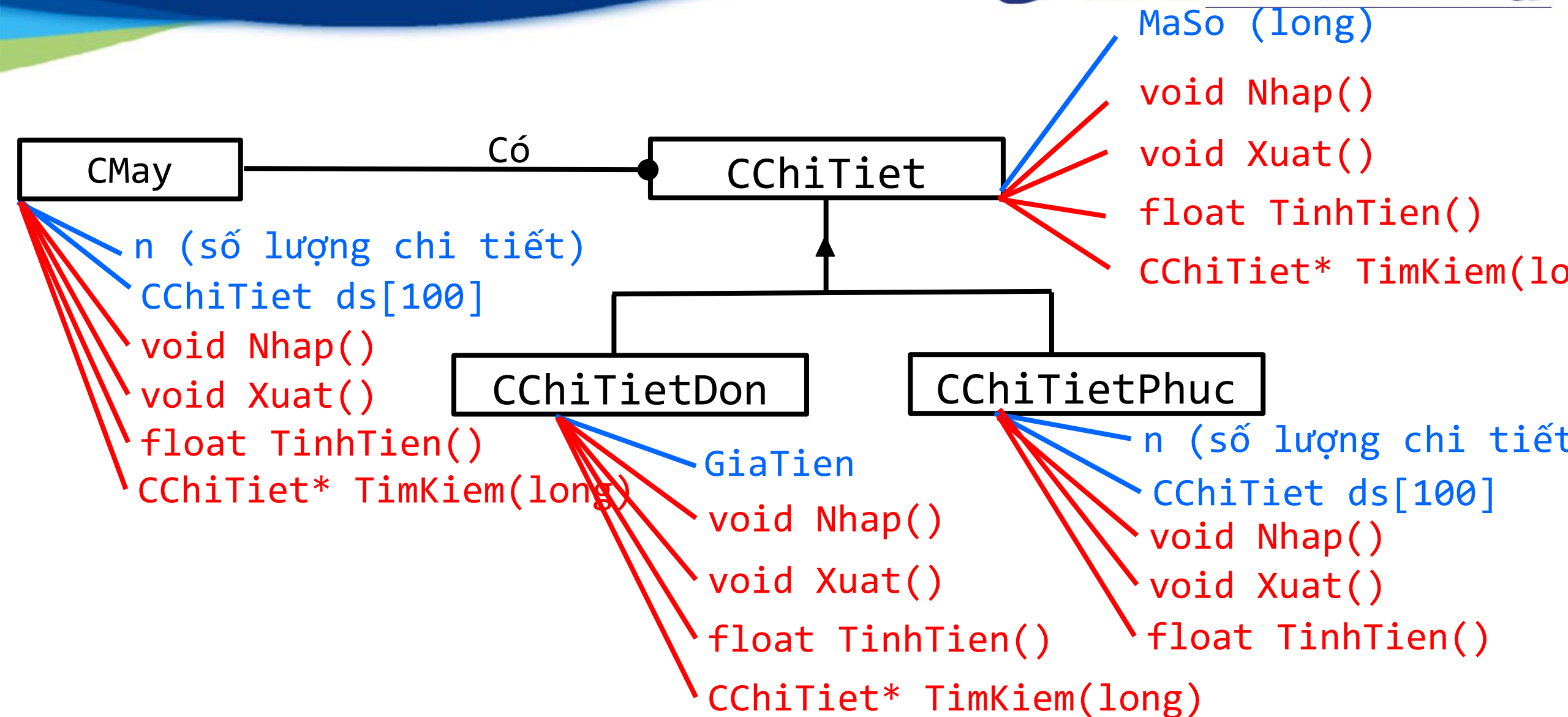
n (số lượng chi tiết)

CChiTiet ds[100]

void Nhap()

void Xuat()

float TinhTien()



MaSo (long)

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

CMay

Có

CChiTiet

n (số lượng chi tiết)

CChiTiet ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

CChiTietDon

GiaTien

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

CChiTietPhuc

n (số lượng chi tiết)

CChiTiet ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Bài toán

Yêu cầu: Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:

- Nhập các chi tiết cho máy.
- **Xuất các chi tiết máy.**
- Tính trị giá của máy.
- **Tìm kiếm một chi tiết máy theo mã số.**
- **Đếm số lượng chi tiết đơn có trong cái máy (Bài tập về nhà).**

KHAI BÁO LỚP

Khai báo lớp

```
11.class CChiTiet
12.{
13.    protected:
14.        long MaSo;
15.    public:
16.        void Nhap();
17.        void Xuat();
18.        float TinhTien();
19.        CChiTiet* TimKiem(long);
20.};
```

CChiTiet

MaSo (long)

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Khai báo lớp

```
11.class CChiTietDon:public CChiTiet
12.{
13.    protected:
14.        float GiaTien;
15.    public:
16.        void Nhap();
17.        void Xuat();
18.        float TinhTien();
19.        CChiTiet* TimKiem(long);
20.};
```

CChiTietDon

GiaTien

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Khai báo lớp

```
11.class CChiTietPhuc:public CChiTiet
12.{
13.    protected:
14.        int n;
15.        CChiTiet ds[100];
16.    public:
17.        void Nhap();
18.        void Xuat();
19.        float TinhTien();
20.        CChiTiet* TimKiem(long);
21.};
```

CChiTietPhuc

n (số lượng chi tiết)

CChiTiet ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Khai báo lớp

```
11.class CMay
12.{
13.    protected:
14.        int n;
15.        CChiTiet ds[100];
16.    public:
17.        void Nhap();
18.        void Xuat();
19.        float TinhTien();
20.        CChiTiet* TimKiem(long);
21.};
```

CMay

n (số lượng chi tiết)

CChiTiet ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

ĐỊNH NGHĨA PHƯƠNG THỨC

Định nghĩa các phương thức

```
11. void CChiTiet::Nhap()  
12. {  
13. |   return;  
14. }  
15. void CChiTiet::Xuat()  
16. {  
17. |   return;  
18. }
```

CChiTiet

MaSo (long)

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Định nghĩa các phương thức

```
11.float CChiTiet::TinhTien()  
12.{  
13.|    return 0;  
14.}  
15.CChiTiet* CChiTiet::TimKiem(long ms)  
16.{  
17.|    if(MaSo==ms)  
18.|        return this;  
19.|    return NULL;  
20.}
```

CChiTiet

MaSo (long)

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Định nghĩa các phương thức

```
11. void CChiTietDon::Nhap()  
12. {  
13.     cout << "Nhap ma so:";  
14.     cin >> MaSo;  
15.     cout << "Nhap gia:";  
16.     cin >> GiaTien;  
17. }
```

```
11. void CChiTietDon::Xuat()  
12. {  
13.     cout << "Ma so:" << MaSo;  
14.     cout << "Gia:" << GiaTien  
15. }
```

CChiTietDon

GiaTien

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Định nghĩa các phương thức

```
11.float CChiTietDon::TinhTien()  
12.{  
13.|    return GiaTien;  
14.}  
15.CChiTiet* CChiTietDon::TimKiem(long ms)  
16.{  
17.|    if(MaSo==ms)  
18.|        return this;  
19.|    return NULL;  
20.}
```

CChiTietDon

GiaTien

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Định nghĩa các phương thức

- Bài toán: Một cái máy có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
 - + Chi tiết đơn: là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: mã số chi tiết, giá tiền.
 - + Chi tiết phức: là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.

Định nghĩa các phương thức

```
11.float CChiTietPhuc::TinhTien()  
12.{  
13.    float s = 0;  
14.    for(int i=0; i<n; i++)  
15.        s = s + ds[i].TinhTien();  
16.    return s;  
17.}
```

CChiTietPhuc

n (số lượng chi tiết)

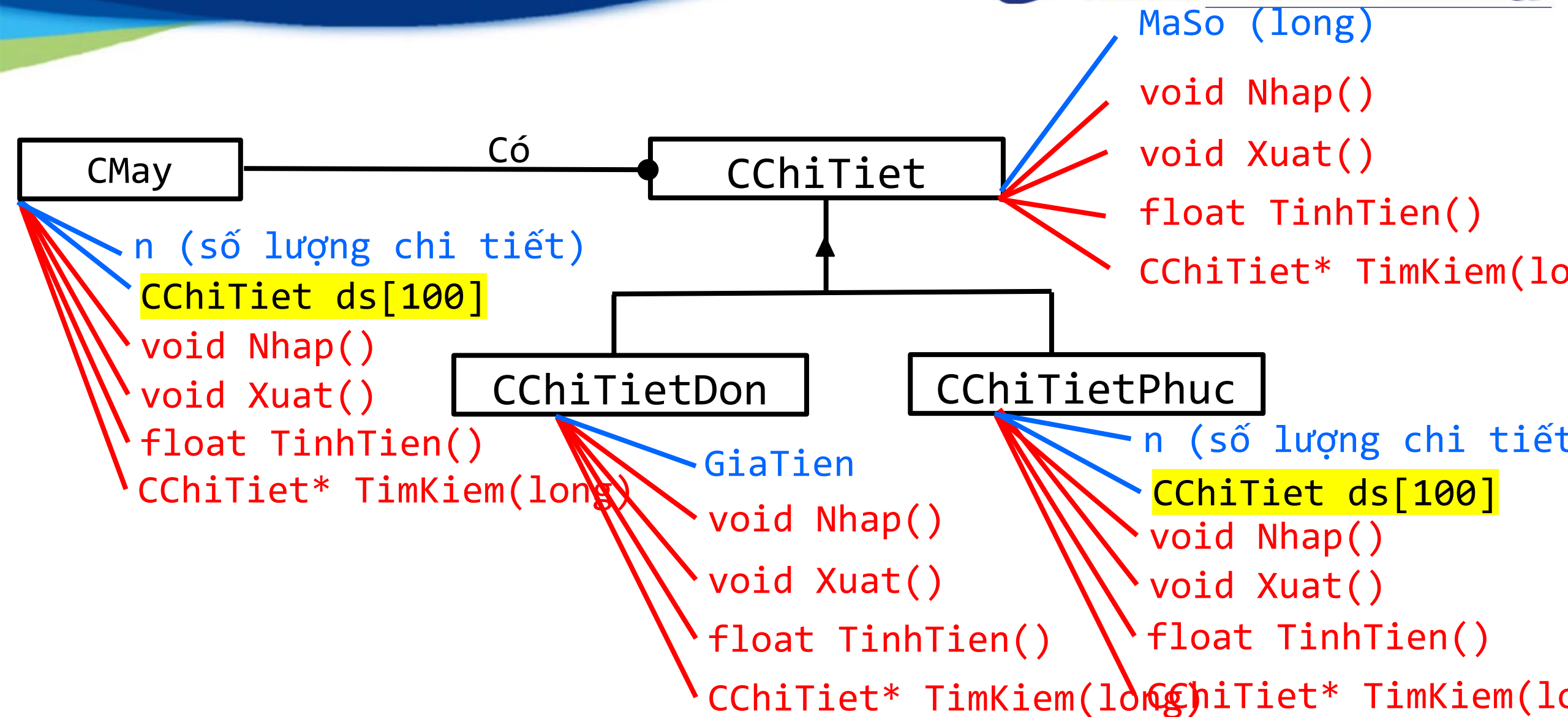
CChiTiet ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)



Định nghĩa các phương thức

```
11.float CChiTiet::TinhTien()  
12.{  
13.|    return 0;  
14.}  
15.CChiTiet* CChiTiet::TimKiem(long ms)  
16.{  
17.|    if(MaSo==ms)  
18.|        return this;  
19.|    return NULL;  
20.}
```

CChiTiet

MaSo (long)

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Định nghĩa các phương thức

```
11.float CChiTietPhuc::TinhTien()  
12.{  
13.    float s = 0;  
14.    for(int i=0; i<n; i++)  
15.        s = s + ds[i].TinhTien();  
16.    return s;  
17.}
```

CChiTietPhuc

n (số lượng chi tiết)

CChiTiet ds[100]

voidNhap()

voidXuat()

floatTinhTien()

CChiTiet* TimKiem(long)

Định nghĩa các phương thức

```
11.float CChiTietPhuc::TinhTien()  
12.{  
13.    float s = 0;  
14.    for(int i=0; i<n; i++)  
15.        s = s + ds[i].TinhTien(): SAI  
16.    return s;  
17.}
```

CChiTietPhuc

n (số lượng chi tiết)

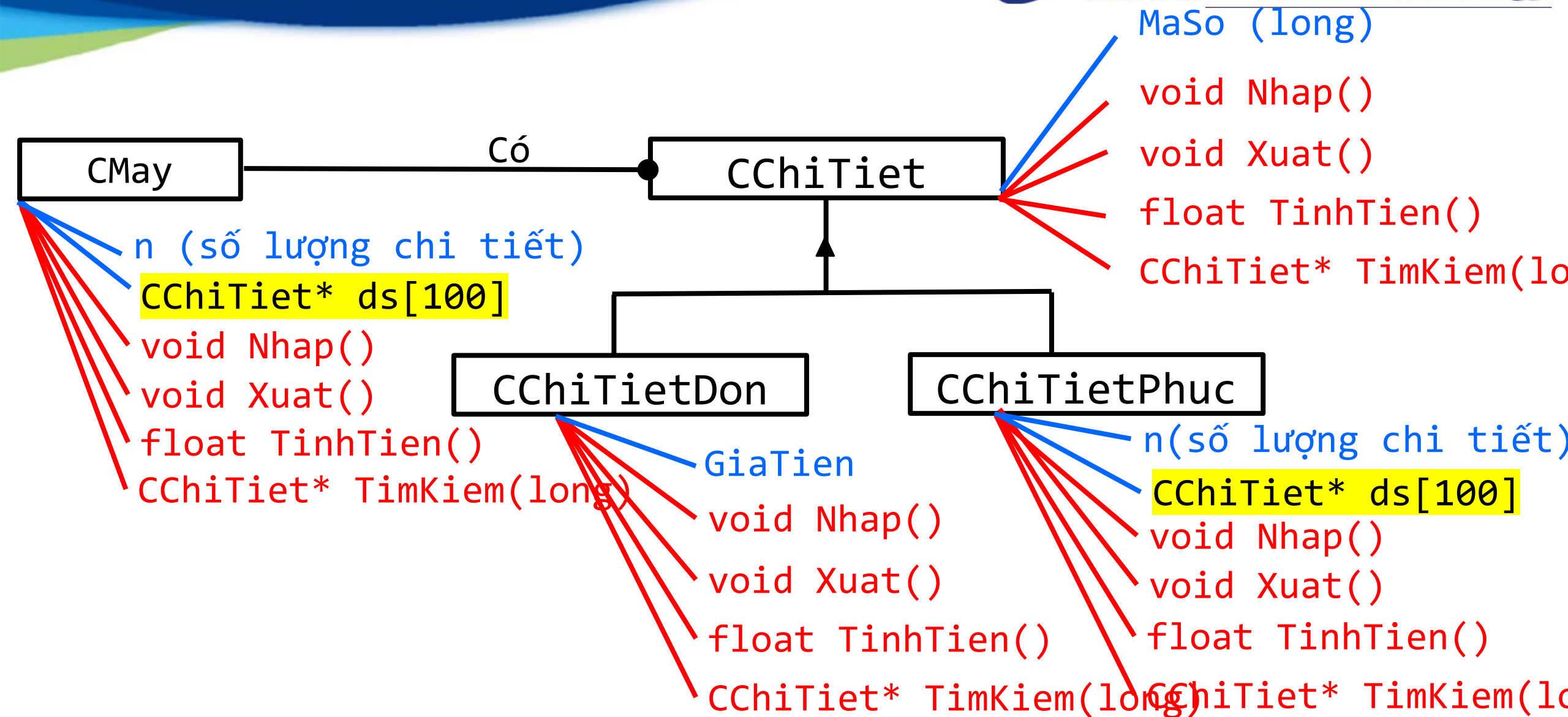
CChiTiet ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)



