

# BÀI TOÁN MẠCH ĐIỆN

- 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

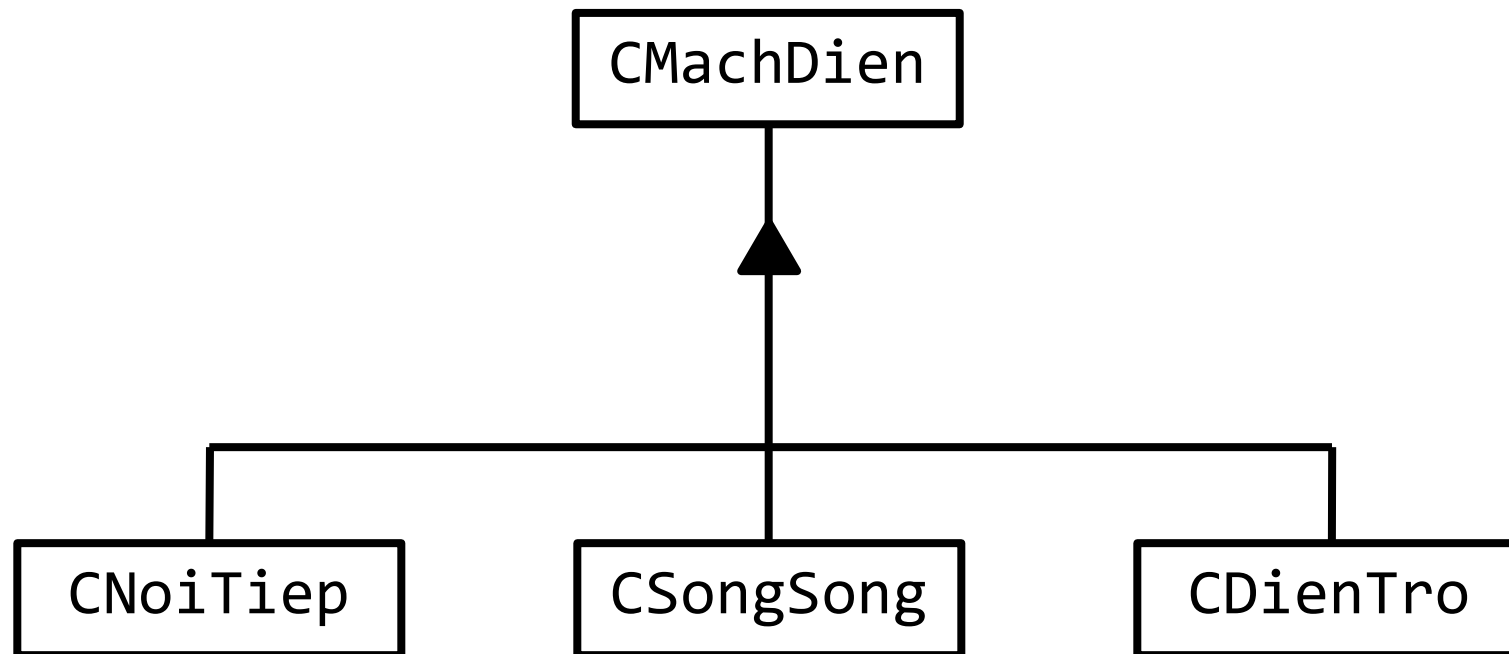
# Bài toán mạch điện

- Một mạch điện có thể là mạch song song hay mạch nối tiếp.
  - + Mạch nối tiếp là mạch điện trong đó có nhiều mạch điện nối tiếp nhau.
  - + Mạch song song là mạch điện trong đó có nhiều mạch điện song song nhau.
  - + Dạng đơn giản nhất của mạch điện là Điện trở. Thông tin của điện trở bao gồm: Mã số thiết bị (string), độ đo điện trở.

# Bài toán mạch điện

- **Yêu cầu:** Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:
  - + Nhập thông tin của một mạch điện.
  - + Tính điện trở tương đương.
  - + Xuất thông tin của mạch điện.

# THIẾT KẾ LỚP SƠ BỘ



# THIẾT KẾ LỚP CHI TIẾT

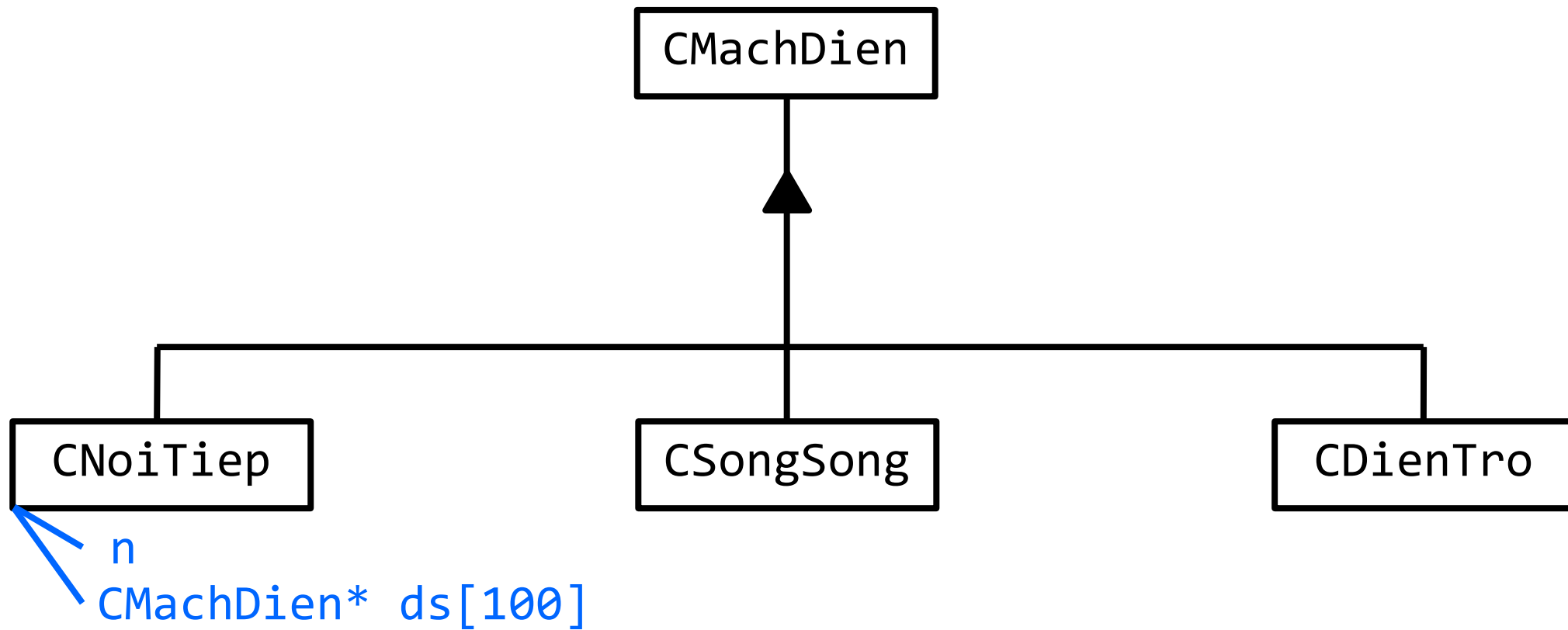
# Bài toán mạch điện

- Một mạch điện có thể là mạch song song hay mạch nối tiếp.
  - + Mạch nối tiếp là mạch điện trong đó có nhiều mạch điện nối tiếp nhau.
  - + Mạch song song là mạch điện trong đó có nhiều mạch điện song song nhau.
  - + Dạng đơn giản nhất của mạch điện là Điện trở. Thông tin của điện trở bao gồm: Mã số thiết bị (string), độ đo điện trở.



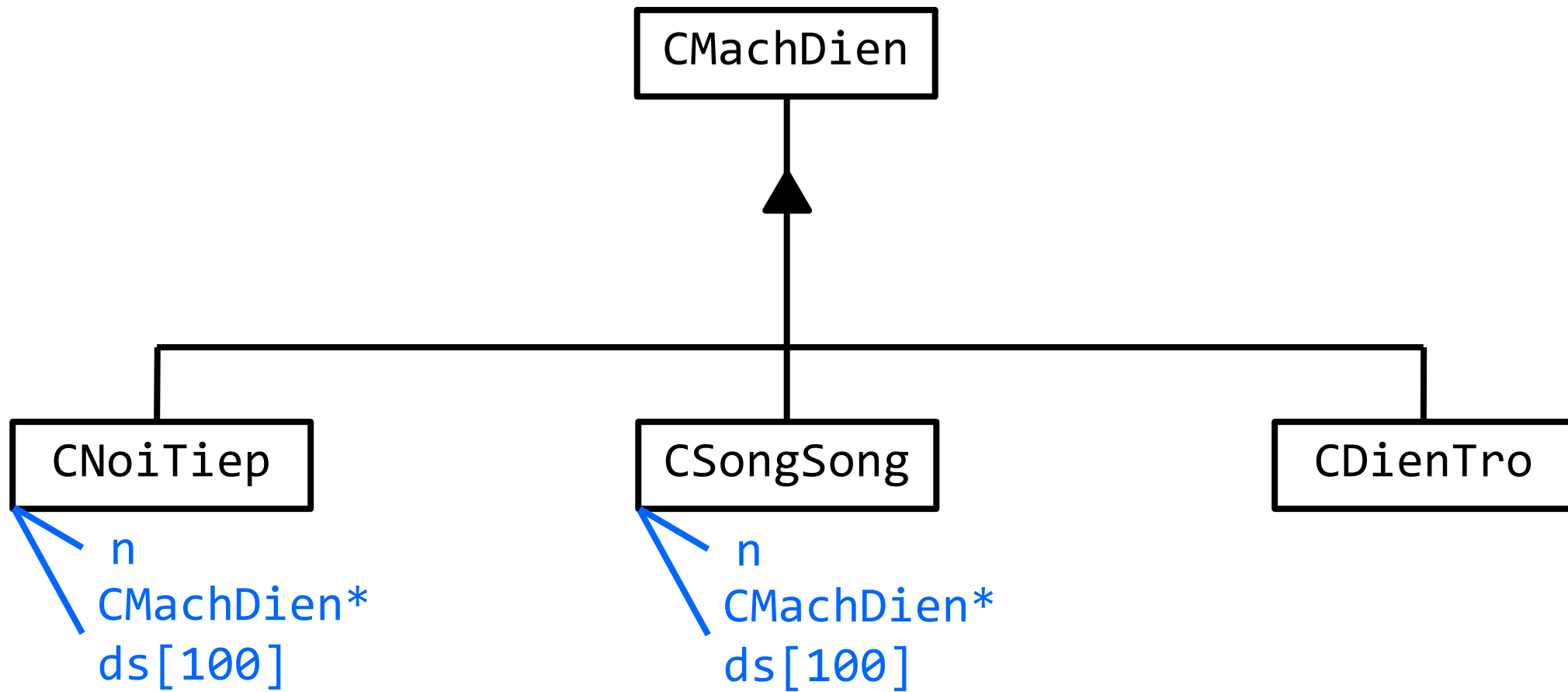
# Bài toán mạch điện

- Một mạch điện có thể là mạch song song hay mạch nối tiếp.
  - + Mạch nối tiếp là mạch điện trong đó có nhiều mạch điện nối tiếp nhau.
  - + Mạch song song là mạch điện trong đó có nhiều mạch điện song song nhau.
  - + Dạng đơn giản nhất của mạch điện là Điện trở. Thông tin của điện trở bao gồm: Mã số thiết bị (string), độ đo điện trở.



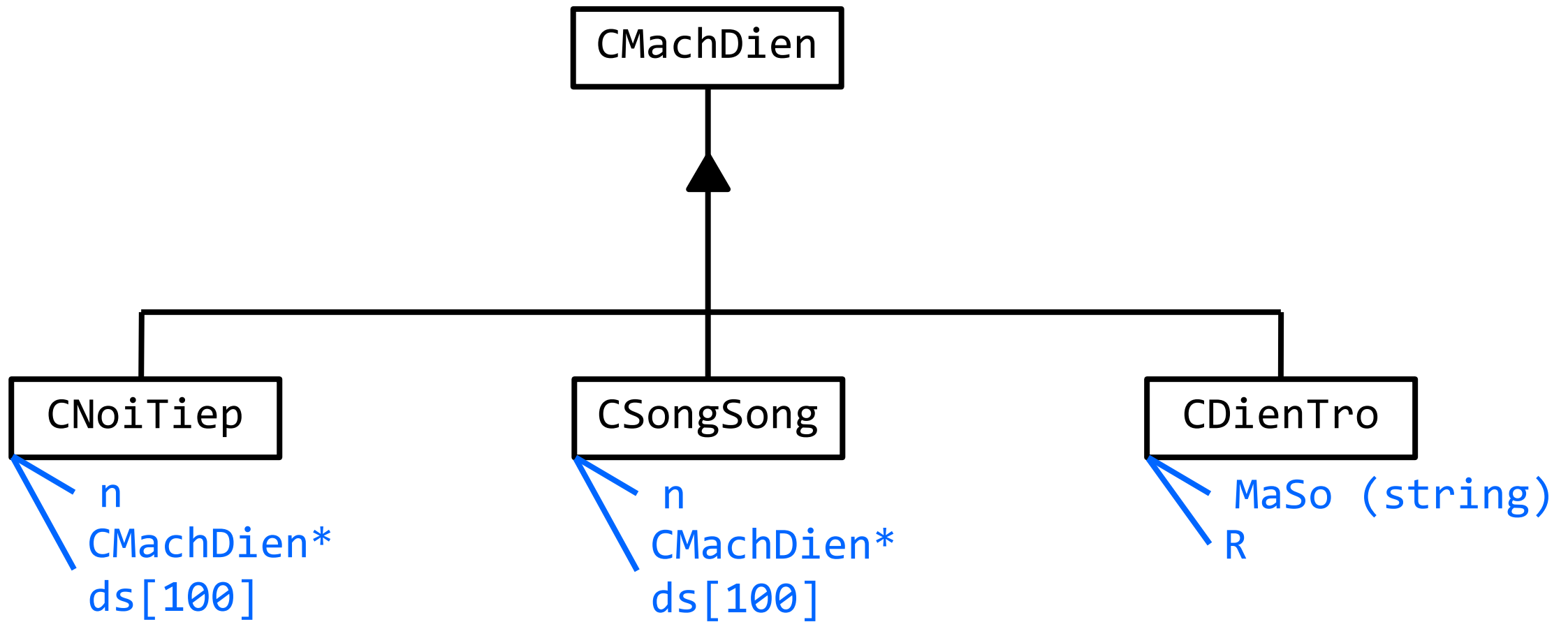
# Bài toán mạch điện

- Một mạch điện có thể là mạch song song hay mạch nối tiếp.
  - + Mạch nối tiếp là mạch điện trong đó có nhiều mạch điện nối tiếp nhau.
  - + Mạch song song là mạch điện trong đó có nhiều mạch điện song song nhau.
  - + Dạng đơn giản nhất của mạch điện là Điện trở. Thông tin của điện trở bao gồm: Mã số thiết bị (string), độ đo điện trở.



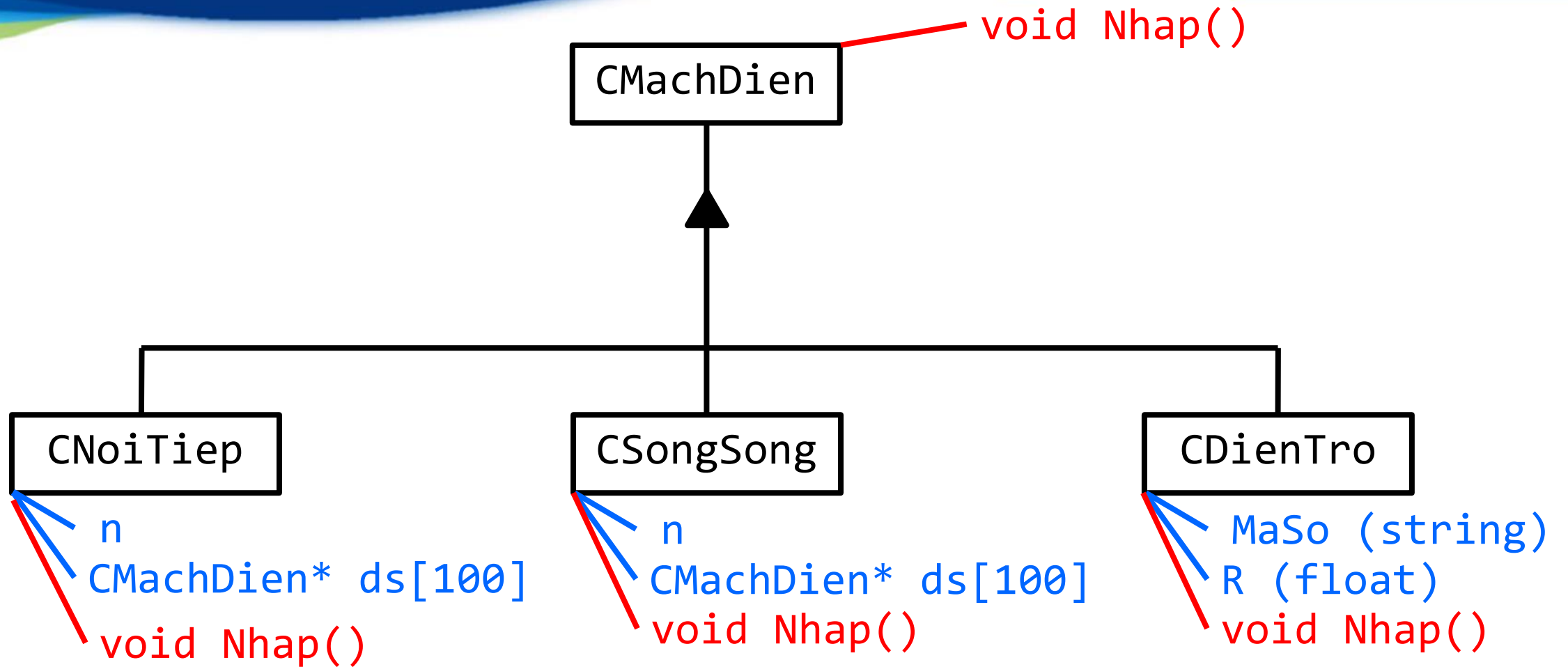
# Bài toán mạch điện

- Một mạch điện có thể là mạch song song hay mạch nối tiếp.
  - + Mạch nối tiếp là mạch điện trong đó có nhiều mạch điện nối tiếp nhau.
  - + Mạch song song là mạch điện trong đó có nhiều mạch điện song song nhau.
  - + Dạng đơn giản nhất của mạch điện là Điện trở. Thông tin của điện trở bao gồm: Mã số thiết bị (string), độ đo điện trở.



# Bài toán mạch điện

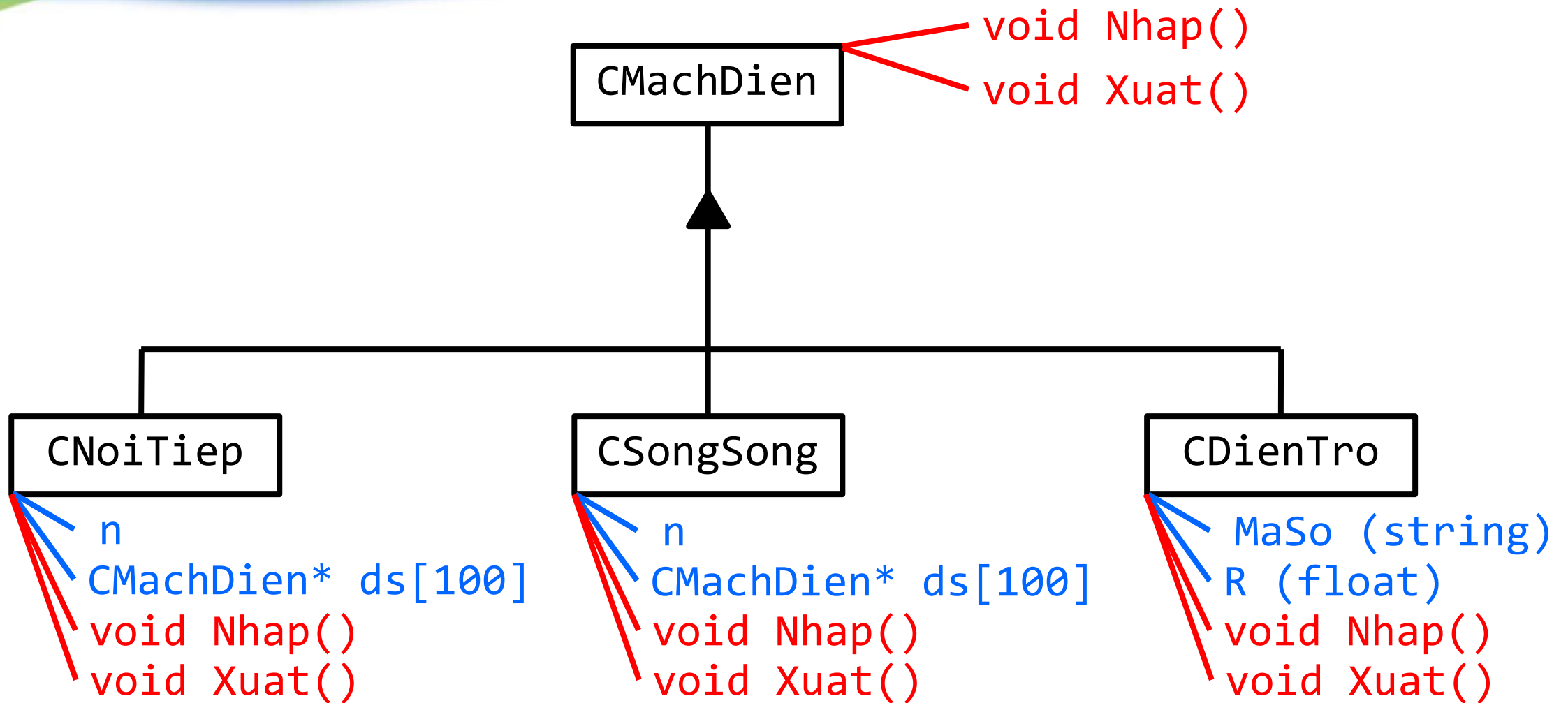
- **Yêu cầu:** Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:
  - + Nhập thông tin của một mạch điện.
  - + Tính điện trở tương đương.
  - + Xuất thông tin của mạch điện.