Khai báo lớp

```
11.class CChiTietPhuc:public CChiTiet
12.{
13.    protected:
14.        int n;
15.        CChiTiet* ds[100];
16.    public:
17.        void Nhap();
18.        void Xuat();
19.        float TinhTien();
20.        CChiTiet* TimKiem(long);
21.};
```

CChiTietPhuc

n (số lượng chi tiết)

CChiTiet* ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Khai báo lớp

```
11.class CMay
12.{
13.    protected:
14.        int n;
15.        CChiTiet* ds[100];
16.    public:
17.        void Nhap();
18.        void Xuat();
19.        float TinhTien();
20.        CChiTiet* TimKiem(long);
21.};
```



n (số lượng chi tiết)

CChiTiet* ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Định nghĩa các phương thức

```
11.float CChiTietPhuc::TinhTien()  
12.{  
13.    float s = 0;  
14.    for(int i=0; i<n; i++)  
15.        s = s + ds[i]->TinhTien();  
16.    return s;  
17.}
```

CChiTietPhuc

n (số lượng chi tiết)

CChiTiet* ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Định nghĩa các phương thức

```
11.float CChiTiet::TinhTien()  
12.{  
13.    return 0;  
14.}  
15.CChiTiet* CChiTiet::TimKiem(long ms)  
16.{  
17.    if(MaSo==ms)  
18.        return this;  
19.    return NULL;  
20.}
```

CChiTiet

MaSo (long)

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Định nghĩa các phương thức

```
11.float CChiTietPhuc::TinhTien()  
12.{  
13.    float s = 0;  
14.    for(int i=0; i<n; i++)  
15.        s = s + ds[i]->TinhTien();  
16.    return s;  
17.}
```

SAI LẦN HẠT

CChiTietPhuc

n (số lượng chi tiết)

CChiTiet* ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Nhắc lại lý thuyết đa xạ

- Khái niệm: Đa xạ là cơ chế **tầm vực động** (dynamic scope) cho phép "**xác định**" đúng hành vi (phương thức – **method**) của đối tượng (**object**) khi yêu cầu thực hiện.
- Việc "**xác định**" được thực hiện theo nguyên tắc tự nhiên: con trỏ đối tượng đang giữ địa chỉ của đối tượng thuộc về lớp nào thì nó sẽ gọi thực hiện phương thức của lớp đối tượng (**class**) đó.
- Tầm vực động (dynamic scope) là cơ chế gọi thực hiện phương thức thông qua con trỏ đối tượng.

Nhắc lại lý thuyết đa xạ

- Một phương thức được khai báo bắt đầu với từ khóa **virtual** thì được gọi là phương thức ảo và phương thức này được gọi thực hiện theo cơ chế đa xạ nếu **lời gọi thực hiện phương thức** được thông qua một con trỏ đối tượng.
- Các phương thức ở lớp dẫn xuất cùng tên và cùng danh sách tham số đầu vào thì cũng sẽ là phương thức ảo nếu ở lớp cơ sở phương thức cùng tên và cùng tham số là phương thức ảo.

MaSo (long)

virtual

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

CMay

Có

CChiTiet

n (số lượng chi tiết)

CChiTiet* ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

CChiTietDon

GiaTien

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

CChiTietPhuc

n(số lượng chi tiết)

CChiTiet* ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Khai báo lớp

```
11.class CChiTiet
12.{
13.    protected:
14.        long MaSo;
15.    public:
16.        virtual void Nhap();
17.        virtual void Xuat();
18.        virtual float TinhTien();
19.        virtual CChiTiet* TimKiem(long);
20.};
```


Khai báo lớp

```
11.class CChiTietPhuc:public CChiTiet
12.{
13.    protected:
14.        int n;
15.        CChiTiet* ds[100];
16.    public:
17.        void Nhap();
18.        void Xuat();
19.        float TinhTien();
20.        CChiTiet* TimKiem(long);
21.};
```

virtual

Các phương thức ở lớp dẫn xuất cùng tên và cùng danh sách tham số đầu vào thì cũng sẽ là phương thức ảo nếu ở lớp cơ sở phương thức cùng tên và cùng tham số là phương thức

CChiTietPhuc

n (số lượng chi tiết)

CChiTiet* ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Khai báo lớp

```
11.class CChiTietDon:public CChiTiet
12.{
13.    protected:
14.        float GiaTien;
15.    public:
16.        void Nhap();
17.        void Xuat();
18.        float TinhTien();
19.        CChiTiet* TimKiem(long);
20.};
```

virtual

Các phương thức ở lớp dẫn xuất cùng tên và cùng danh sách tham số đầu vào thì cũng sẽ là phương thức ảo nếu ở lớp cơ sở phương thức cùng tên và cùng tham số là phương thức ảo.

CChiTietDon

GiaTien

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Định nghĩa các phương thức

```
11.float CChiTietPhuc::TinhTien()  
12.{  
13.    float s = 0;  
14.    for(int i=0; i<n; i++)  
15.        s = s + ds[i]->TinhTien();  
16.    return s;  
17.}
```

CChiTietPhuc

n (số lượng chi tiết)

CChiTiet* ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Định nghĩa các phương thức

```
11.float CChiTietPhuc::TinhTien()  
12.{  
13.    float s = 0;  
14.    for(int i=0; i<n; i++)  
15.        s = s + ds[i]->TinhTien(); ĐÚNG  
16.    return s;  
17.}
```

- Dòng 15 ta nói con trỏ đối tượng `ds[i]` gọi thực thực hiện phương thức `TinhTien` và đây là phương thức ảo nên việc gọi thực hiện phương thức này sẽ theo cơ chế đa xạ.

Định nghĩa các phương thức

```
11.float CChiTietPhuc::TinhTien()  
12.{  
13.    float s = 0;  
14.    for(int i=0; i<n; i++)  
15.        s = s + ds[i]->TinhTien();  
16.    return s;  
17.}
```

Nếu con trỏ đối tượng `ds[i]` giữ địa chỉ của đối tượng thuộc về lớp `CChiTietDon` thì chương trình sẽ gọi thực hiện phương thức tính tiền của lớp `CChiTietDon`.

— Dòng 15 ta nói con trỏ đối tượng `ds[i]` gọi thực thực hiện phương thức `TinhTien` và đây là phương thức ảo nên việc gọi thực hiện phương thức này sẽ theo cơ chế đa xạ.

Định nghĩa các phương thức

```
11.float CChiTietPhuc::TinhTien()  
12.{  
13.    float s = 0;  
14.    for(int i=0; i<n; i++)  
15.        s = s + ds[i]->TinhTien();  
16.    return s;  
17.}
```

Nếu con trỏ đối tượng `ds[i]` giữ địa chỉ của đối tượng thuộc về lớp `CChiTietPhuc` thì chương trình sẽ gọi thực hiện phương thức tính tiền của lớp `CChiTietPhuc`.

- Dòng 15 ta nói con trỏ đối tượng `ds[i]` gọi thực thực hiện phương thức `TinhTien` và đây là phương thức ảo nên việc gọi thực hiện phương thức này sẽ theo cơ chế đa xạ.

Nhắc lại lý thuyết đa xạ

- Khái niệm: Đa xạ là cơ chế **tầm vực động** (dynamic scope) cho phép "**xác định**" đúng hành vi (phương thức – **method**) của đối tượng (**object**) khi yêu cầu thực hiện.
- Việc "**xác định**" được thực hiện theo nguyên tắc tự nhiên: con trỏ đối tượng đang giữ địa chỉ của đối tượng thuộc về lớp nào thì nó sẽ gọi thực hiện phương thức của lớp đối tượng (**class**) đó.
- Tầm vực động (dynamic scope) là cơ chế gọi thực hiện phương thức thông qua con trỏ đối tượng.

Định nghĩa các phương thức

```
11. void CChiTietPhuc::Xuat()  
12. {  
13.     cout << "\n Ma so:" << MaSo;  
14.     cout << "\n So lượng chi tiet thanh phan:" << n;  
15.     for(int i=0; i<n; i++)  
16.         ds[i]->Xuat();  
17. }
```

CChiTietPhuc

n (số lượng chi tiết)

CChiTiet* ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Định nghĩa các phương thức

```
11.CChiTiet* CChiTietPhuc::TimKiem(long ms)
12.{
13.    if(MaSo==ms)
14.        return this;
15.    for(int i=0; i<n; i++)
16.    {
17.        CChiTiet* kq = ds[i]->TimKiem(ms);
18.        if(kq!=NULL)
19.            return kq;
20.    }
21.    return NULL;
22.}
```

n (số lượng chi tiết)

CChiTiet* ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Định nghĩa các phương thức

```
11. void CChiTietPhuc::Nhap()  
12. {  
13.     cout << "Nhap ma so:";  
14.     cin >> MaSo;  
15.     cout << "Nhap so luong chi tiet thanh phan:";  
16.     cin >> n;  
17.     for(int i=0; i<n; i++)  
18.     {  
19.         cout << "Nhap chi tiet a[" << i << "]:";  
20.         ds[i]->Nhap();  
21.     }  
22. }
```

CChiTietPhuc

n (số lượng chi tiết)

CChiTiet* ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Định nghĩa các phương thức

```
11. void CChiTietPhuc::Nhap()  
12. {  
13.     cout << "Nhap ma so:";  
14.     cin >> MaSo;  
15.     cout << "Nhap so luong chi tiet thanh phan:";  
16.     cin >> n;  
17.     for(int i=0; i<n; i++)  
18.     {  
19.         cout << "Nhap chi tiet a[" << i  
20.         | ds[i]->Nhap(); SAI  
21.     }  
22. }
```

CChiTietPhuc

n (số lượng chi tiết)

CChiTiet* ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

```
11. void CChiTietPhuc::Nhap()  
12. {  
13.     cout << "Nhap ma so:";  
14.     cin >> MaSo;  
15.     cout << "Nhap so luong chi tiet thanh phan:";  
16.     cin >> n;  
17.     for(int i=0; i<n; i++)  
18.     {  
19.         int loai;  
20.         cout << "Nhap loai (0. Don, 1. Phuc):";  
21.         cin >> loai;  
22.         switch(loai)  
23.         {  
24.             case 0: ds[i] = new CChiTietDon;  
25.                 break;  
26.             case 1: ds[i] = new CChiTietPhuc;  
27.                 break;  
28.         }  
29.         ds[i]->Nhap();  
30.     }  
31. }
```

CChiTietPhuc

n (số lượng chi tiết)

CChiTiet* ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)


```
11. void CMay::Nhap()  
12. {  
13.     cout << "Nhap so luong chi tiet thanh phan:";  
14.     cin >> n;  
15.     for(int i=0; i<n; i++)  
16.     {  
17.         int loai;  
18.         cout << "Nhap loai (0. Don, 1. Phuc):";  
19.         cin >> loai;  
20.         switch(loai)  
21.         {  
22.             case 0: ds[i] = new CChiTietDon;  
23.                 break;  
24.             case 1: ds[i] = new CChiTietPhuc;  
25.                 break;  
26.         }  
27.         ds[i]->Nhap();  
28.     }  
29. }
```

CChiTietPhuc

n (số lượng chi tiết)

CChiTiet* ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Định nghĩa các phương thức

```
11. void CMay::Xuat()  
12. {  
13.     cout << "\n So lượng chi tiet thanh phan:" << n;  
14.     for(int i=0; i<n; i++)  
15.         ds[i]->Xuat();  
16. }
```

CMay

n (số lượng chi tiết)

CChiTiet* ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Định nghĩa các phương thức

```
11.CChiTiet* CMay::TimKiem(long ms)
12.{
13.    for(int i=0; i<n; i++)
14.    {
15.        CChiTiet* kq = ds[i]->TimKiem(ms);
16.        if(kq!=NULL)
17.            return kq;
18.    }
19.    return NULL;
20.}
```

CMay

n (số lượng chi tiết)

CChiTiet* ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Định nghĩa các phương thức

```
11. float CMay::TinhTien()  
12. {  
13.     float s = 0;  
14.     for(int i=0; i<n; i++)  
15.         s = s + ds[i]->TinhTien();  
16.     return s;  
17. }
```

CMay

n (số lượng chi tiết)

CChiTiet* ds[100]

void Nhap()

void Xuat()

float TinhTien()

CChiTiet* TimKiem(long)

Cảm ơn quý vị đã lắng nghe

Nhóm tác giả

Hồ Thái Ngọc

ThS. Võ Duy Nguyên

TS. Nguyễn Tấn Trần Minh Khang

BÀI TOÁN CHI TIẾT MÁY PYTHON

1. Hồ Thái Ngọc
2. ThS. Võ Duy Nguyên
3. TS. Nguyễn Tấn Trần Minh Khang

BÀI TOÁN

Bài toán

- Bài toán: Một cái máy có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
 - + Chi tiết đơn: là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: mã số chi tiết, giá tiền.
 - + Chi tiết phức: là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.

Bài toán

Yêu cầu: Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:

- Nhập các chi tiết cho máy.
- **Xuất các chi tiết máy.**
- Tính trị giá của máy.
- **Tìm kiếm một chi tiết máy theo mã số.**
- **Đếm số lượng chi tiết đơn có trong cái máy (Bài tập về nhà).**

THIẾT KẾ LỚP SƠ BỘ

Bài toán

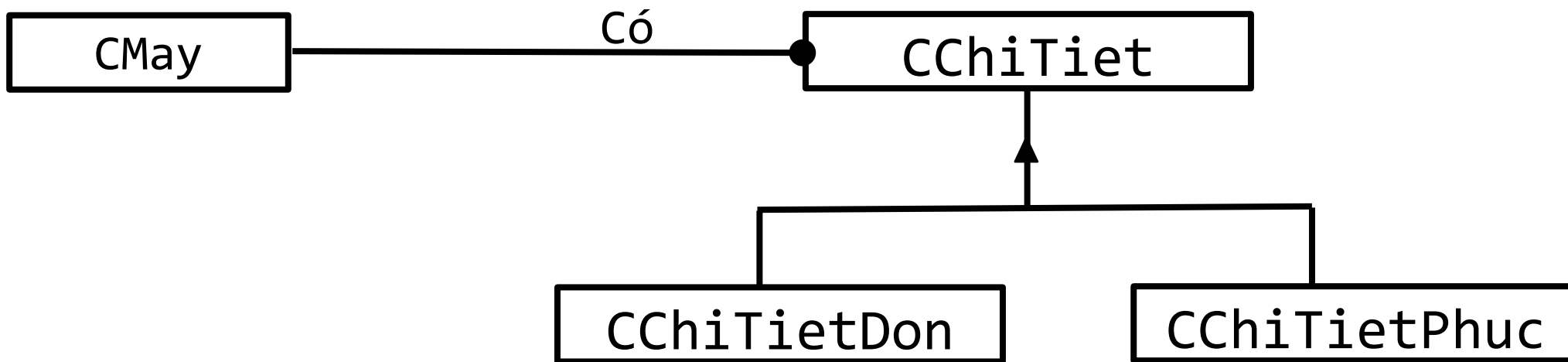
- Bài toán: Một cái máy có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
 - + Chi tiết đơn: là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: mã số chi tiết, giá tiền.
 - + Chi tiết phức: là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.