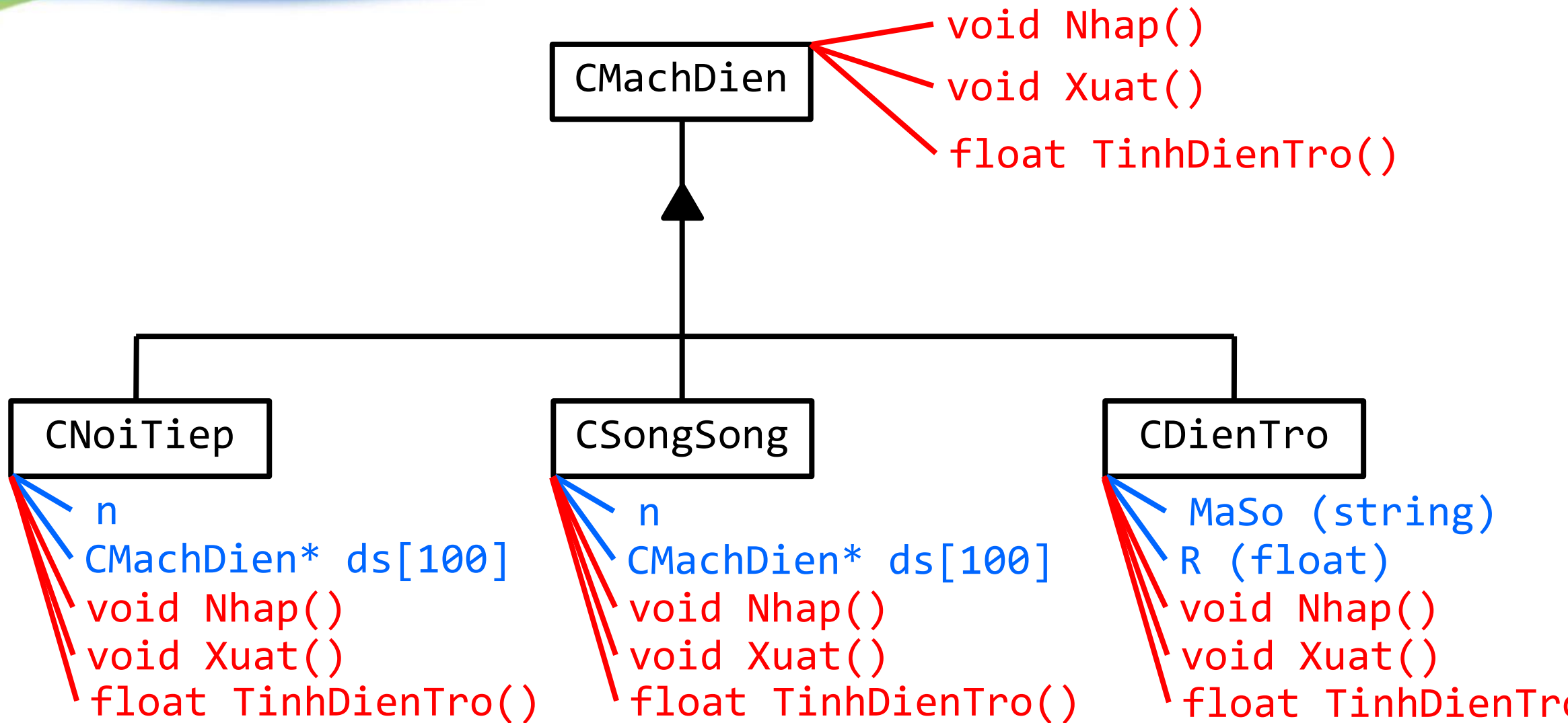
# Bài toán mạch điện

- **Yêu cầu:** Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:
  - + Nhập thông tin của một mạch điện.
  - + Tính điện trở tương đương.
  - + Xuất thông tin của mạch điện.

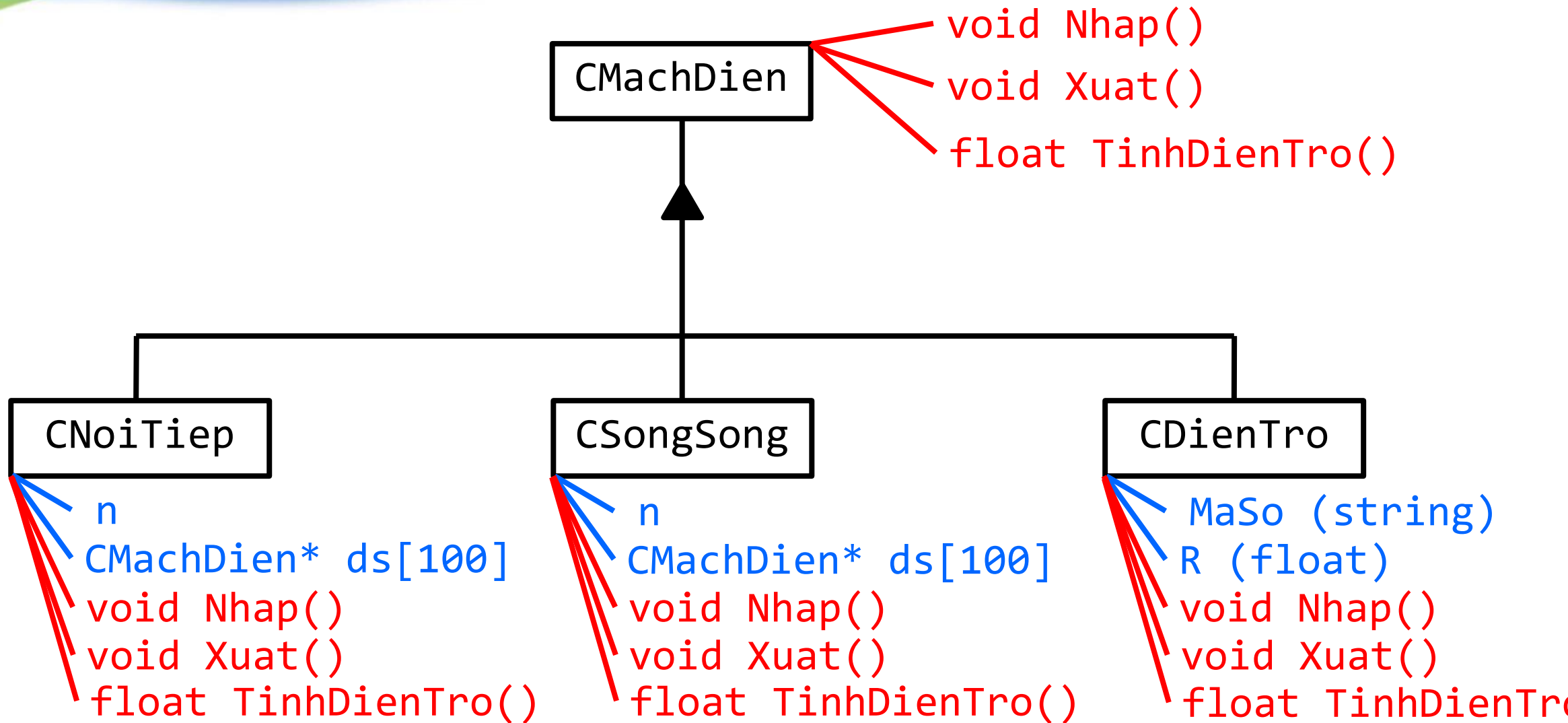


# Bài toán mạch điện

- **Yêu cầu:** Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:
  - + Nhập thông tin của một mạch điện.
  - + Tính điện trở tương đương.
  - + Xuất thông tin của mạch điện.



# KHAI BÁO LỚP



# Khai báo lớp

```
11.class CMachDien
12.{
13.    public:
14.        virtual void Nhap();
15.        virtual void Xuat();
16.        virtual float TinhDienTro();
17.};
```

CMachDien

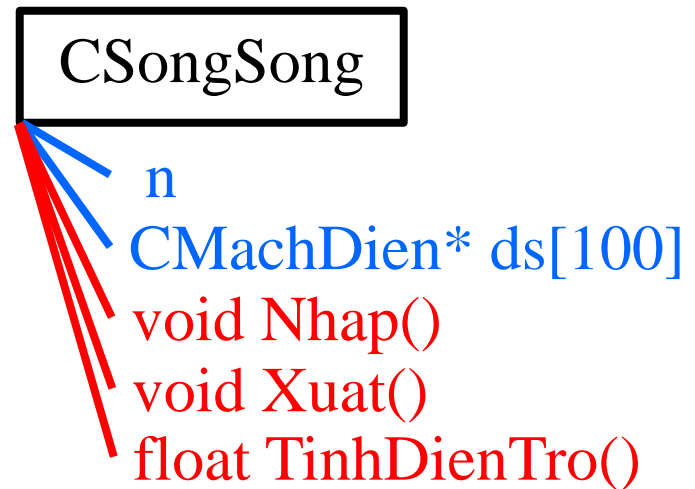
void Nhap()

void Xuat()

float TinhDienTro()

# Khai báo lớp

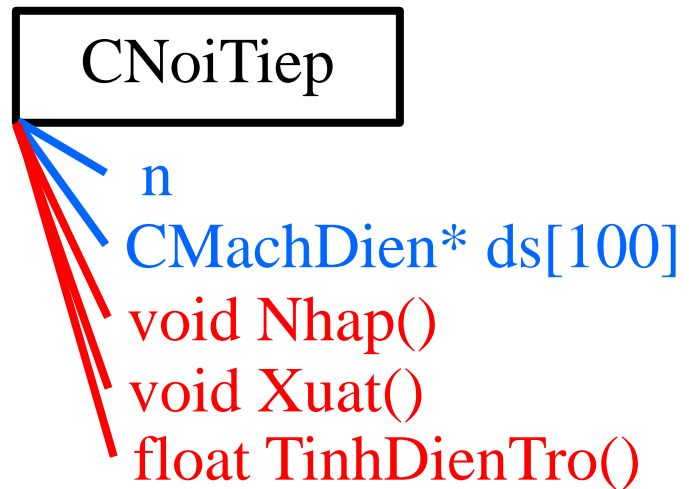
```
11.class CMachSongSong:public CMachDien
12.{
13.    protected:
14.        int n;
15.        CMachDien* ds[100];
16.    public:
17.        void Nhap();
18.        void Xuat();
19.        float TinhDienTro();
20.};
```





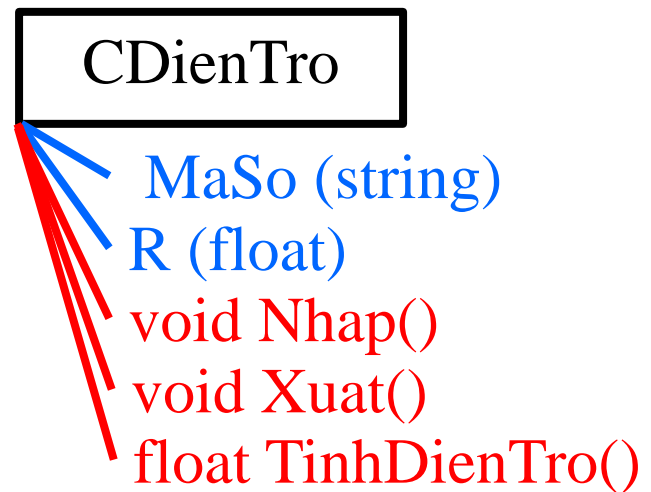
# Khai báo lớp

```
11.class CMachNoiTiep:public CMachDien
12.{
13.    protected:
14.        int n;
15.        CMachDien* ds[100];
16.    public:
17.        void Nhap();
18.        void Xuat();
19.        float TinhDienTro();
20.};
```



# Khai báo lớp


```
11.class CDienTro:public CMachDien
12.{
13.    protected:
14.        string MaSo
15.        float R;
16.    public:
17.        void Nhap();
18.        void Xuat();
19.        float TinhDienTro();
20.};
```



# ĐỊNH NGHĨA CÁC PHƯƠNG THỨC

# Định nghĩa phương thức

```
11.float CMachDien::TinhDienTro()  
12.{  
13.    return 0;  
14.}
```



Định nghĩa  
phương thức  
TinhDienTro

CMachDien

void Nhap()

void Xuat()

float TinhDienTro()

# Định nghĩa phương thức

```
11.float CDienTro::TinhDienTro()  
12.{  
13.|    return R;  
14.}
```

CDienTro

MaSo (string)

R (float)

void Nhap()

void Xuat()

float TinhDienTro()

# Định nghĩa phương thức

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

CNoiTiep

n

CMachDien\* ds[100]

void Nhap()

void Xuat()

float TinhDienTro()

# Định nghĩa phương thức

```
11.float CMachSongSong::TinhDienTro()  
12.{  
13.    float s=0;  
14.    for(int i=0;i<n;i++)  
15.        s = s + 1/ds[i]->TinhDienTro();  
16.    return 1/s;  
17.}
```

CSongSong

n

CMachDien\* ds[100]


void Nhap()

void Xuat()

float TinhDienTro()

# Định nghĩa phương thức

```
11. void CMachDien::Nhap()  
12. {  
13. |   return;  
14. }
```



Định nghĩa  
phương thức  
Nhap

CMachDien

void Nhap()

void Xuat()

float TinhDienTro()



# Định nghĩa phương thức

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

CDienTro

MaSo (string)

R (float)

void Nhap()

void Xuat()

float TinhDienTro()

```
11. void CMachNoiTiep::Nhap()  
12. {  
13.     cout<<"Nhap n:";  
14.     cin>>n;  
15.     for(int i=0;i<n;i++)  
16.     {  
17.         int loai;  
18.         cout<<"Nhap loai (0. Noi tiep, 1. Song song):";  
19.         cin>>loai;  
20.         switch(loai)  
21.         {  
22.             case 0: ds[i] = new CMachNoiTiep;  
23.                 break;  
24.             case 1: ds[i] = new CMachSongSong;  
25.                 break;  
26.         }  
27.         ds[i]->Nhap();  
28.     }  
29. }
```

CNoiTiep

n

CMachDien\* ds[100]

void Nhap()

void Xuat()

float TinhDienTro()

```
11. void CMachSongSong::Nhap()  
12. {  
13.     cout<<"Nhap n:";  
14.     cin>>n;  
15.     for(int i=0;i<n;i++)  
16.     {  
17.         int loai;  
18.         cout<<"Nhap loai (0. Noi tiep, 1. Song song):";  
19.         cin>>loai;  
20.         switch(loai)  
21.         {  
22.             case 0: ds[i] = new CMachNoiTiep;  
23.                 break;  
24.             case 1: ds[i] = new CMachSongSong;  
25.                 break;  
26.         }  
27.         ds[i]->Nhap();  
28.     }  
29. }
```

CSongSong

n

CMachDien\* ds[100]


void Nhap()

void Xuat()

float TinhDienTro()

# Định nghĩa phương thức

```
11. void CMachDien::Xuat()  
12. {  
13. |   return;  
14. }
```



Định nghĩa  
phương thức  
Xuat

CMachDien

void Nhap()

void Xuat()

float TinhDienTro()

# Định nghĩa phương thức

```
11. void CDienTro::Xuat()  
12. {  
13.     cout<<"Ma so:";  
14.     cout<<MaSo;  
15.     cout<<"Dien tro R:";  
16.     cout<<R;  
17. }
```

CDienTro

MaSo (string)

R (float)

void Nhap()

void Xuat()

float TinhDienTro()

```
11. void CMachNoiTiep::Xuat()  
12. {  
13.     cout<<"So luong mach dien thanh phan:";  
14.     cout<<n;  
15.     for(int i=0;i<n;i++)  
16.     {  
17.         ds[i]->Xuat();  
18.     }  
19. }
```

CNoiTiep

n

CMachDien\* ds[100]

void Nhap()

void Xuat()

float TinhDienTro()

```
11. void CMachSongSong::Xuat()  
12. {  
13.     cout<<"So luong mach dien thanh phan:";  
14.     cout<<n;  
15.     for(int i=0;i<n;i++)  
16.     {  
17.         ds[i]->Xuat();  
18.     }  
19. }
```

CSongSong

n

CMachDien\* ds[100]

void Nhap()

void Xuat()

float TinhDienTro()

# ĐỊNH NGHĨA HÀM MAIN



```
11. int main()
12. {
13.     CMachDien *a;
14.     int loai;
15.     cout<<"Nhap loai (0. NT, 1. SS, 2. R:";
16.     cin>>loai;
17.     switch(loai)
18.     {
19.         case 0: a = new CMachNoiTiep; break;
20.         case 1: a = new CMachSongSong; break;
21.         case 2: a = new CDienTro; break;
22.     }
23.     a->Nhap();
```

```
11. int main()  
12. {  
13.     ...  
14.     cout<<"Mach dien ban dau: ";  
15.     a->Xuat();  
  
16.     float kq = a->TinhDienTro();  
17.     cout<<"Tong dien tro mach dien la: ";  
18.     cout<<kq;  
19.     return 1;  
20. }
```

**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 MẠCH ĐIỆN PYTHON

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

# BÀI TOÁN

# Bài toán mạch điện

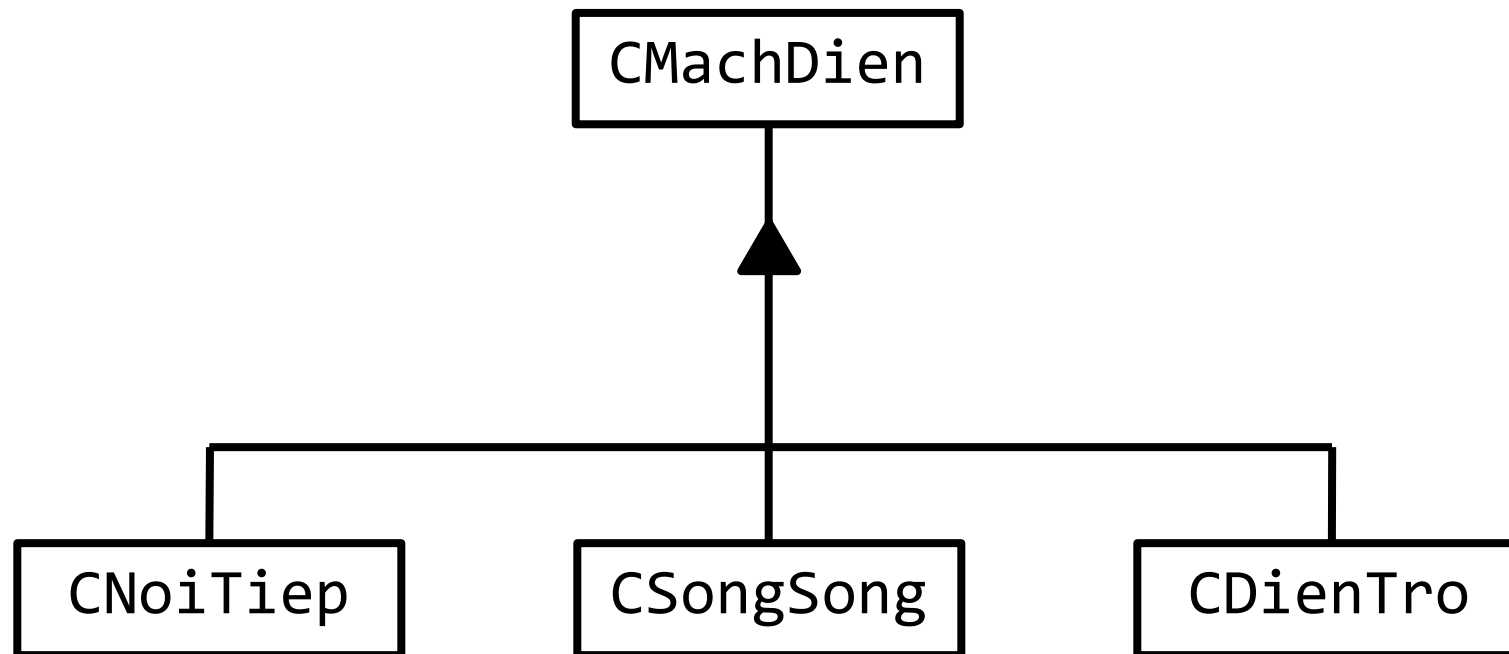
- Một mạch điện có thể là mạch song song hay mạch nối tiếp.
  - + Mạch nối tiếp là mạch điện trong đó có nhiều mạch điện nối tiếp nhau.
  - + Mạch song song là mạch điện trong đó có nhiều mạch điện song song nhau.
  - + Dạng đơn giản nhất của mạch điện là Điện trở. Thông tin của điện trở bao gồm: Mã số thiết bị (string), độ đo điện trở.

# Bài toán mạch điện

- **Yêu cầu:** Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:
  - + Nhập thông tin của một mạch điện.
  - + Tính điện trở tương đương.
  - + Xuất thông tin của mạch điện.