Bài toán

- Bài toán: Một cái máy có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
- + **Chi tiết đơn**: là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: mã số chi tiết, giá tiền.
- + **Chi tiết phức**: là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.

Bài toán

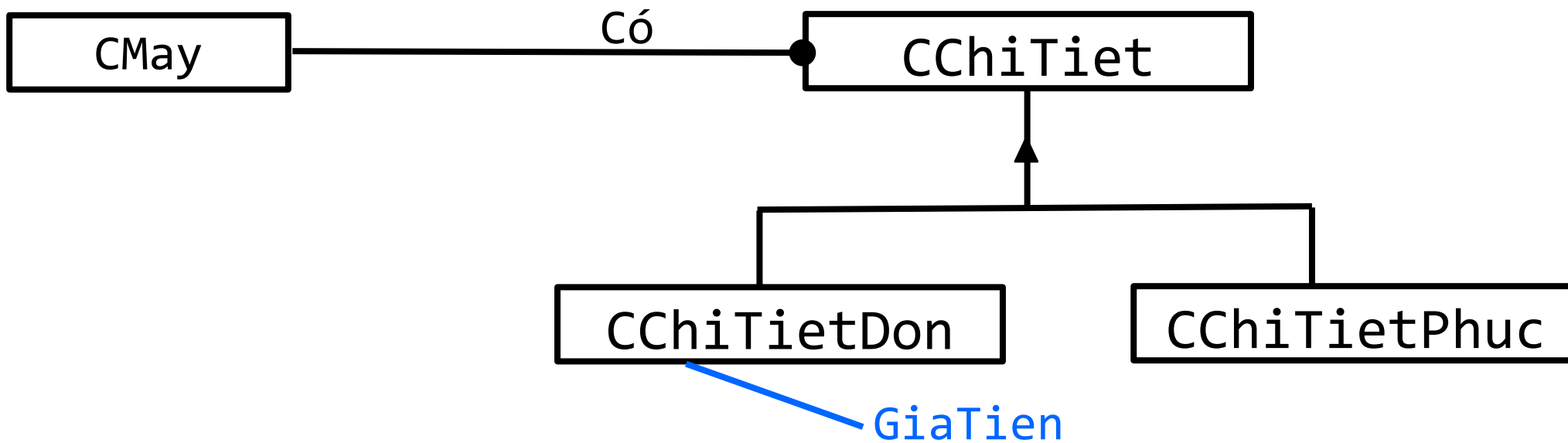
- Bài toán: **Một cái máy** có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
- + **Chi tiết đơn**: là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: mã số chi tiết, giá tiền.
- + **Chi tiết phức**: là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.



THIẾT KẾ LỚP CHI TIẾT

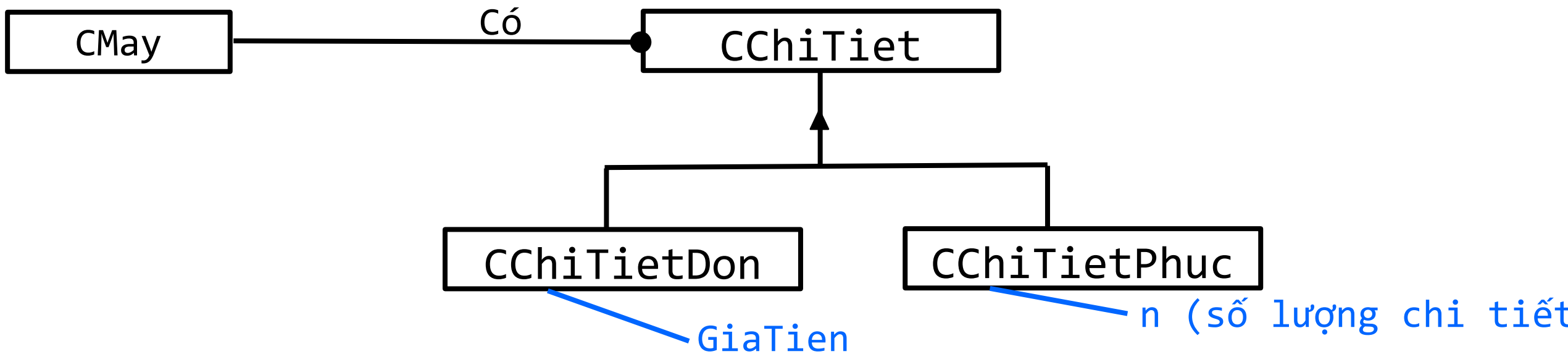
Bài toán

- Bài toán: Một cái máy có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
 - + Chi tiết đơn: là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: mã số chi tiết, giá tiền.
 - + Chi tiết phức: là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.



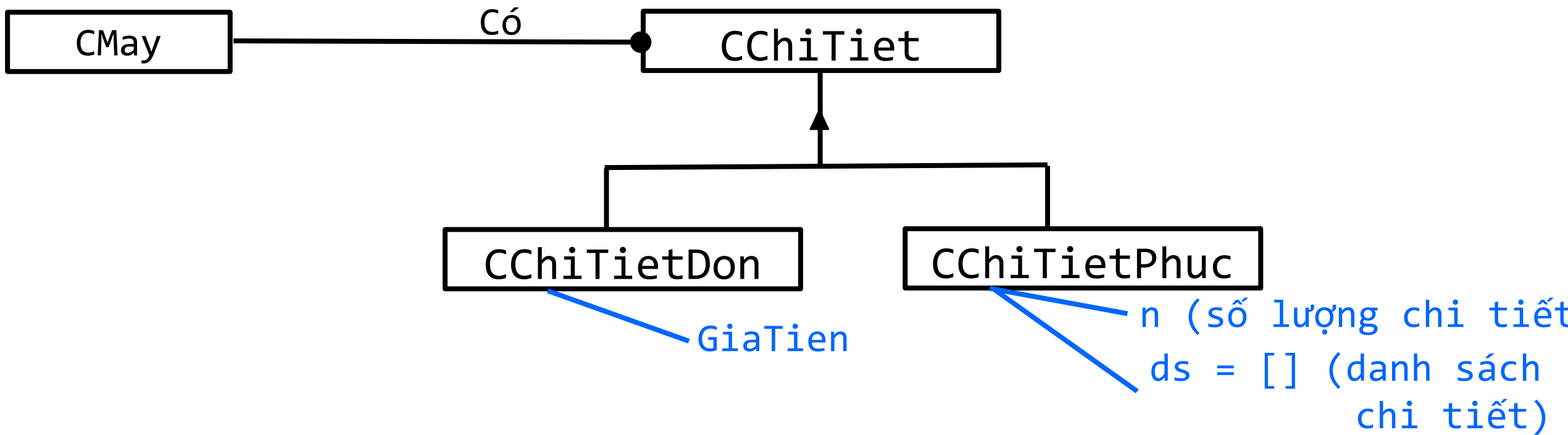
Bài toán

- Bài toán: Một cái máy có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
 - + Chi tiết đơn: là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: mã số chi tiết, giá tiền.
 - + Chi tiết phức: là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.



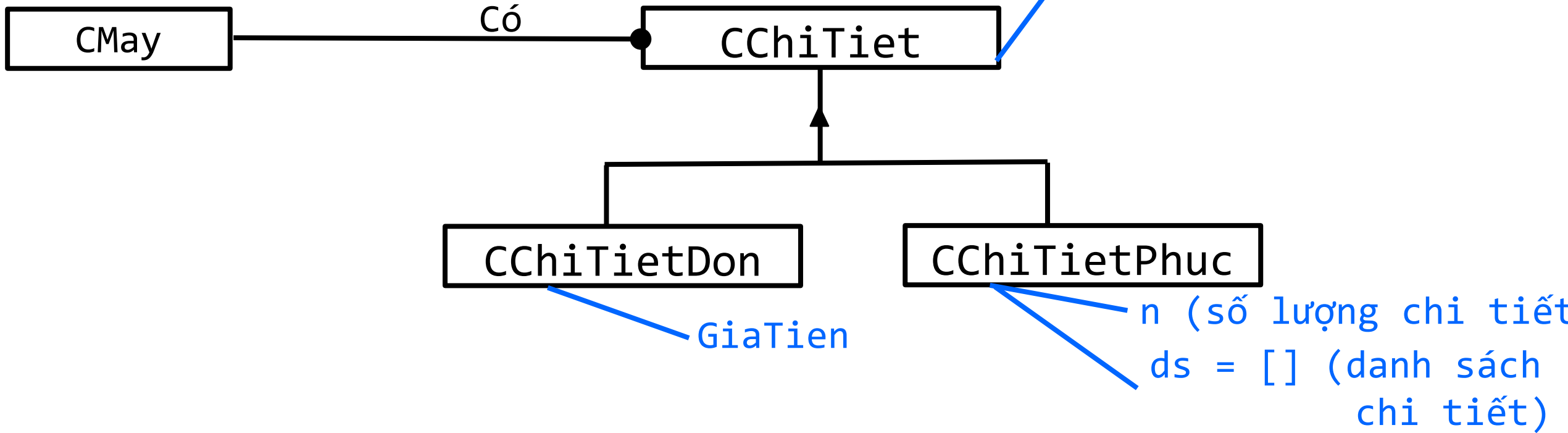
Bài toán

- Bài toán: Một cái máy có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
 - + Chi tiết đơn: là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: mã số chi tiết, giá tiền.
 - + Chi tiết phức: là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.



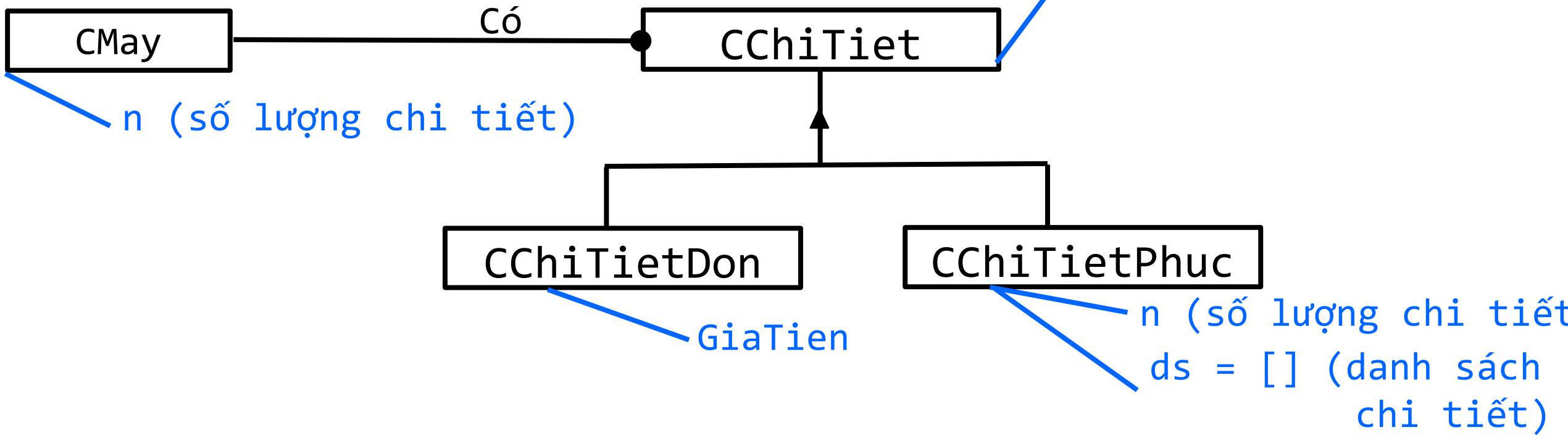
Bài toán

- Bài toán: Một cái máy có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
 - + Chi tiết đơn: là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: mã số chi tiết, giá tiền.
 - + Chi tiết phức: là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.



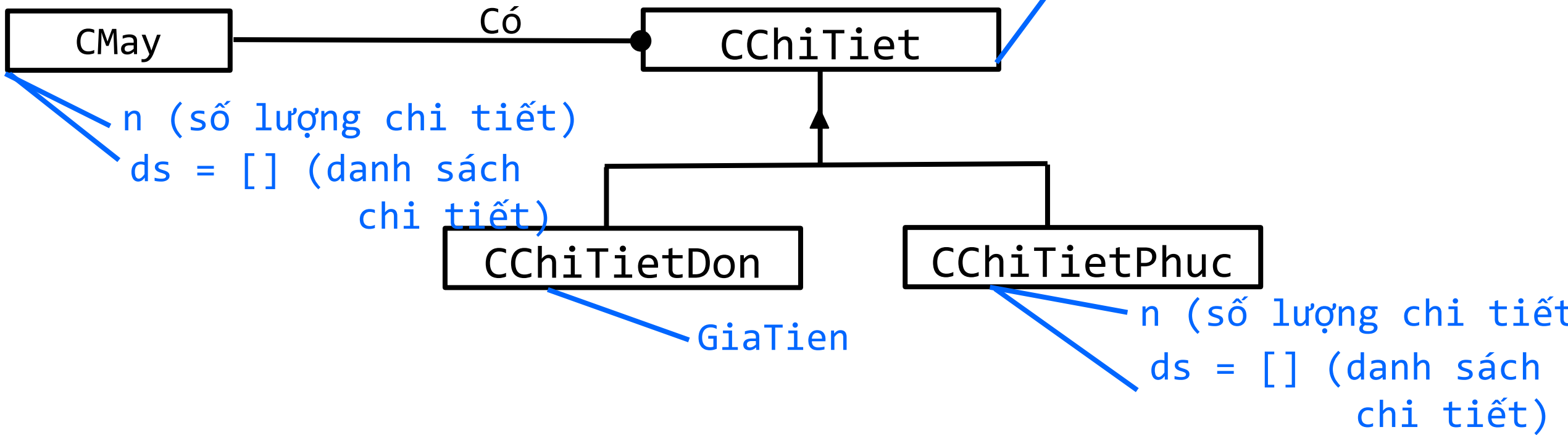
Bài toán

- Bài toán: Một cái máy có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
 - + Chi tiết đơn: là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: mã số chi tiết, giá tiền.
 - + Chi tiết phức: là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.



Bài toán

- Bài toán: Một cái máy có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
 - + Chi tiết đơn: là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: mã số chi tiết, giá tiền.
 - + Chi tiết phức: là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.