# THIẾT KẾ LỚP SƠ BỘ





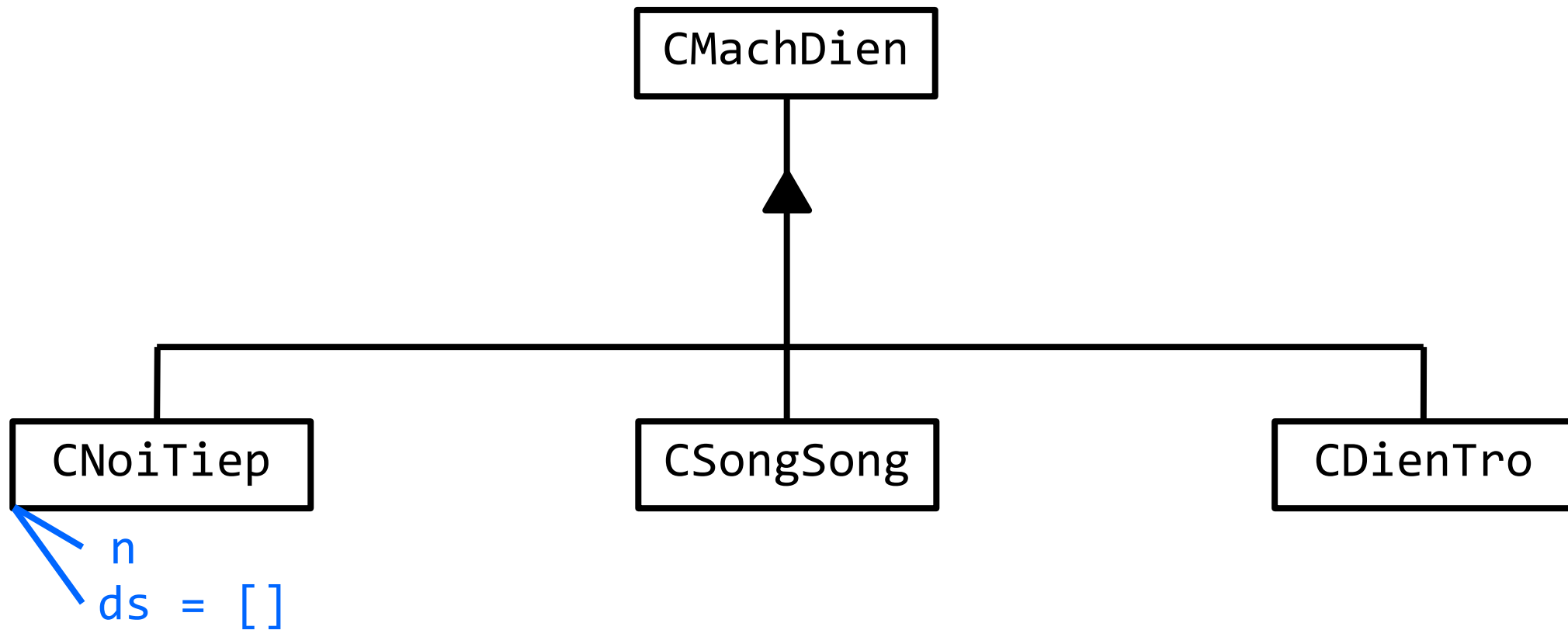
# THIẾT KẾ LỚP CHI TIẾT

# Bài toán mạch điện

- Một mạch điện có thể là mạch song song hay mạch nối tiếp.
  - + Mạch nối tiếp là mạch điện trong đó có nhiều mạch điện nối tiếp nhau.
  - + Mạch song song là mạch điện trong đó có nhiều mạch điện song song nhau.
  - + Dạng đơn giản nhất của mạch điện là Điện trở. Thông tin của điện trở bao gồm: Mã số thiết bị (string), độ đo điện trở.

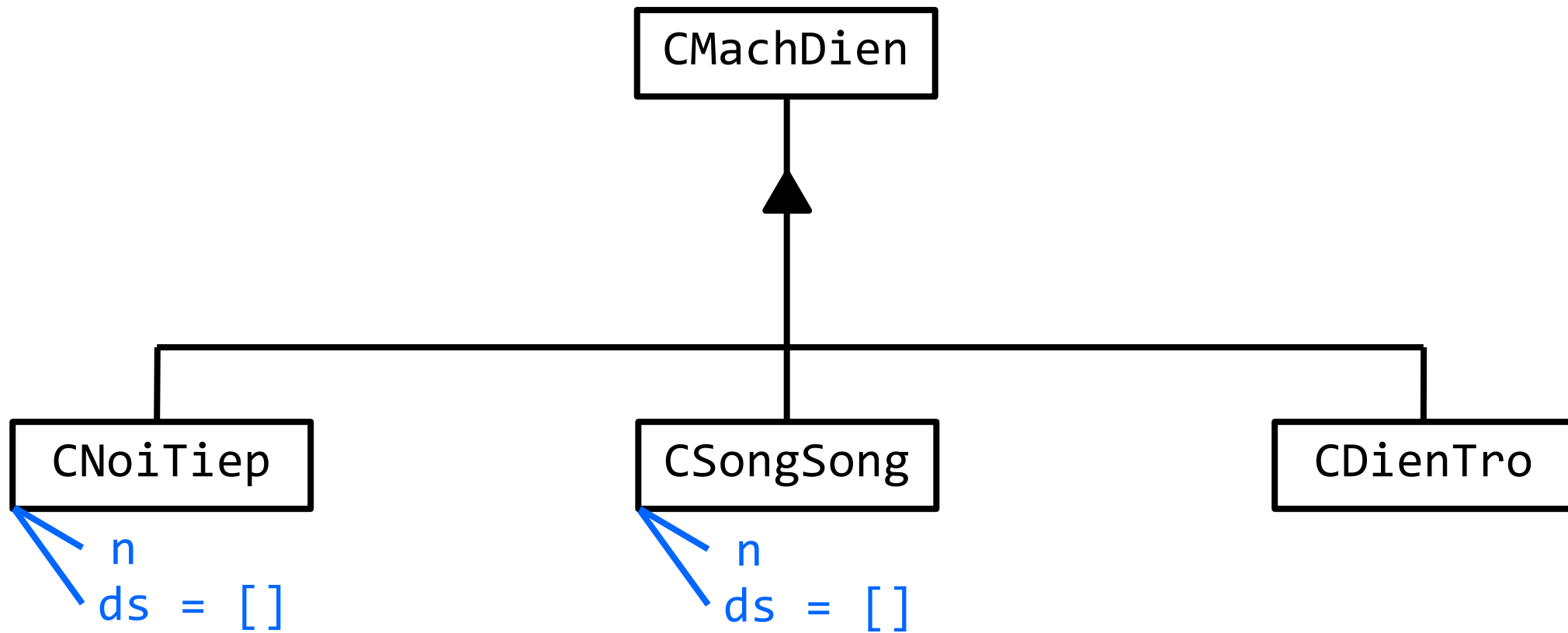
# Bài toán mạch điện

- Một mạch điện có thể là mạch song song hay mạch nối tiếp.
  - + Mạch nối tiếp là mạch điện trong đó có nhiều mạch điện nối tiếp nhau.
  - + Mạch song song là mạch điện trong đó có nhiều mạch điện song song nhau.
  - + Dạng đơn giản nhất của mạch điện là Điện trở. Thông tin của điện trở bao gồm: Mã số thiết bị (string), độ đo điện trở.



# Bài toán mạch điện

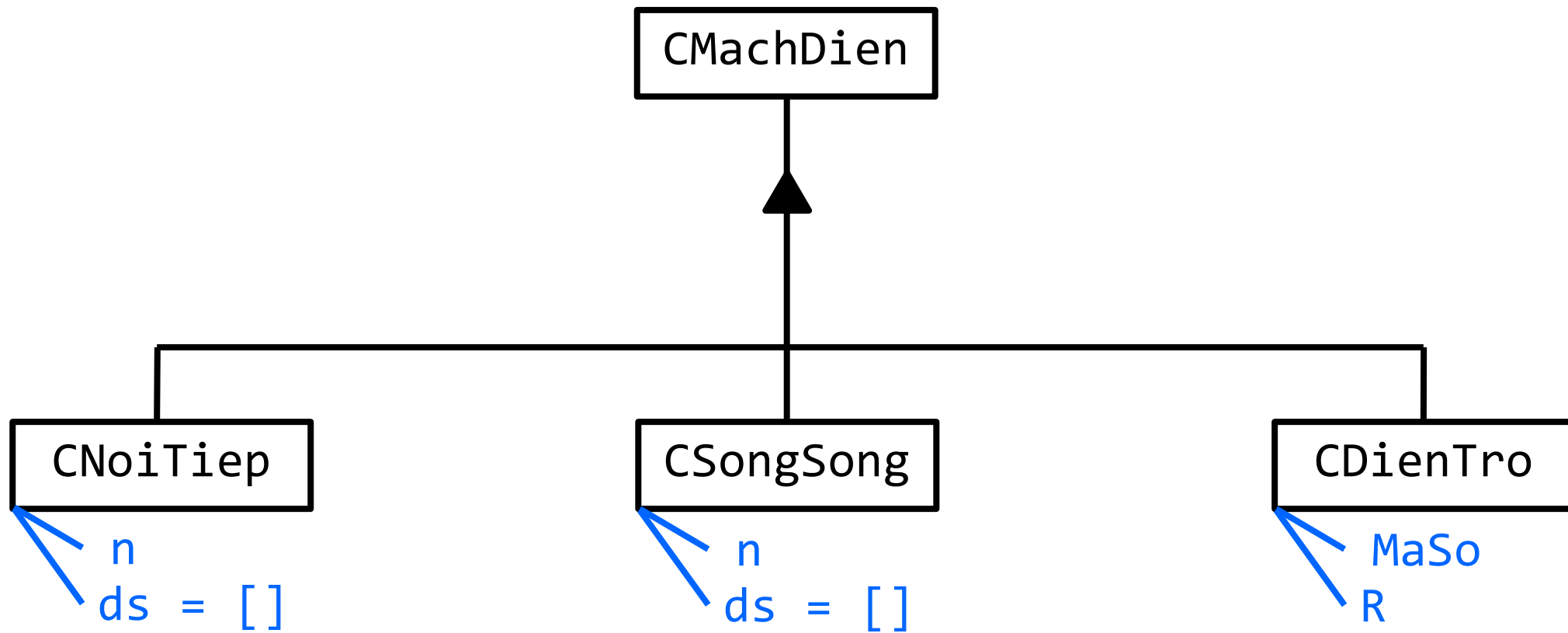
- Một mạch điện có thể là mạch song song hay mạch nối tiếp.
  - + Mạch nối tiếp là mạch điện trong đó có nhiều mạch điện nối tiếp nhau.
  - + Mạch song song là mạch điện trong đó có nhiều mạch điện song song nhau.
  - + Dạng đơn giản nhất của mạch điện là Điện trở. Thông tin của điện trở bao gồm: Mã số thiết bị (string), độ đo điện trở.