Bài toán

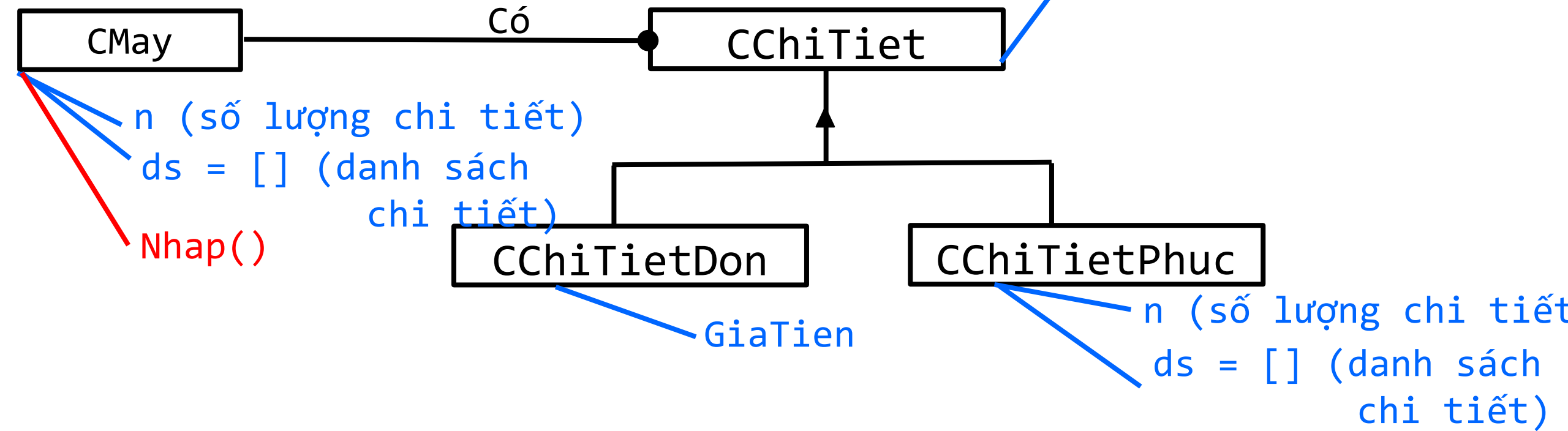
Yêu cầu: Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:

- Nhập các chi tiết cho máy.
- **Xuất các chi tiết máy.**
- Tính trị giá của máy.
- **Tìm kiếm một chi tiết máy theo mã số.**
- **Đếm số lượng chi tiết đơn có trong cái máy (Bài tập về nhà).**

Bài toán

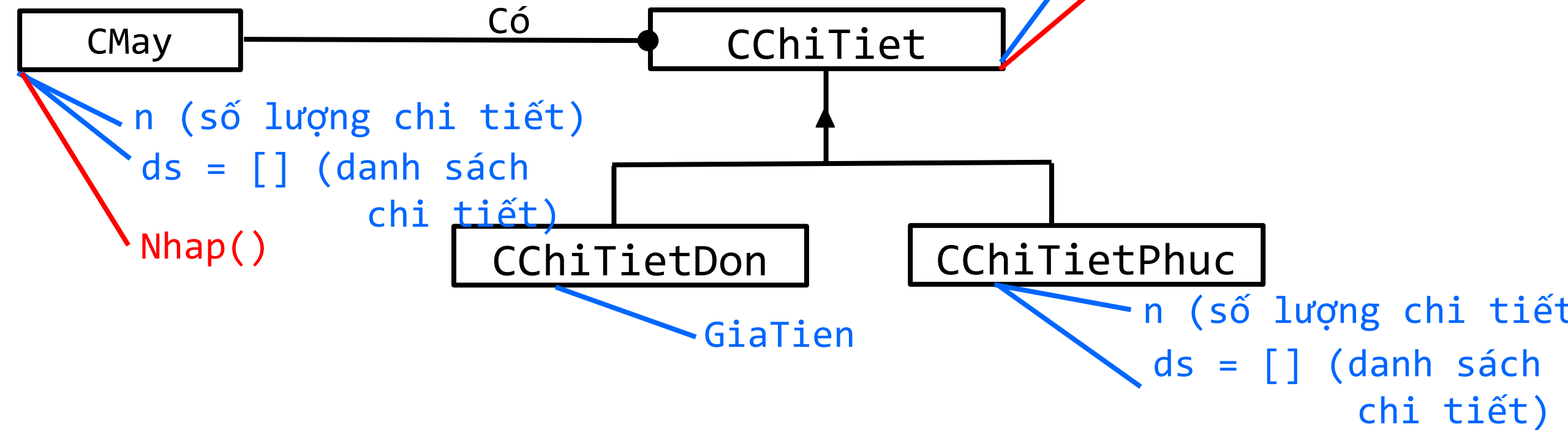
Yêu cầu: Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:

- Nhập các chi tiết cho máy.
- Xuất các chi tiết máy.
- Tính trị giá của máy.
- Tìm kiếm một chi tiết máy theo mã số.
- Đếm số lượng chi tiết đơn có trong cái máy (Bài tập về nhà).



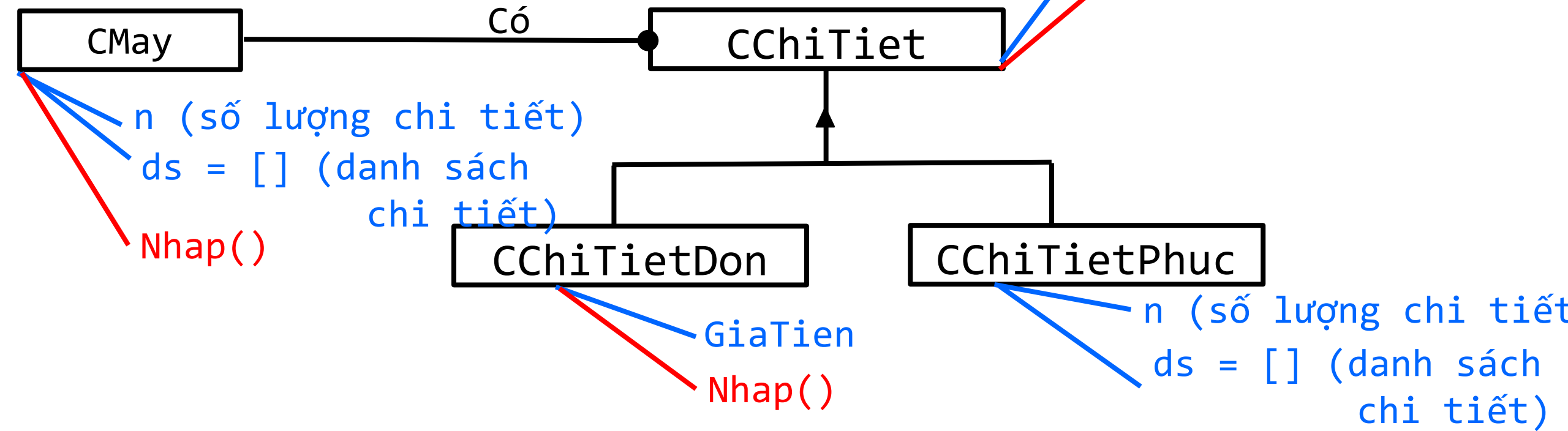
MaSo

Nhap()



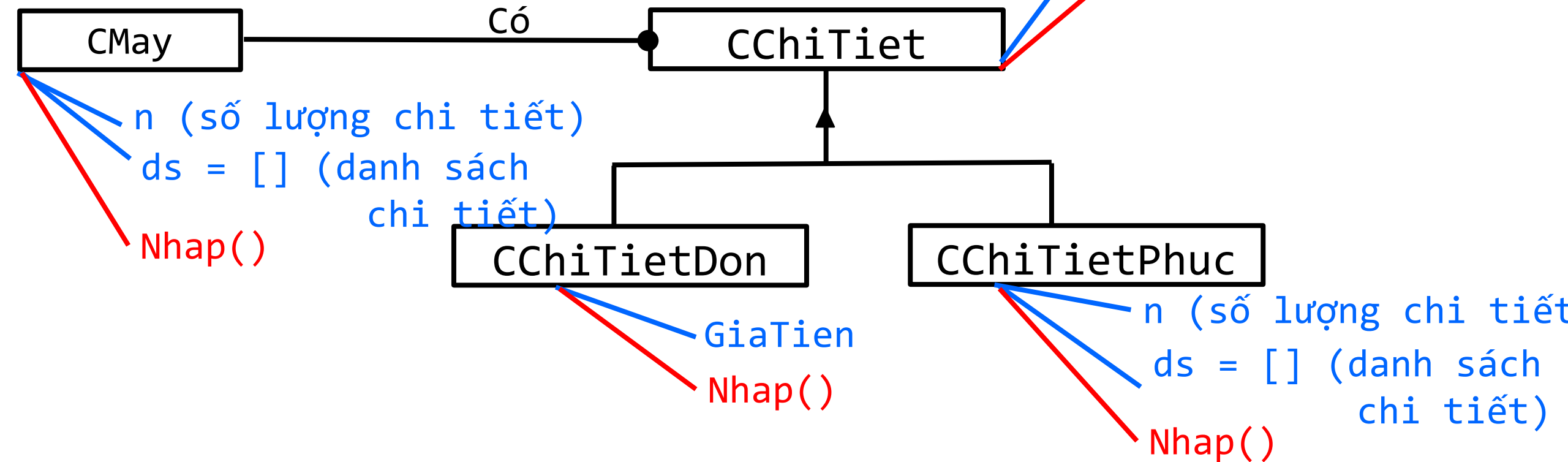
MaSo

Nhap()



MaSo

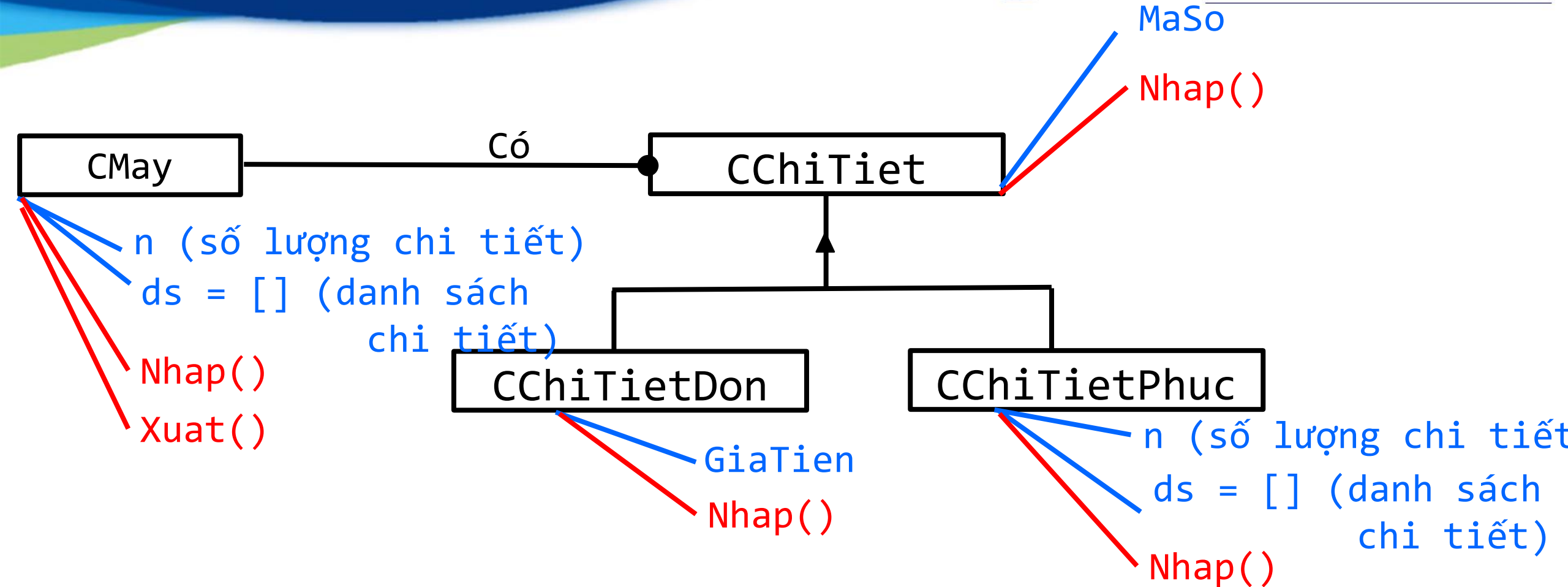
Nhap()

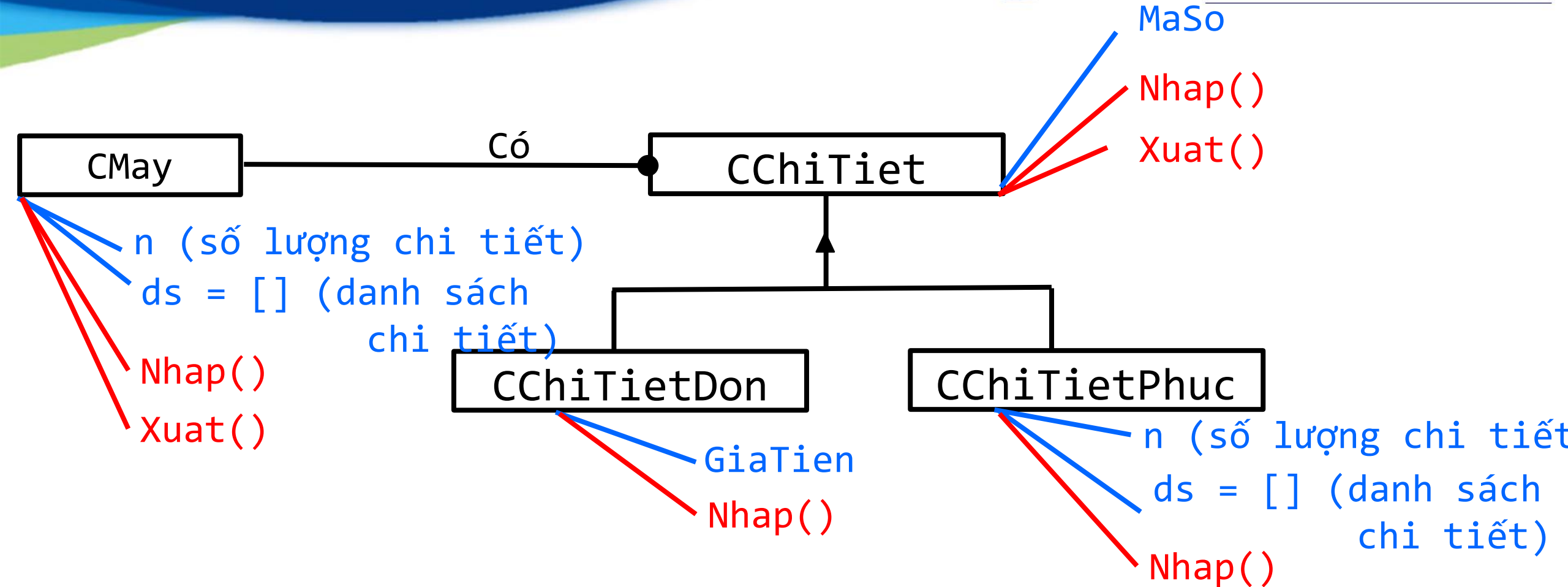


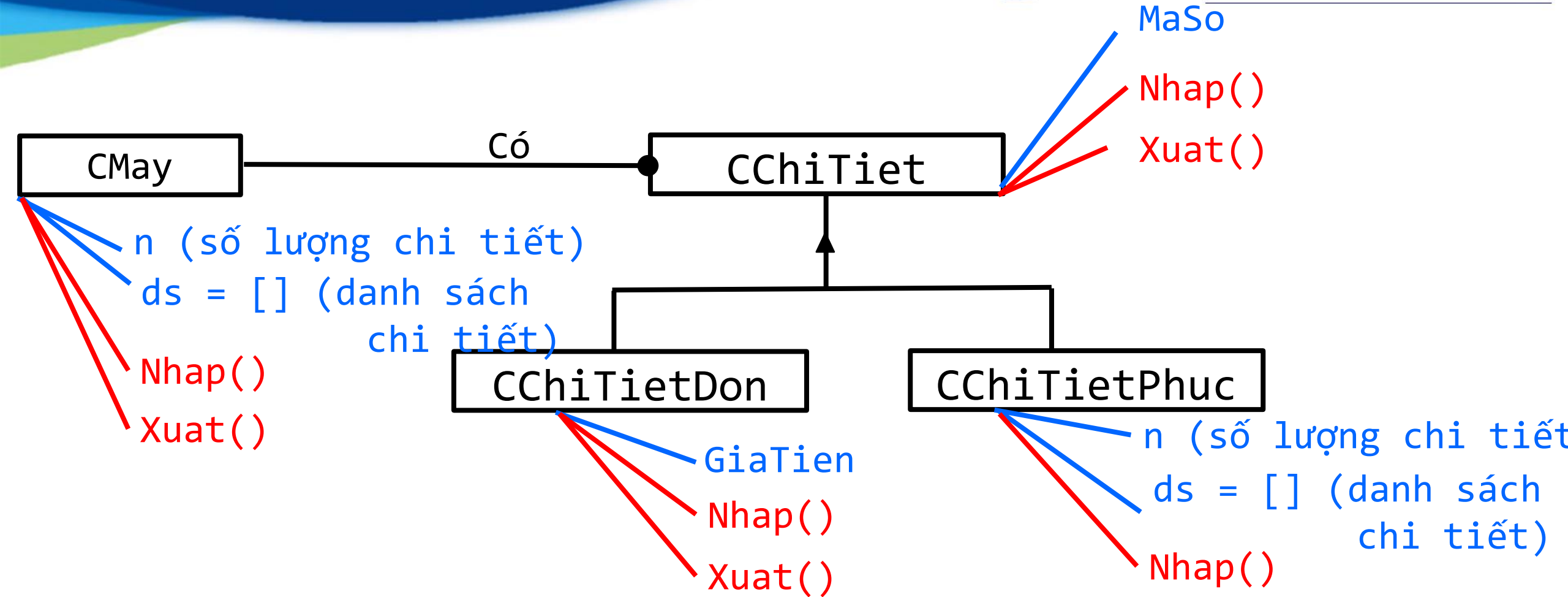
Bài toán

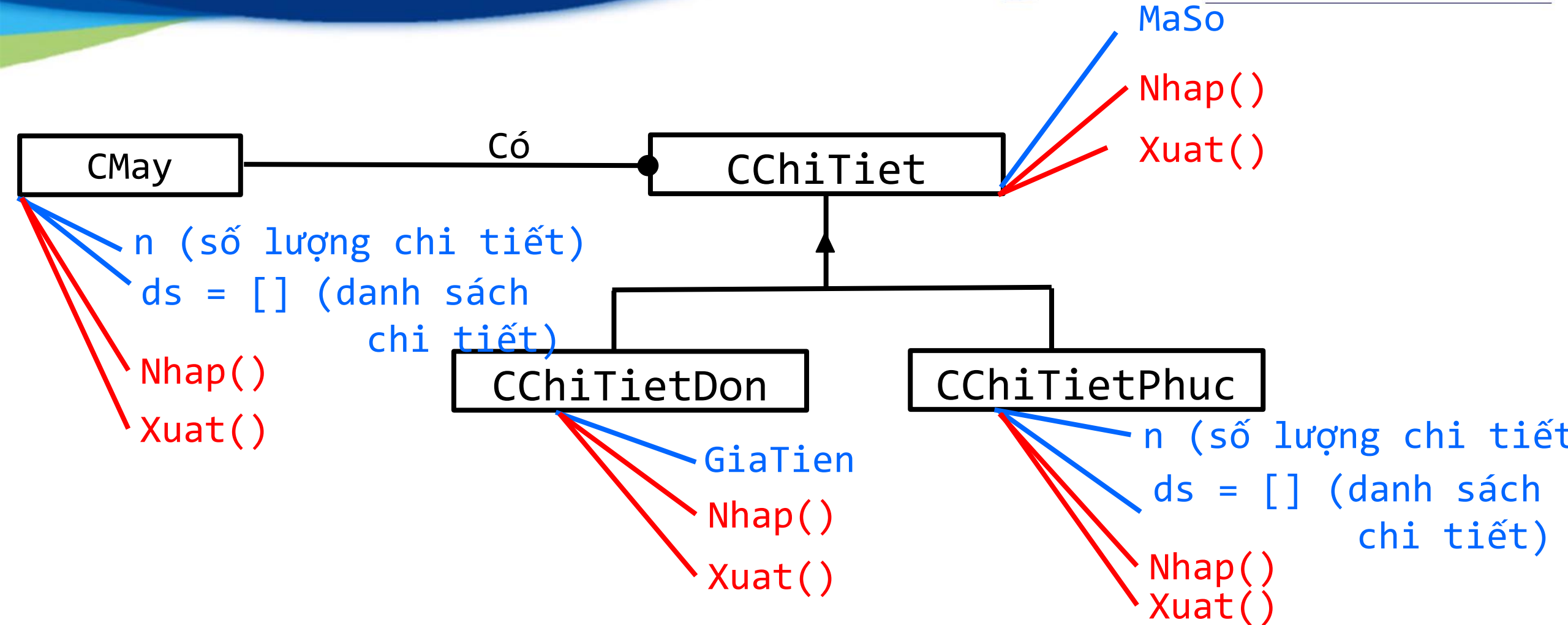
Yêu cầu: Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:

- Nhập các chi tiết cho máy.
- Xuất các chi tiết máy.
- Tính trị giá của máy.
- Tìm kiếm một chi tiết máy theo mã số.
- Đếm số lượng chi tiết đơn có trong cái máy (Bài tập về nhà).