# Bài toán mạch điện

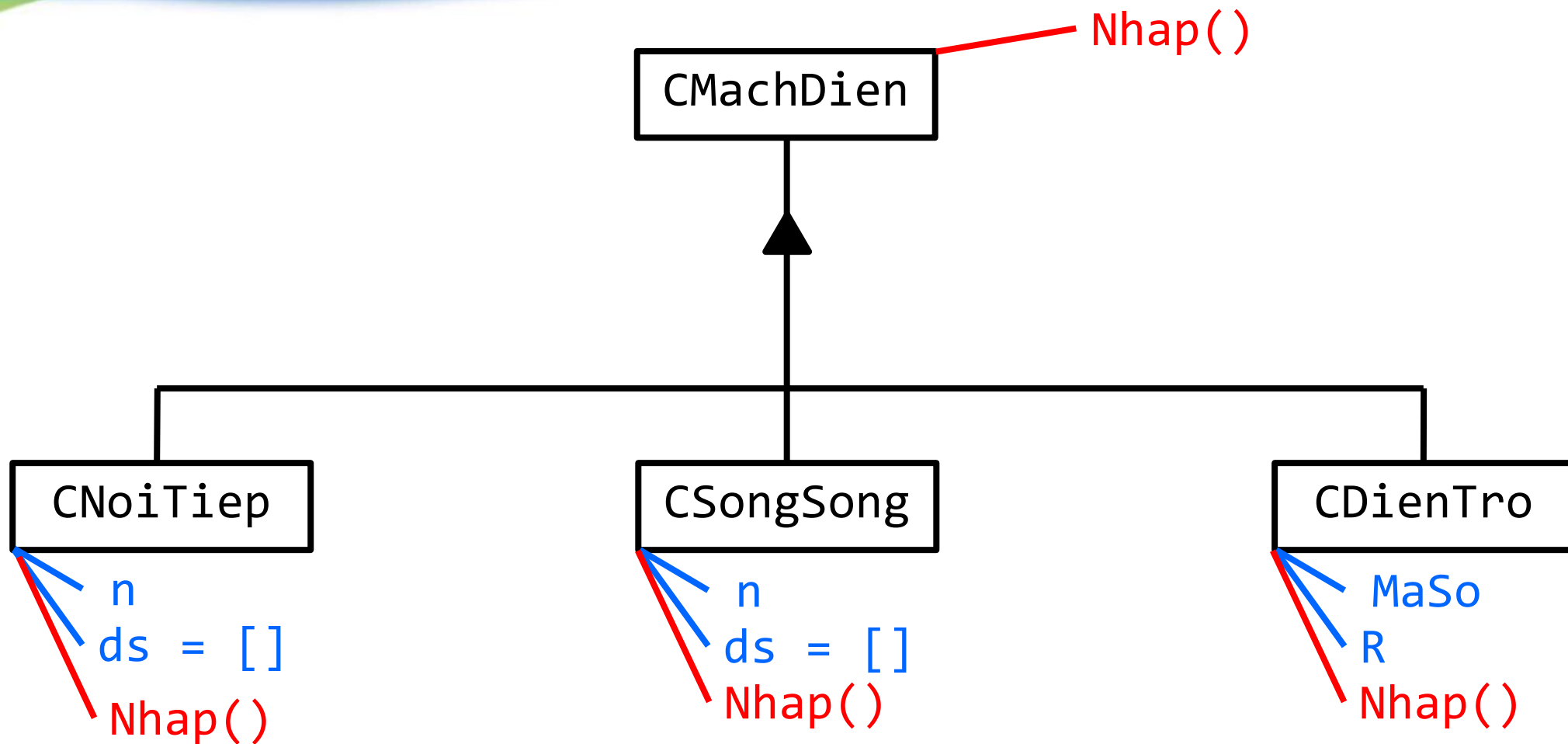
- Một mạch điện có thể là mạch song song hay mạch nối tiếp.
  - + Mạch nối tiếp là mạch điện trong đó có nhiều mạch điện nối tiếp nhau.
  - + Mạch song song là mạch điện trong đó có nhiều mạch điện song song nhau.
  - + Dạng đơn giản nhất của mạch điện là Điện trở. Thông tin của điện trở bao gồm: Mã số thiết bị (string), độ đo điện trở.





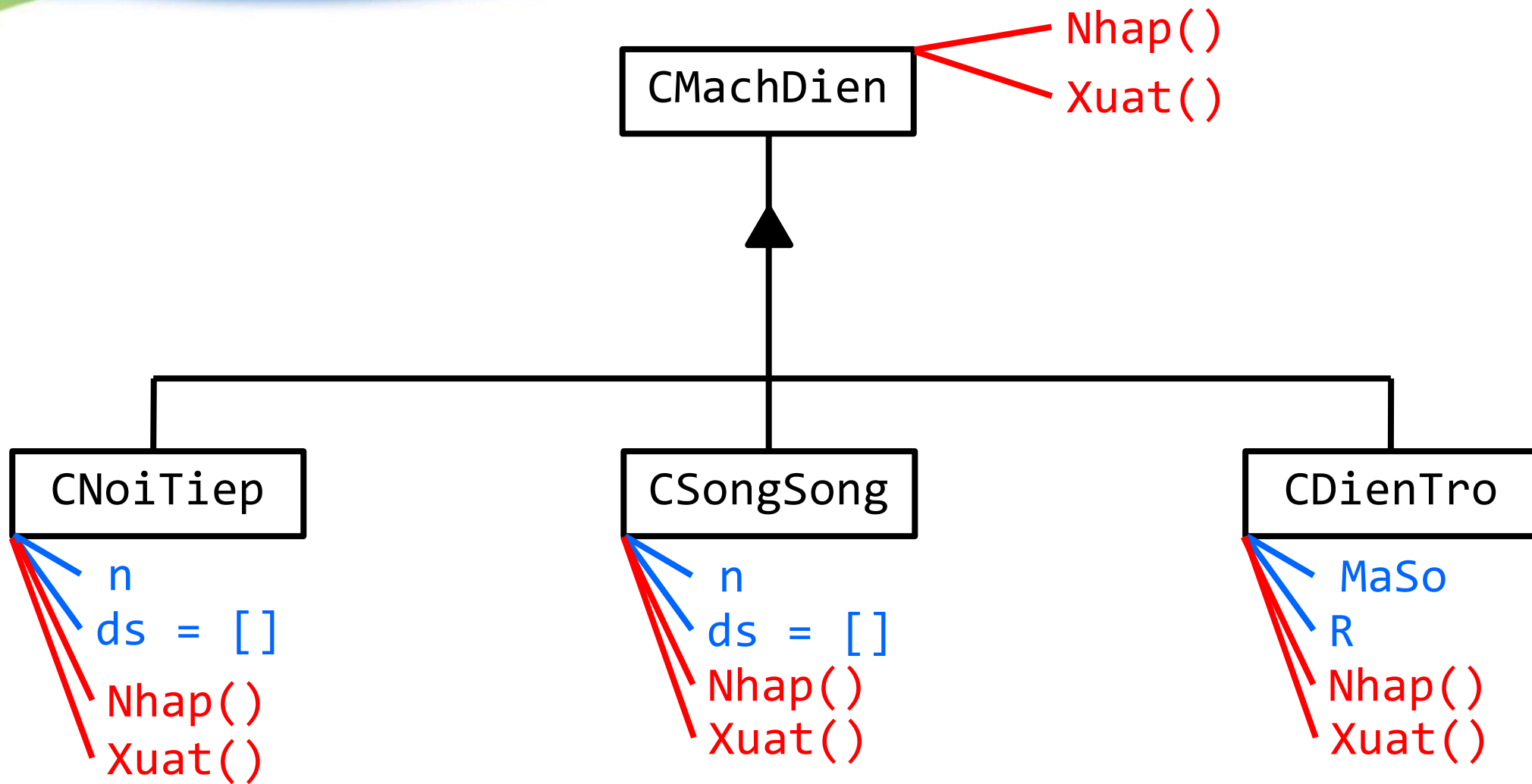
# Bài toán mạch điện

- **Yêu cầu:** Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:
  - + Nhập thông tin của một mạch điện.
  - + Tính điện trở tương đương.
  - + Xuất thông tin của mạch điện.



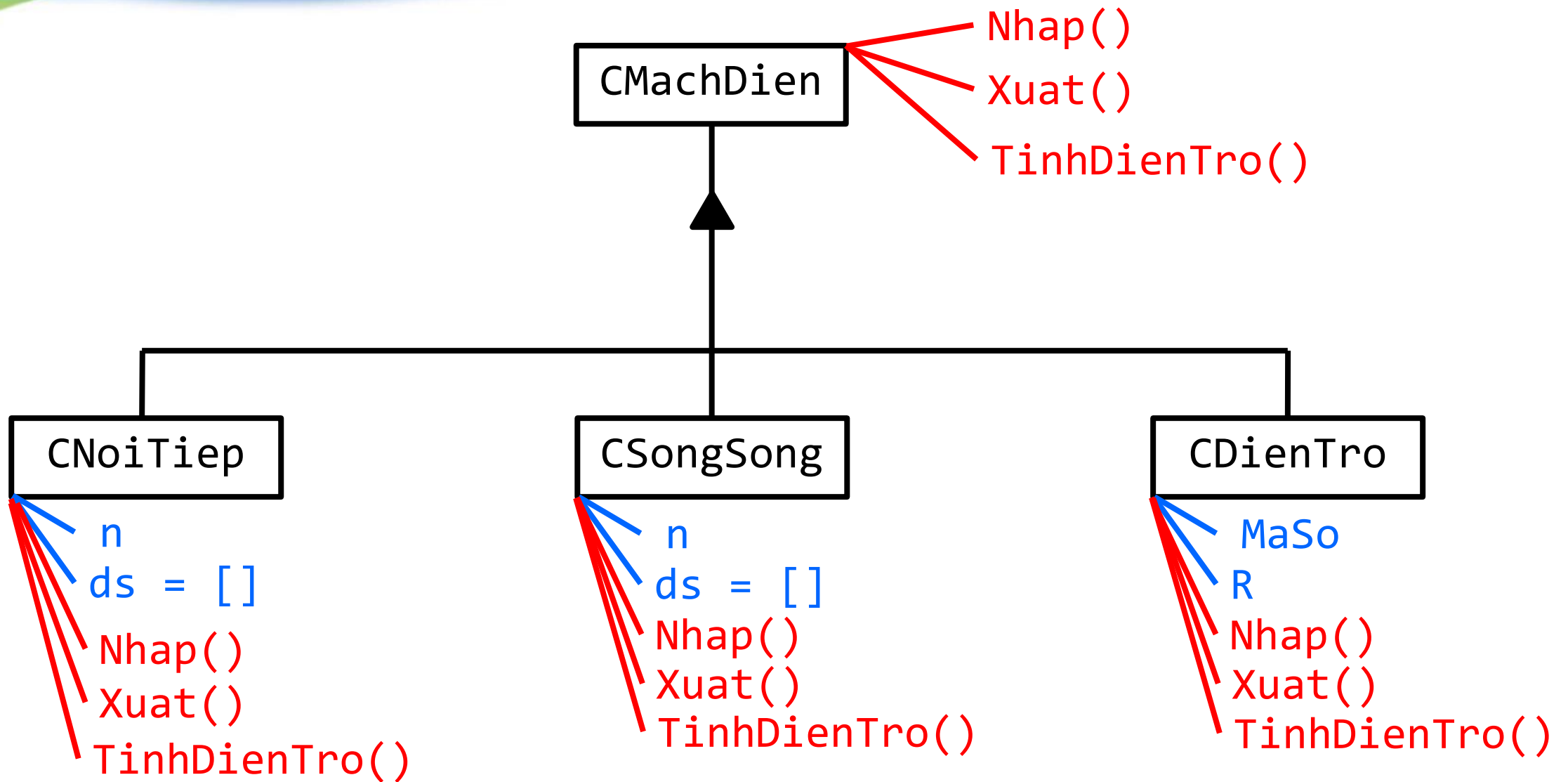
# Bài toán mạch điện

- **Yêu cầu:** Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:
  - + Nhập thông tin của một mạch điện.
  - + Tính điện trở tương đương.
  - + Xuất thông tin của mạch điện.



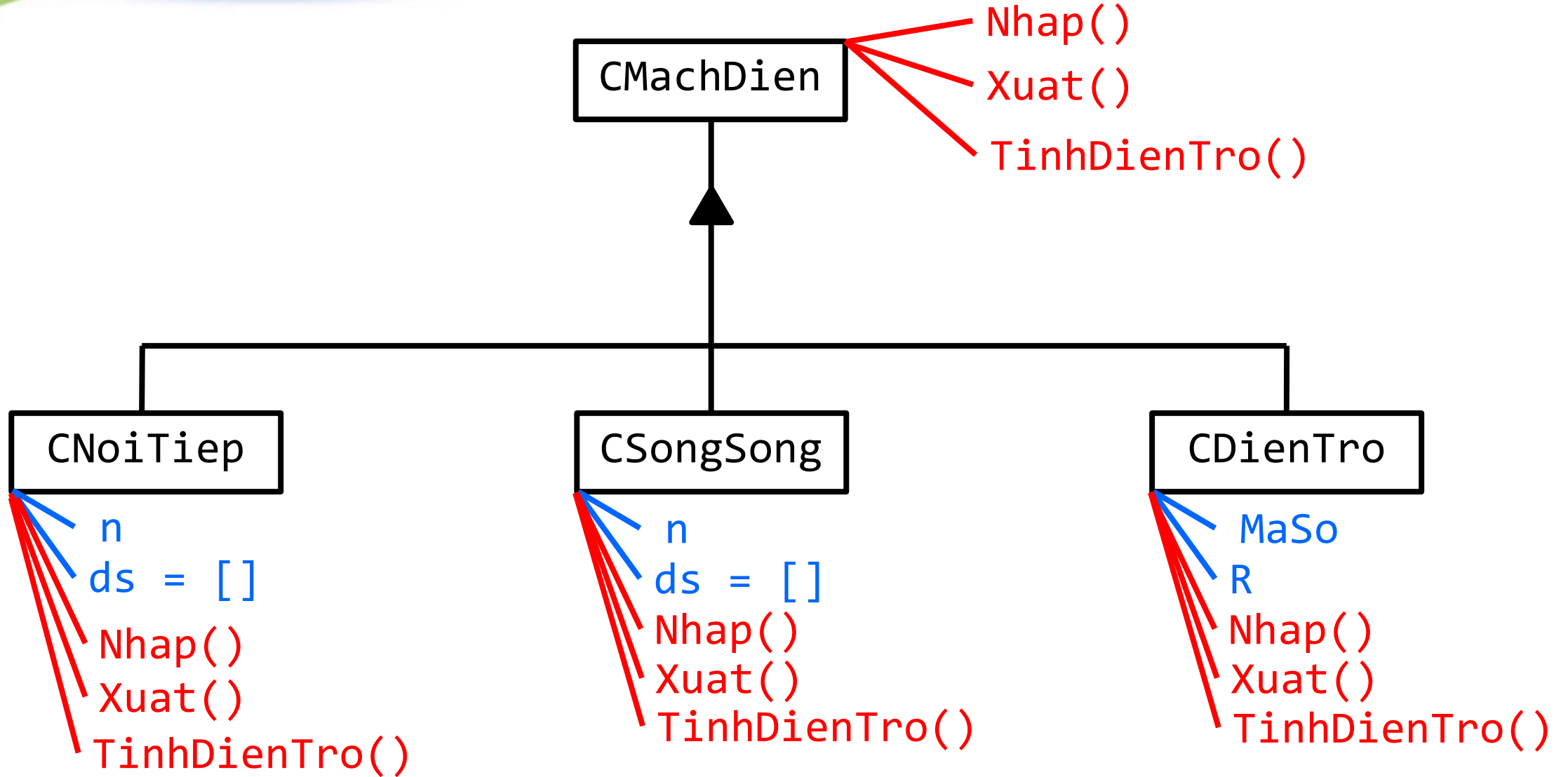
# Bài toán mạch điện

- **Yêu cầu:** Thiết kế các lớp thích hợp để thực hiện các yêu cầu sau:
  - + Nhập thông tin của một mạch điện.
  - + Tính điện trở tương đương.
  - + Xuất thông tin của mạch điện.



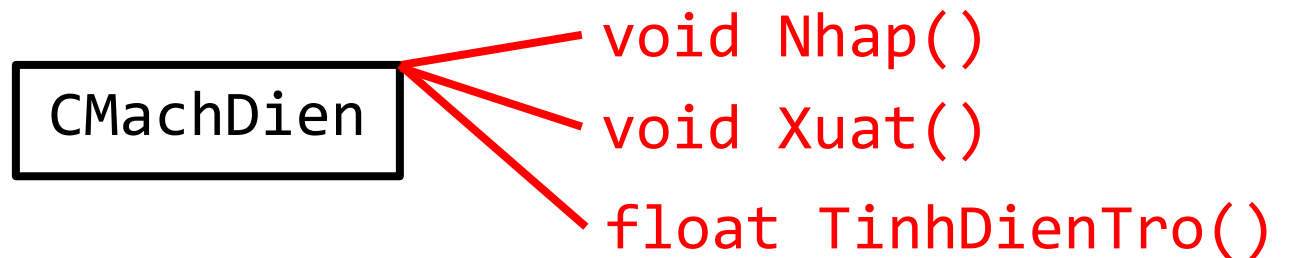
# ĐỊNH NGHĨA PHƯƠNG THỨC KHỞI TẠO





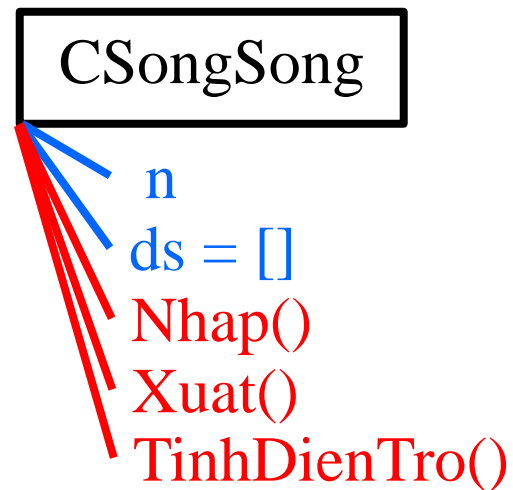
# Định nghĩa phương thức khởi tạo

```
1. class CMachDien:  
2.     def __init__(self):  
3.         return  
4.     ...
```



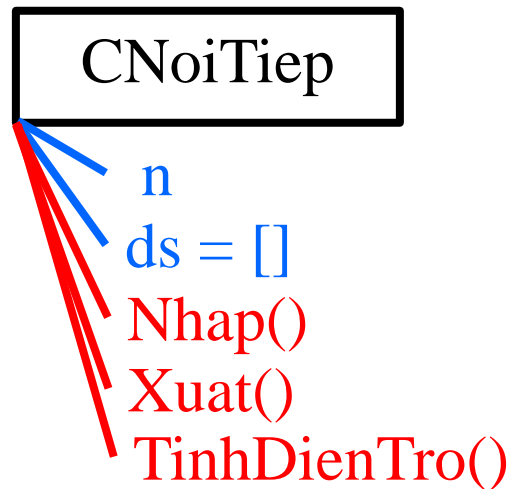
# Định nghĩa phương thức khởi tạo

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



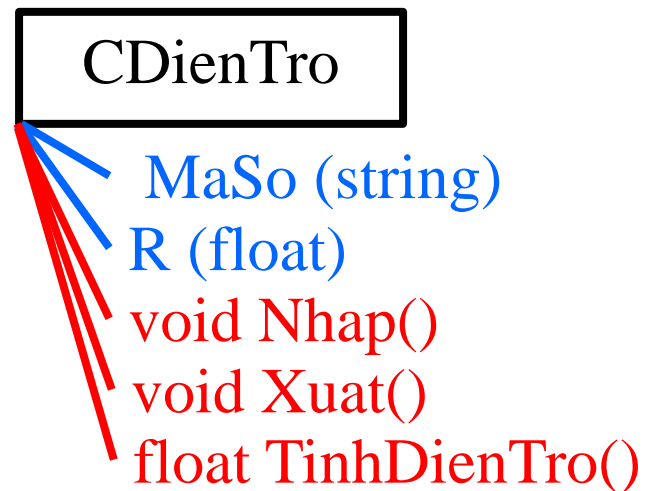
# Định nghĩa phương thức khởi tạo

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

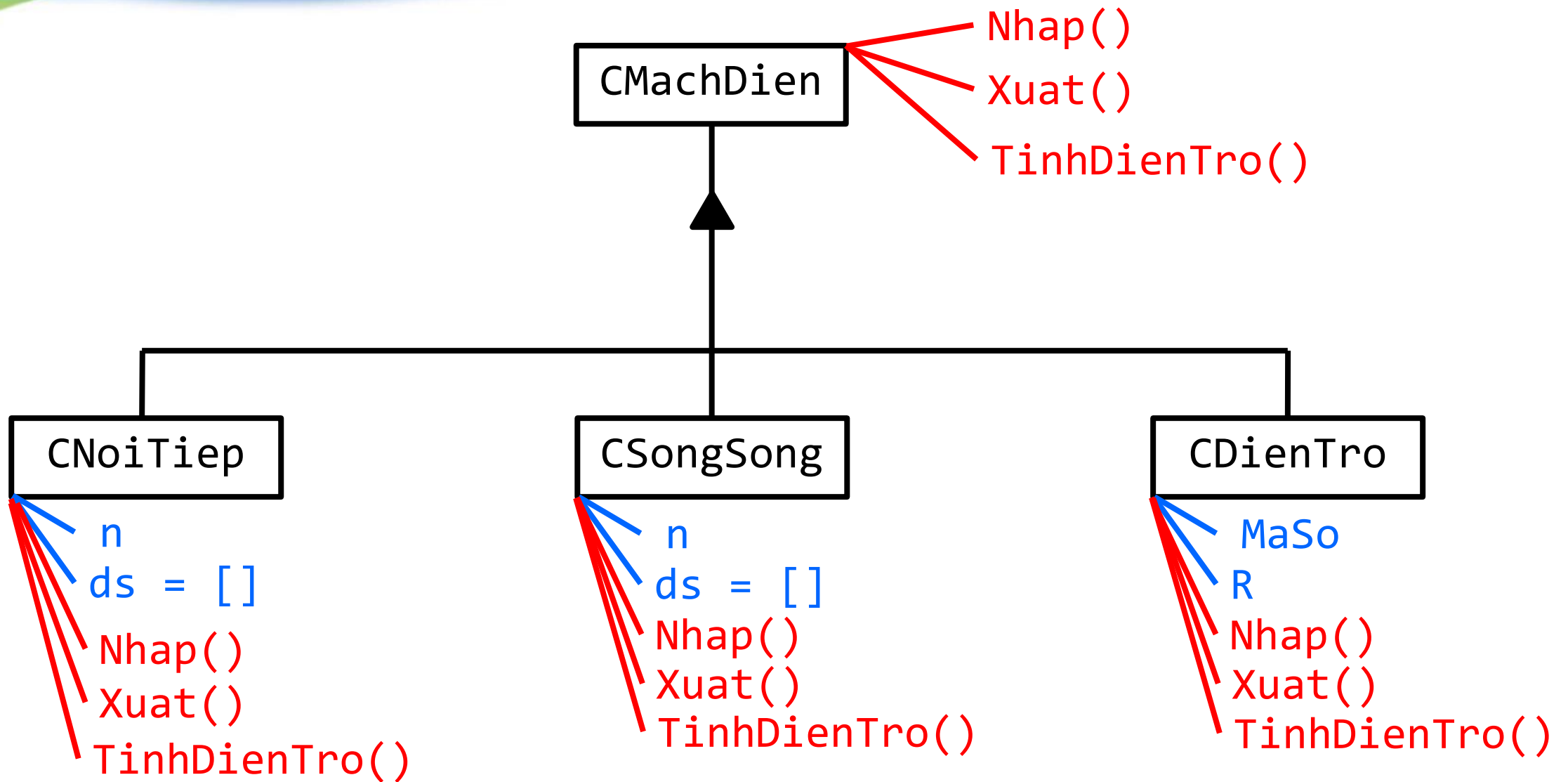


# Khai báo lớp

```
1. class CDienTro(CMachDien):  
2.     def __init__(self):  
3.         self._MaSo = 1  
4.         self._r = 1
```