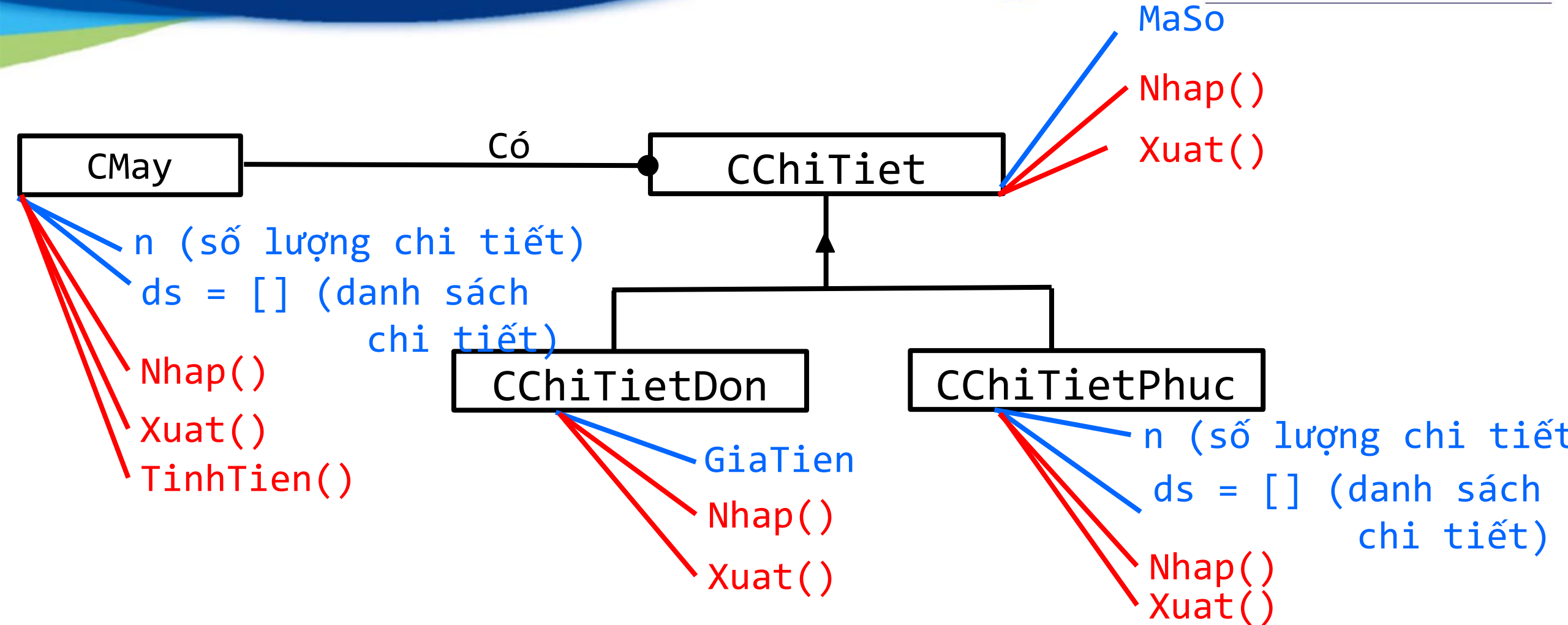


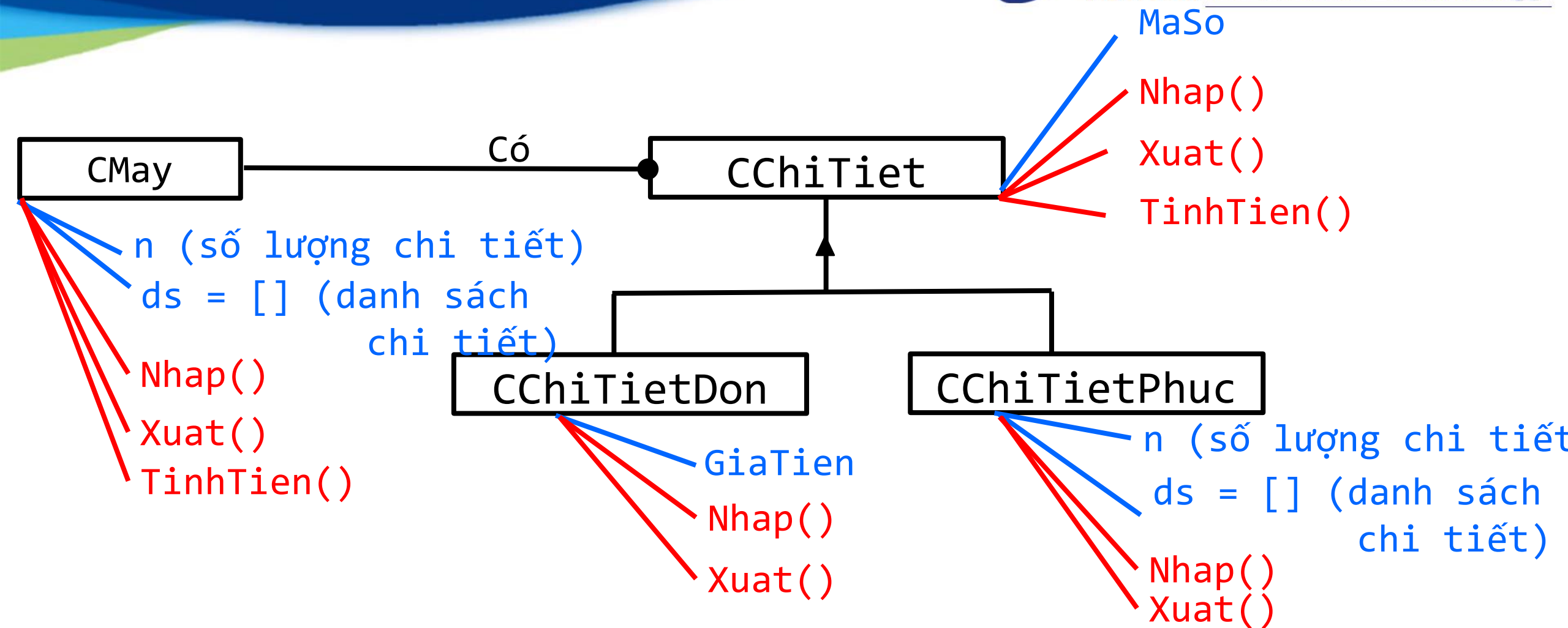


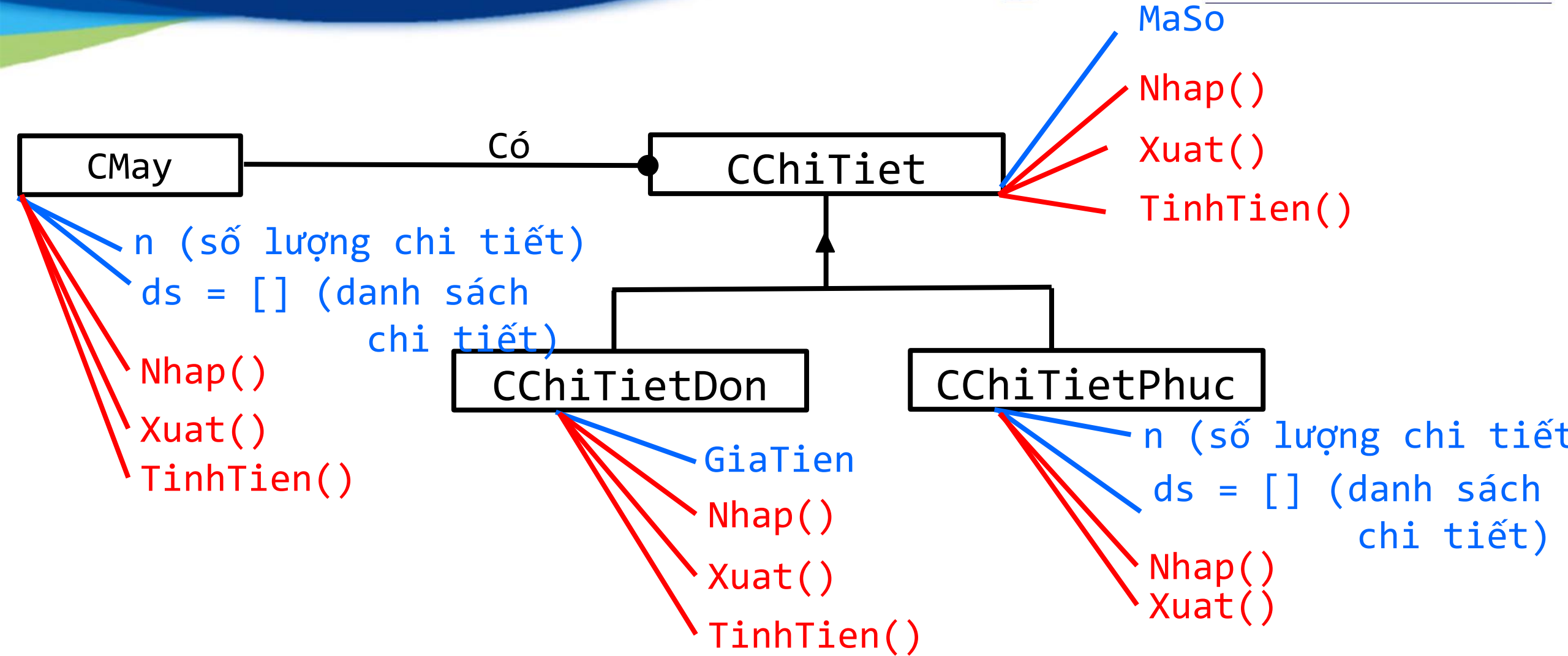
Bài toán

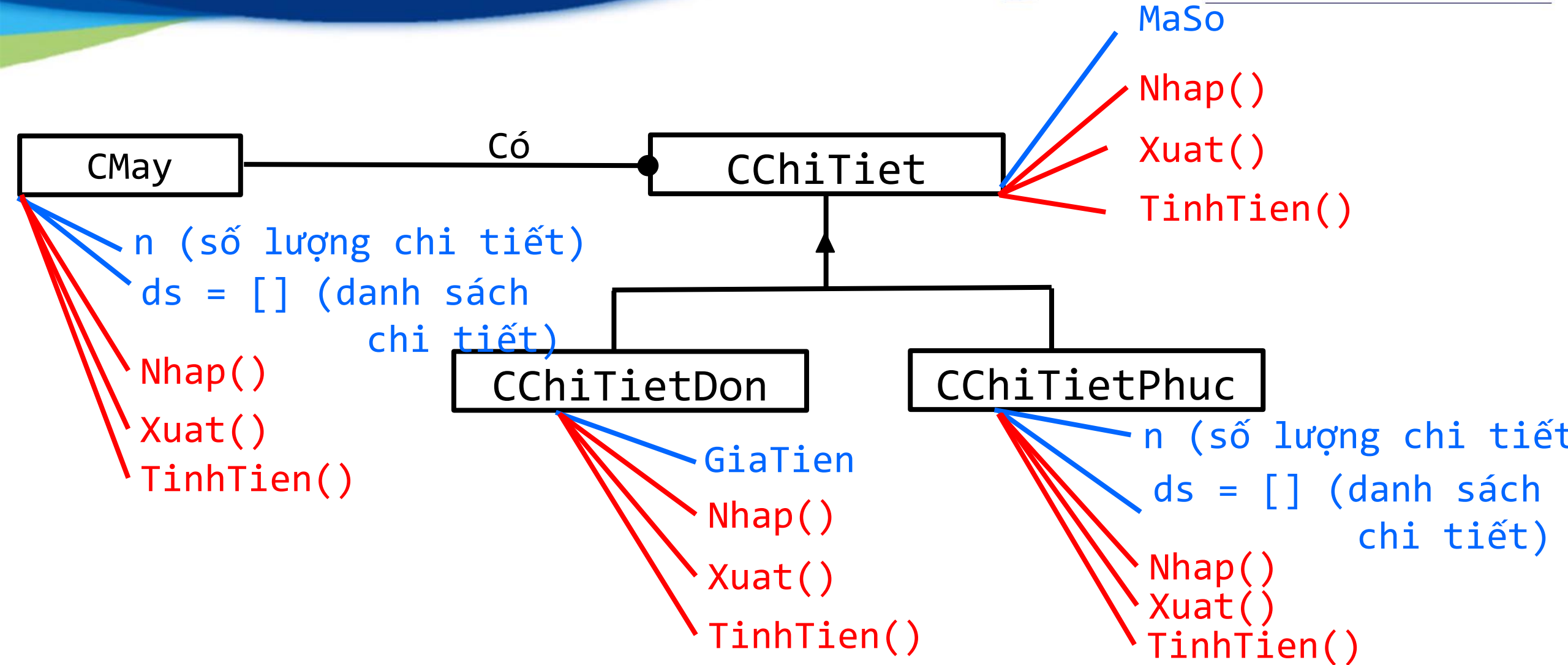
Yêu cầu: Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:

- Nhập các chi tiết cho máy.
- Xuất các chi tiết máy.
- Tính trị giá của máy.
- Tìm kiếm một chi tiết máy theo mã số.
- Đếm số lượng chi tiết đơn có trong cái máy (Bài tập về nhà).





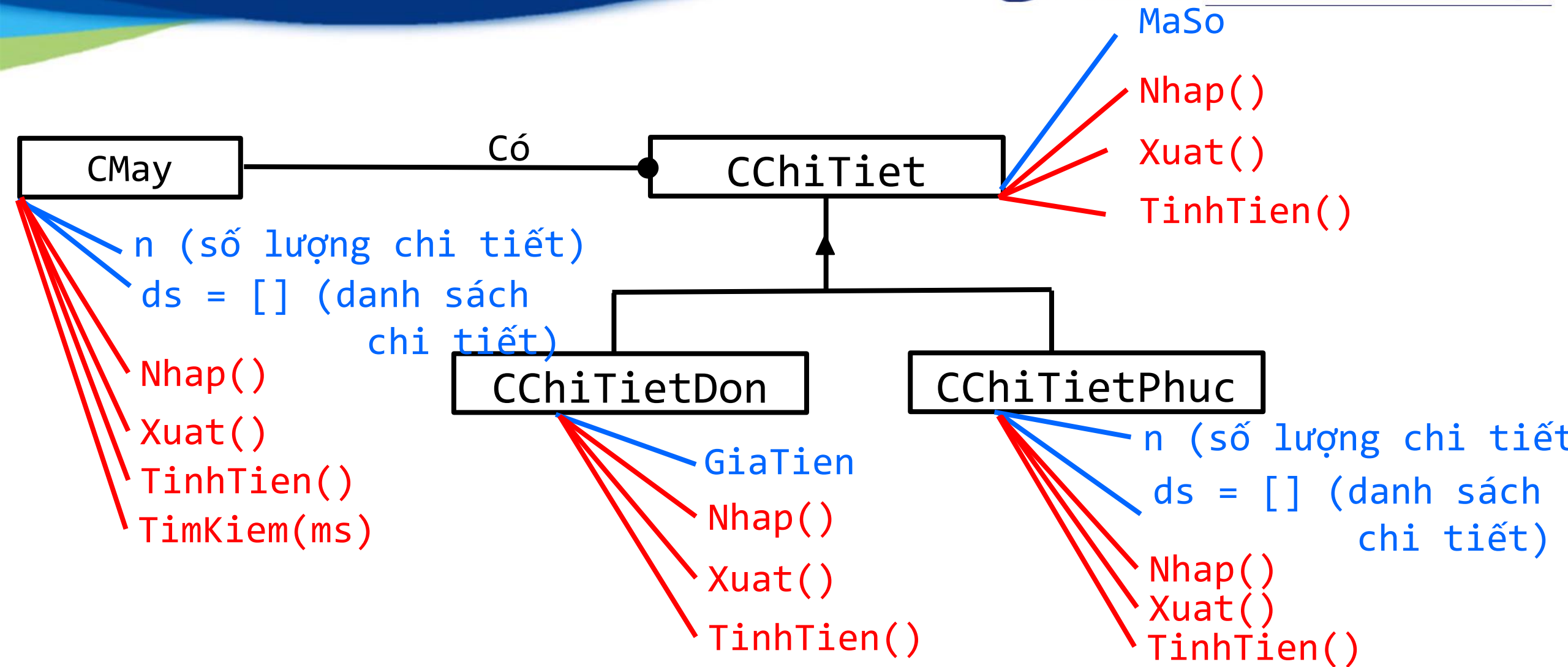


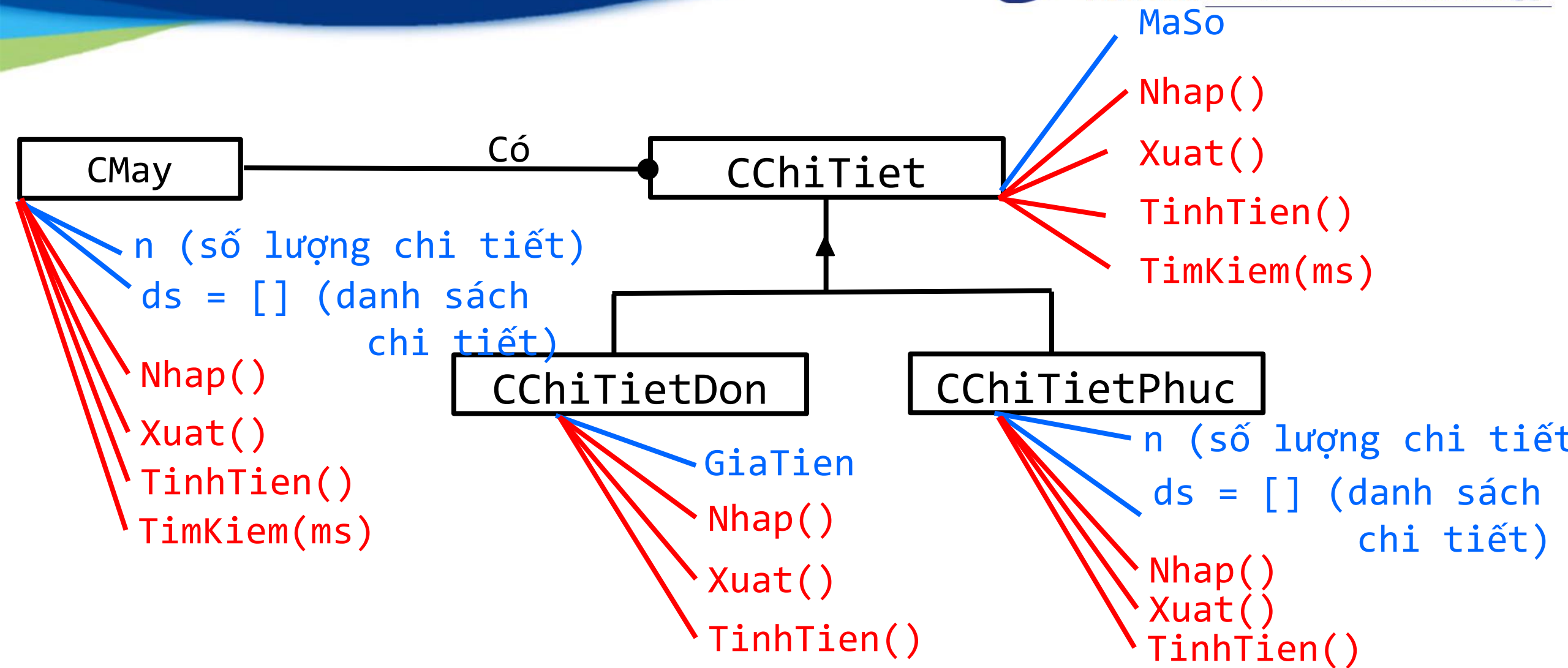


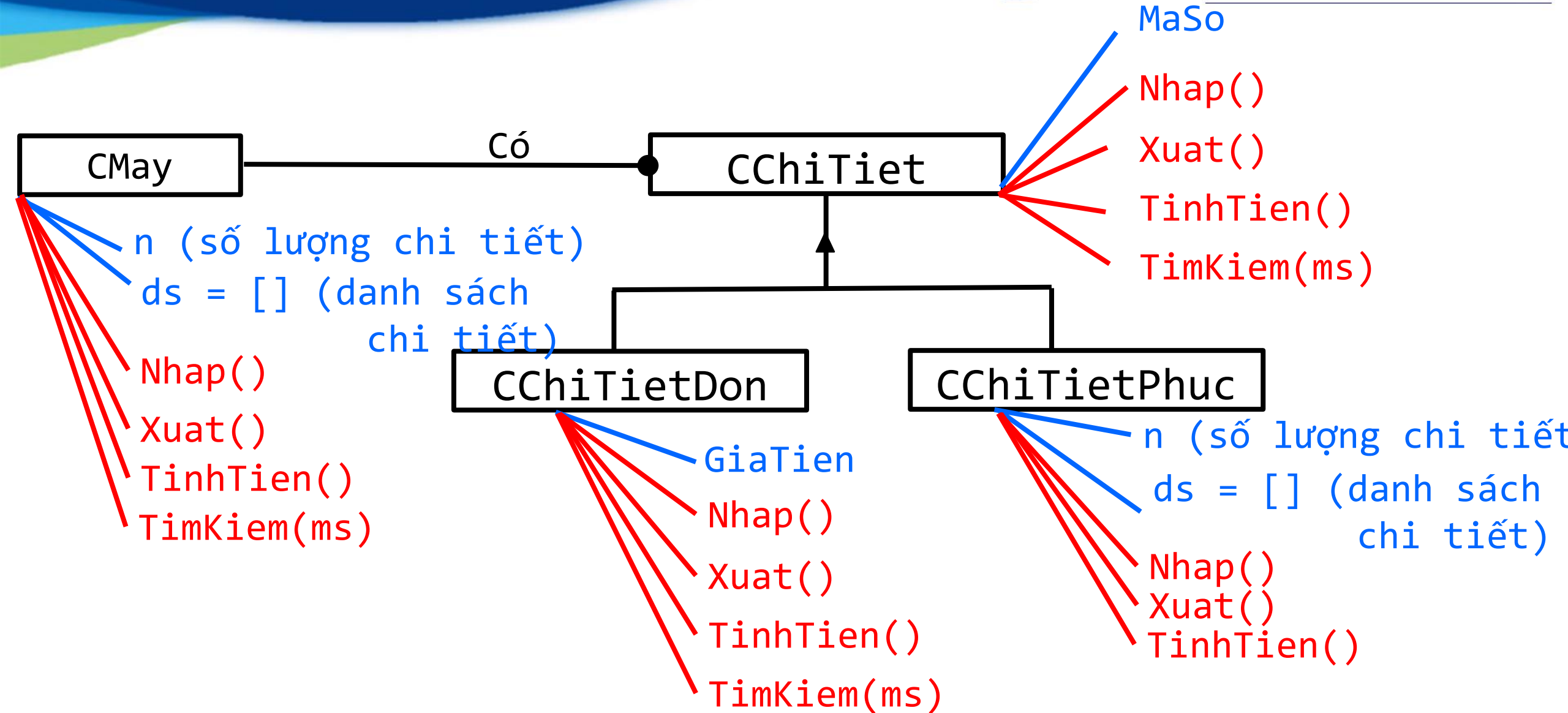
Bài toán

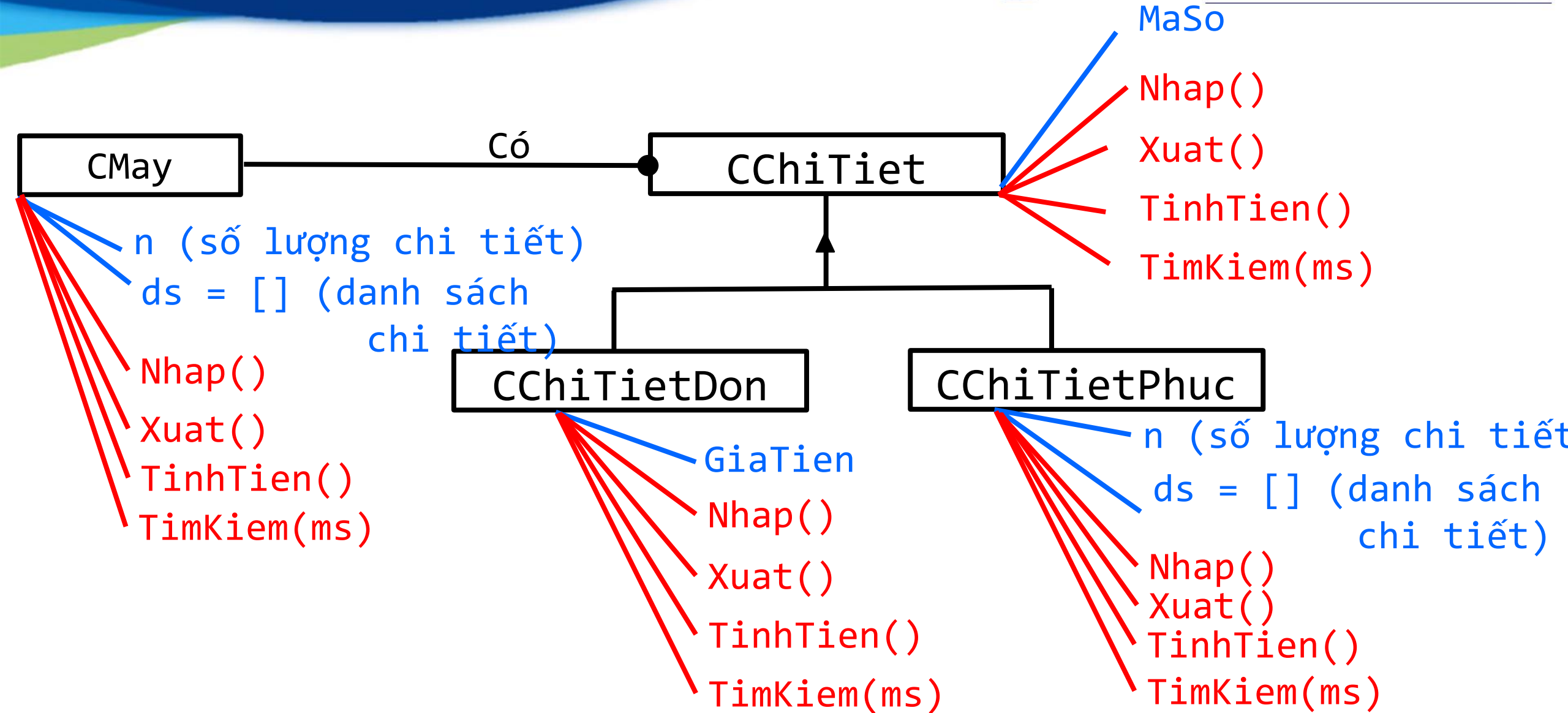
Yêu cầu: Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:

- Nhập các chi tiết cho máy.
- Xuất các chi tiết máy.
- Tính trị giá của máy.
- Tìm kiếm một chi tiết máy theo mã số.
- Đếm số lượng chi tiết đơn có trong cái máy (Bài tập về nhà).









Bài toán

Yêu cầu: Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:

- Nhập các chi tiết cho máy.
- **Xuất các chi tiết máy.**
- Tính trị giá của máy.
- **Tìm kiếm một chi tiết máy theo mã số.**
- **Đếm số lượng chi tiết đơn có trong cái máy (Bài tập về nhà).**

ĐỊNH NGHĨA PHƯƠNG THỨC KHỞI TẠO

Định nghĩa phương thức khởi tạo

```
1. class CChiTiet:  
2.     def __init__(self):  
3.         self._MaSo = 0
```

CChiTiet

MaSo

Nhap()

Xuat()

TinhTien()

TimKiem(ms)

Định nghĩa phương thức khởi tạo

```
1. class CChiTietDon(CChiTiet):  
2.     def __init__(self):  
3.         super().__init__()  
4.         self._GiaTien = 0
```

CChiTietDon

GiaTien

Nhap()

Xuat()

TinhTien()

TimKiem(ms)

Định nghĩa phương thức khởi tạo

```
1. class CChiTietPhuc(CChiTiet):  
2.     def __init__(self):  
3.         super().__init__()  
4.         self._n = 0  
5.         self._ds = []
```

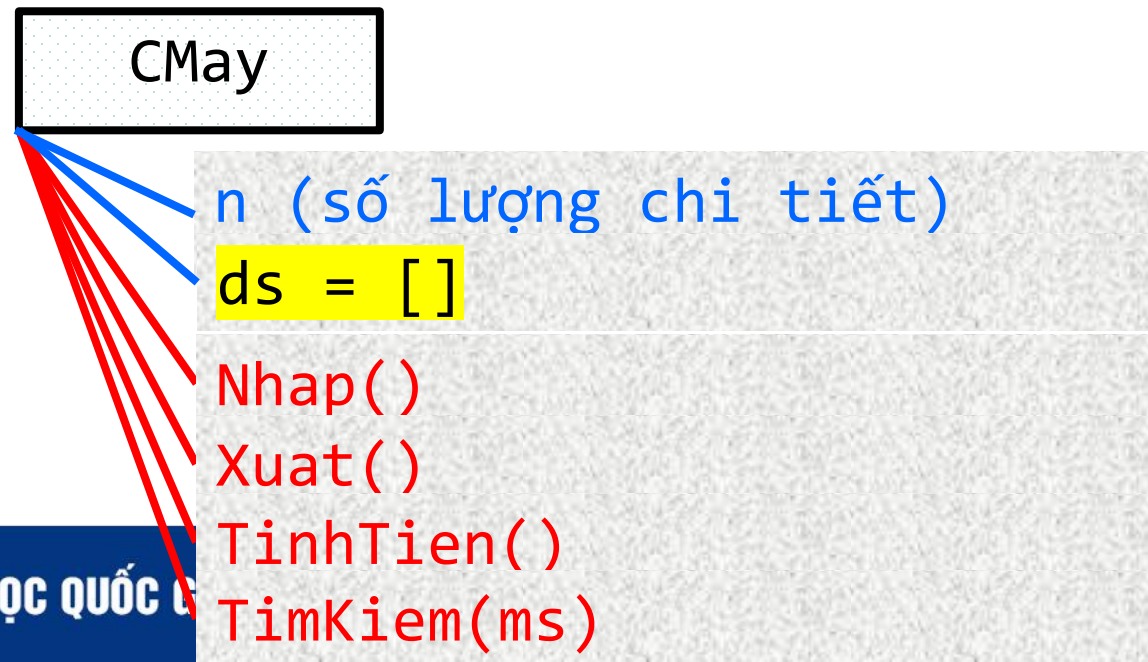
CChiTietPhuc

n (số lượng chi tiết)
ds = []

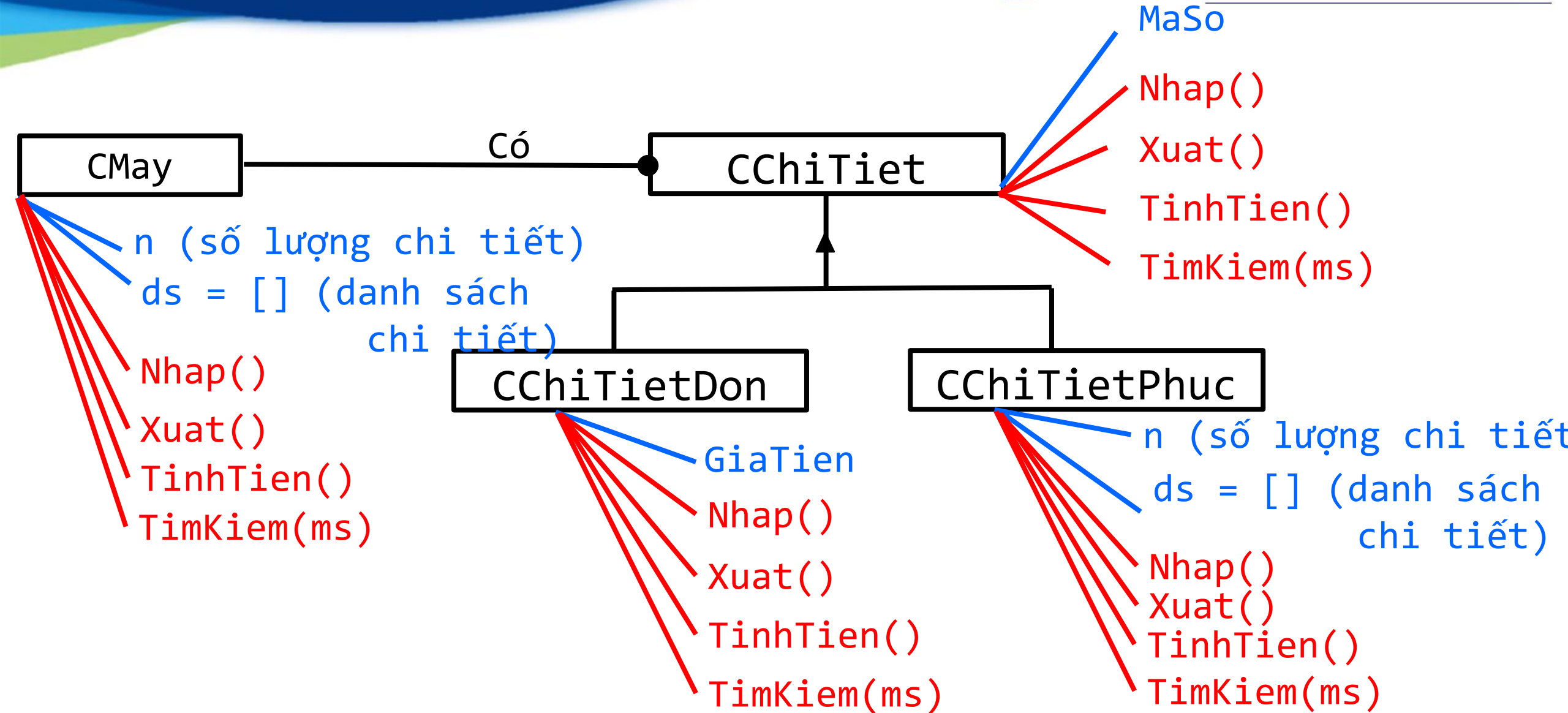
Nhap()
Xuat()
TinhTien()
TimKiem(ms)

Định nghĩa phương thức khởi tạo

```
1. class CMay:  
2.     def __init__(self):  
3.         self._n = 0  
4.         self._ds = []
```



ĐỊNH NGHĨA PHƯƠNG THỨC



Định nghĩa các phương thức

```
1. class CChiTiet:
2.     def Nhap(self):
3.         self._MaSo = int(input("Nhap ma so:"))
4.
5.     def Xuat(self):
6.         print("Ma so:", self._MaSo)
7.     ...
```

CChiTiet

MaSo

Nhap()

Xuat()

TinhTien()

TimKiem(ms)

Định nghĩa các phương thức

```
1. class CChiTiet:
2.     def TinhTien(self):
3.         return 0
4.     def TimKiem(self, ms):
5.         if(self._MaSo == ms):
6.             return self
7.         else:
8.             return None
```

CChiTiet

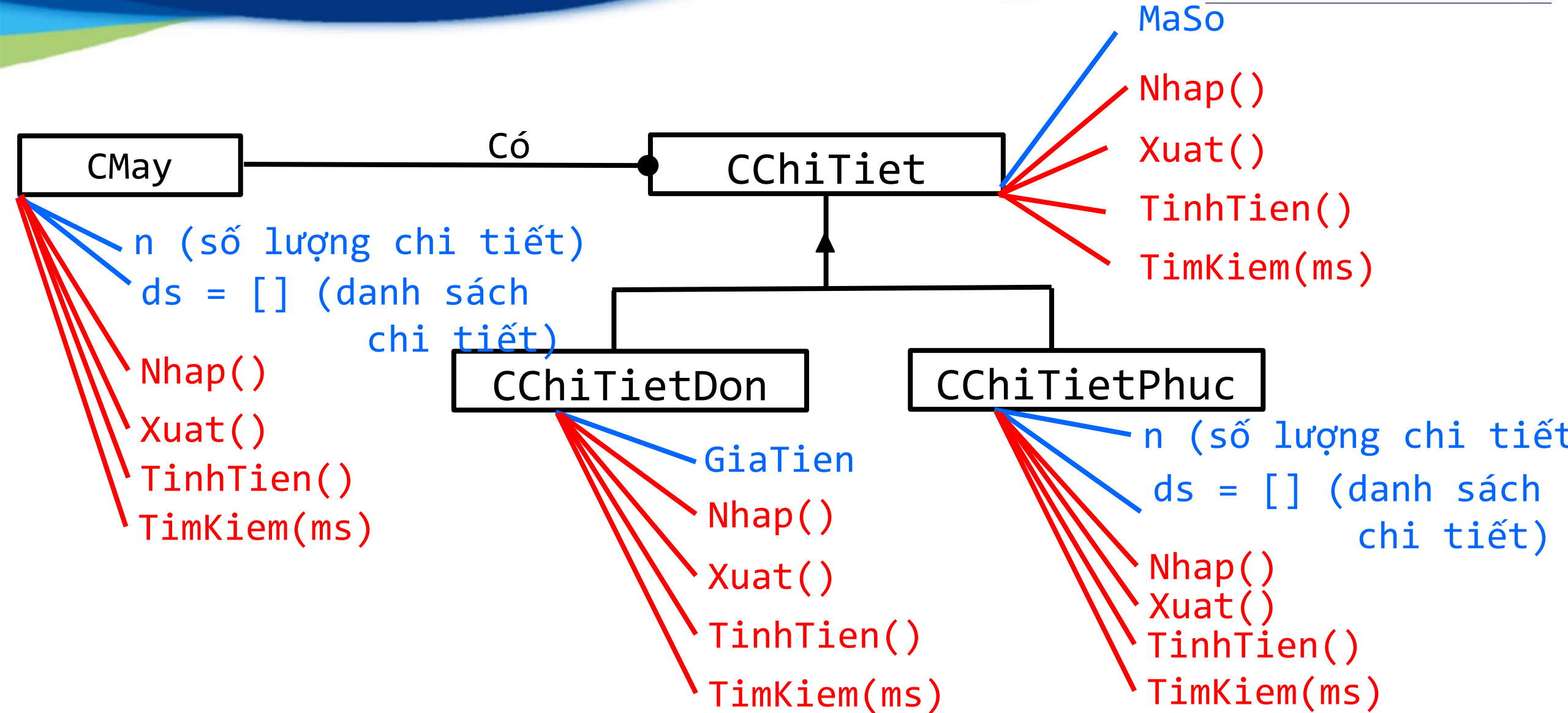
MaSo

Nhap()

Xuat()

TinhTien()

TimKiem(ms)



Định nghĩa các phương thức

```
1. class CChiTietDon(CChiTiet):  
2.     def Nhap(self):  
3.         super().Nhap()  
4.         self._GiaTien = float(input("Nhap gia: "))  
5.     ...
```

CChiTietDon

GiaTien

Nhap()

Xuat()

TinhTien()

TimKiem(ms)

Định nghĩa các phương thức

```
1. class CChiTietDon(CChiTiet):  
2.     ...  
3.     def Xuat(self):  
4.         super().Xuat()  
5.         print("Gia tien:", self.GiaTien)  
6.     ...
```

CChiTietDon

GiaTien

Nhap()

Xuat()

TinhTien()

TimKiem(ms)

Định nghĩa các phương thức

```
1. class CChiTietDon(CChiTiet):  
2.     ...  
3.     def TinhTien(self):  
4.         return self._GiaTien
```

```
5.     def TimKiem(self, ms):  
6.         return super().TimKiem(ms)
```

CChiTietDon

GiaTien

Nhap()

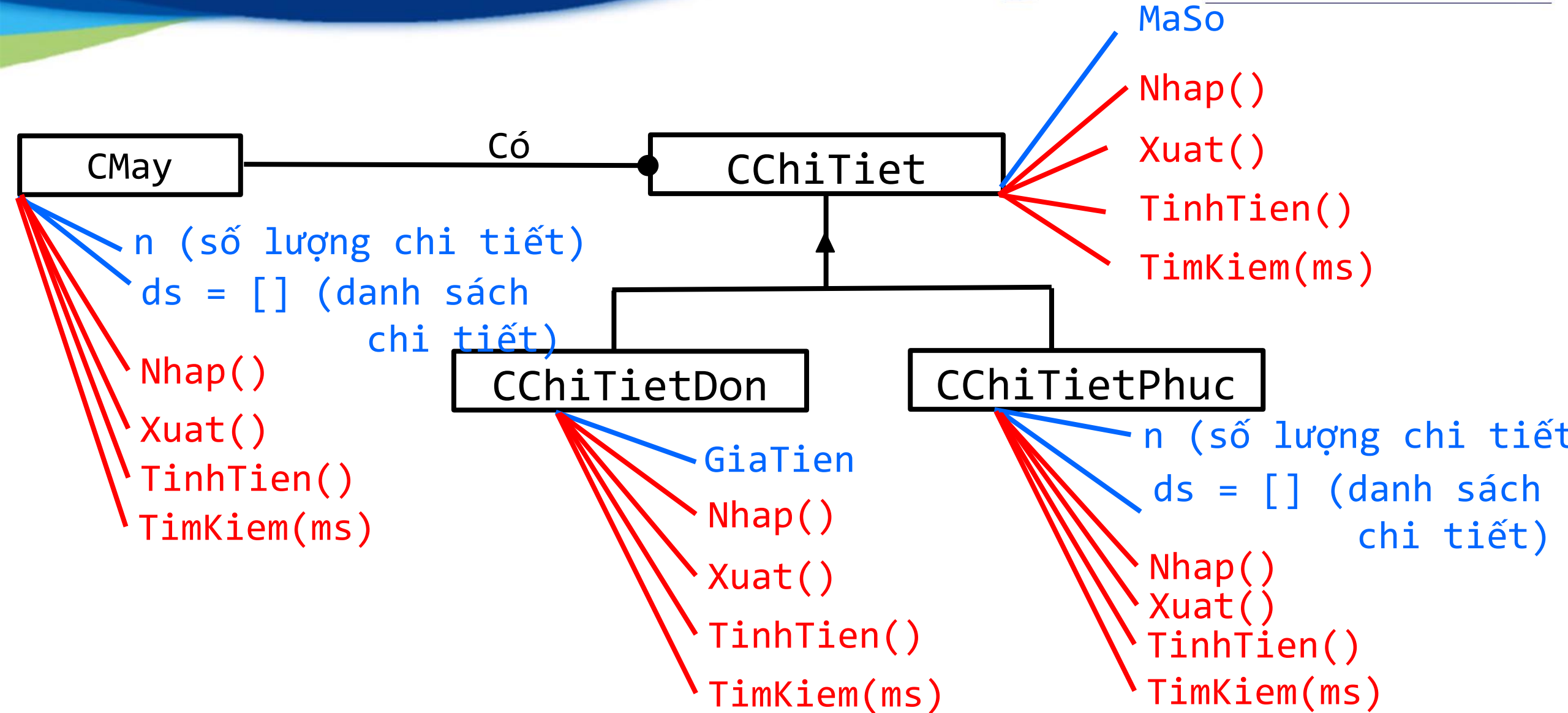
Xuat()

TinhTien()

TimKiem(ms)

Định nghĩa các phương thức

- Bài toán: Một cái máy có nhiều chi tiết. Mỗi chi tiết máy có thể là chi tiết đơn hoặc là chi tiết phức.
 - + Chi tiết đơn: là chi tiết không chứa bên trong nó chi tiết khác. Thông tin của chi tiết đơn bao gồm: mã số chi tiết, giá tiền.
 - + Chi tiết phức: là chi tiết chứa bên trong nó nhiều chi tiết thành phần, mỗi một chi tiết thành phần này có thể là chi tiết đơn hoặc là chi tiết phức. Thông tin của chi tiết phức bao gồm: mã số chi tiết, số lượng chi tiết thành phần, danh sách các chi tiết thành phần. Giá tiền của chi tiết phức bằng tổng giá tiền của các chi tiết thành phần.



Định nghĩa các phương thức

```
1. class CChiTietPhuc(CChiTiet):
2.     ...
3.     def TinhTien(self):
4.         s = 0
5.         for x in self._ds:
6.             s += x.TinhTien()
7.         return s
8.     ...
```

CChiTietPhuc

n (số lượng chi tiết)

ds = []

Nhap()

Xuat()

TinhTien()

TimKiem(ms)

Định nghĩa các phương thức

```
1. class CChiTietPhuc(CChiTiet):
```

```
2.     ...
```

```
3.     def TimKiem(self, ms):
```

```
4.         if self._MaSo == ms:
```

```
5.             return self
```

```
6.         for x in self._ds:
```

```
7.             if(x.TimKiem(ms) != None):
```

```
8.                 return x.TimKiem(ms)
```

```
9.         return None
```

CChiTietPhuc

n (số lượng chi tiết)

ds = []

Nhap()

Xuat()

TinhTien()

TimKiem(ms)

Định nghĩa các phương thức

```
1. class CChiTietPhuc(CChiTiet):
```

```
2.     ...
```

```
3.     def Xuat(self):
```

```
4.         super().Xuat()
```

```
5.         print("So luong chi tiet
```

```
6.             thanh phan:", self._n)
```

```
7.         for x in self._ds:
```

```
8.             x.Xuat()
```

```
9.     ...
```

CChiTietPhuc

n (số lượng chi tiết)

ds = []

Nhap()

Xuat()

TinhTien()

TimKiem(ms)

Định nghĩa các phương thức

```
1. class CChiTietPhuc(CChiTiet):  
2.     ...  
3.     def Nhap(self):  
4.         super().Nhap()  
5.         self._n = int(input("Nhap so luong chi tiet  
6.         CChiTietPhuc thanh phan:"))  
7.     ...
```

n (số lượng chi tiết)

ds = []

Nhap()

Xuat()

TinhTien()

Định nghĩa các phương thức

```
1. class CChiTietPhuc(CChiTiet):
2.     def Nhap(self):
3.         ...
4.         for i in range (self._n):
5.             type = int(input("Nhap loai (0. Don, 1. Phuc): "))
6.             if type == 0:
7.                 temp = CChiTietDon()
8.             else:
9.                 temp = CChiTietPhuc()
10.            temp.Nhap()
11.            self._ds.append(temp)
```

CChiTietPhuc

n (số lượng chi tiết)

ds = []

Nhap()

Xuat()

TinhTien()

TimKiem(ms)

Nhắc lại lý thuyết đa xạ

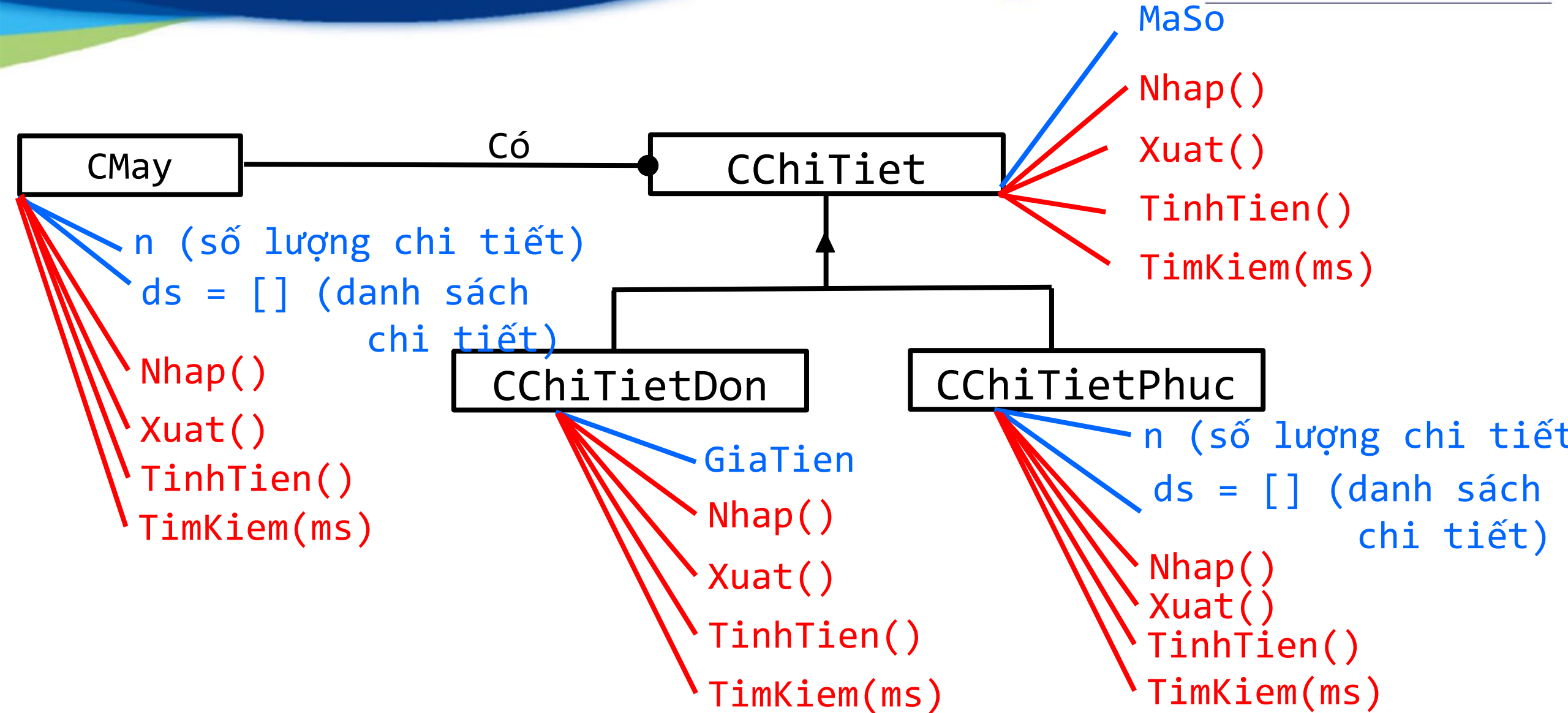
- Khái niệm: Đa xạ là cơ chế **tầm vực động** (dynamic scope) cho phép "**xác định**" đúng hành vi (phương thức – **method**) của đối tượng (**object**) khi yêu cầu thực hiện.
- Việc "**xác định**" được thực hiện theo nguyên tắc tự nhiên: con trỏ đối tượng đang giữ địa chỉ của đối tượng thuộc về lớp nào thì nó sẽ gọi thực hiện phương thức của lớp đối tượng (**class**) đó.
- Tầm vực động (dynamic scope) là cơ chế gọi thực hiện phương thức thông qua con trỏ đối tượng.

Nhắc lại lý thuyết đa xạ

- Một phương thức được khai báo bắt đầu với từ khóa **virtual** thì được gọi là phương thức ảo và phương thức này được gọi thực hiện theo cơ chế đa xạ nếu **lời gọi thực hiện phương thức** được thông qua một con trỏ đối tượng.
- Các phương thức ở lớp dẫn xuất cùng tên và cùng danh sách tham số đầu vào thì cũng sẽ là phương thức ảo nếu ở lớp cơ sở phương thức cùng tên và cùng tham số là phương thức ảo.

Nhắc lại lý thuyết đa xạ

- Khái niệm: Đa xạ là cơ chế **tầm vực động** (**dynamic scope**) cho phép "**xác định**" đúng hành vi (phương thức – **method**) của đối tượng (**object**) khi yêu cầu thực hiện.
- Việc "**xác định**" được thực hiện theo nguyên tắc tự nhiên: con trỏ đối tượng đang giữ địa chỉ của đối tượng thuộc về lớp nào thì nó sẽ gọi thực hiện phương thức của lớp đối tượng (**class**) đó.
- Tầm vực động (**dynamic scope**) là cơ chế gọi thực hiện phương thức thông qua con trỏ đối tượng.



Định nghĩa các phương thức

```
1. class CMay:
2.     ...
3.     def TinhTien(self):
4.         s = 0
5.         for x in self._ds:
6.             s += x.TinhTien()
7.         return s
8.     ...
```

CMay

n (số lượng chi tiết)

ds = []

Nhap()

Xuat()

TinhTien()

TimKiem(ms)

Định nghĩa các phương thức

```
1. class CMay:
2.     ...
3.     def TimKiem(self, ms):
4.         for x in self._ds:
5.             if(x.TimKiem(ms) != None):
6.                 return x.TimKiem(ms)
7.         return None
8.     ...
```

CMay

n
ds = []
Nhap()
Xuat()
TinhTien()
TimKiem(ms)

Định nghĩa các phương thức

```
1. class CChiTietPhuc:
2.     ...
3.     def Xuat(self):
4.         print("So chi tiet thanh phan:", self._n)
5.         for x in self._ds:
6.             x.Xuat()
7.         ...
```

CMay

n (số lượng chi tiết)

ds = []

Nhap()

Xuat()

TinhTien()

TimKiem(ms)

Định nghĩa các phương thức

```
1. class CMay:
2.     ...
3.     def Nhap(self):
4.         self._n = int(input("Nhập số lượng chi tiết
5.                             thành phần:"))
6.     ...
```

CMay

n (số lượng chi tiết)

ds = []

Nhap()

Xuat()

TinhTien()

TimKiem(ms)

Định nghĩa các phương thức

```
1. class CMay:
2.     def Nhap(self):
3.         ...
4.         for i in range (self._n):
5.             type = int(input("Nhap loai (0. Don, 1. Phuc): "))
6.             if type == 0:
7.                 temp = CChiTietDon()
8.             else:
9.                 temp = CChiTietPhuc()
10.            temp.Nhap()
11.            self._ds.append(temp)
```

CMay

n (số lượng chi tiết)

ds = []

Nhap()

Xuat()

TinhTien()

TimKiem(ms)

Cảm ơn quý vị đã lắng nghe

Nhóm tác giả

TS. Nguyễn Tấn Trần Minh Khang