# ĐỊNH NGHĨA CÁC PHƯƠNG THỨC

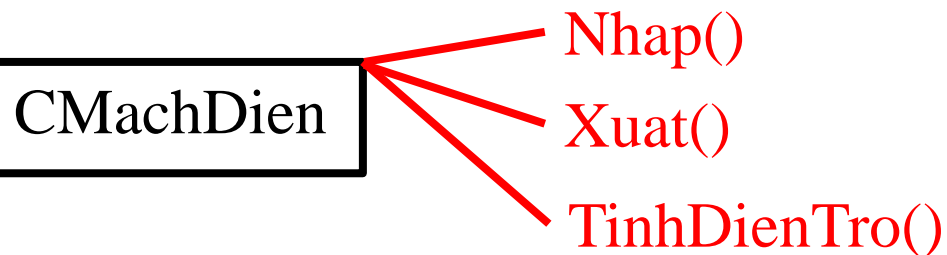


# Định nghĩa phương thức

```
1. class CMachDien:  
2.     ...  
3.     def TinhDienTro(self):  
4.         return 0
```



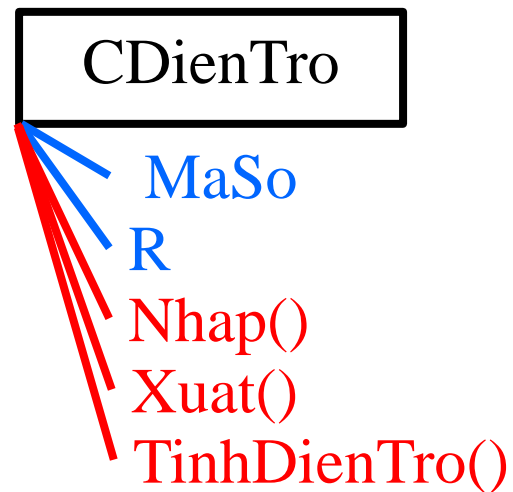
Định nghĩa  
phương thức  
TinhDienTro





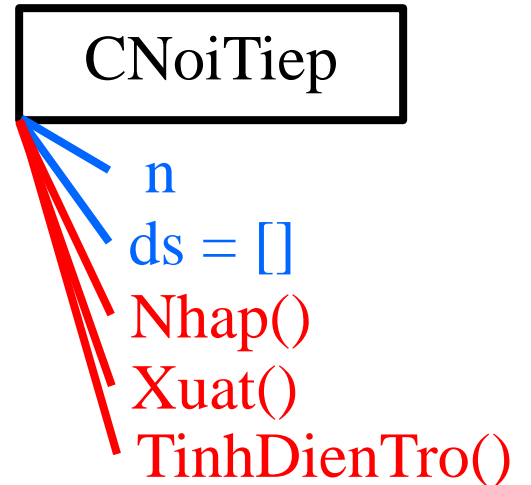
# Định nghĩa phương thức

```
1. class CDienTro(CMachDien):  
2.     ...  
3.     def TinhDienTro(self):  
4.         return self._r
```



# Định nghĩa phương thức

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



# Định nghĩa phương thức

```
1. class CMachSongSong(CMachDien):  
2.     def TinhDienTro(self):  
3.         s = 0  
4.         for x in self._ds:  
5.             s += 1 / x.TinhDienTro()  
6.         return 1 / s
```

CSongSong

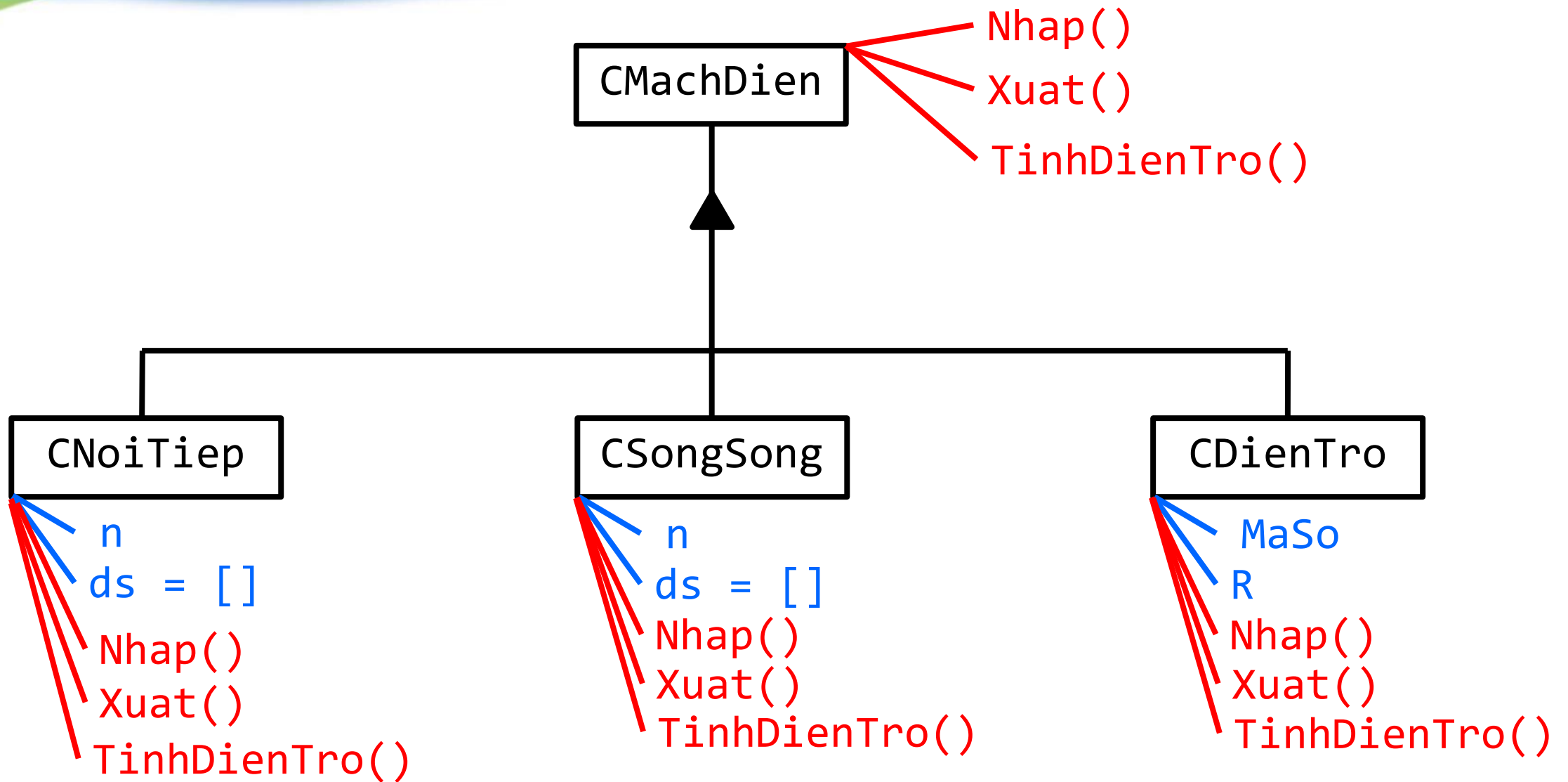
n

ds = []

Nhap()


Xuat()

TinhDienTro()



# Định nghĩa phương thức

```
1. class CMachDien:  
2.     def Nhap(self):  
3.         return
```



Định nghĩa  
phương thức  
Nhap



CMachDien

Nhap()

Xuat()

TinhDienTro()

# Định nghĩa phương thức

```
1. class CDienTro(CMachDien):  
2.     ...  
3.     def Nhap(self):  
4.         self._MaSo = int(input("Nhap ma so: "))  
5.         self._r = int(input("Nhap gia tri r: "))
```

CDienTro

MaSo

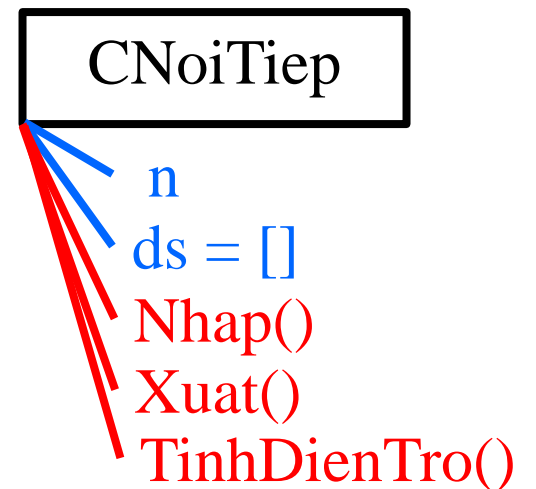
R

Nhap()

Xuat()

TinhDienTro()

```
1. class CMachNoiTiep(CMachDien):
2.     def Nhap(self):
3.         self._n=int(input("Nhap so mach dien thanh phan cua mach noi tiep: "))
4.         for i in range (self._n):
5.             print("Nhap thanh phan thu ", i + 1, "cua mach noi tiep")
6.             type = int(input("Nhap loai (0. Noi tiep, 1. Song song, 2.R): "))
7.             if type == 0:
8.                 temp = CMachNoiTiep()
9.             if type == 1:
10.                from CMachSongSong import CMachSongSong
11.                temp = CMachSongSong()
12.            if type == 2:
13.                from CDienTro import CDienTro
14.                temp = CDienTro()
15.            temp.Nhap()
16.            self._ds.append(temp)
```





```
1. class CMachSongSong(CMachDien):
2.     def Nhap(self):
3.         self._n=int(input("Nhap so mach dien thanh phan cua mach song song: "))
4.         for i in range (self._n):
5.             print("Nhap thanh phan thu", i + 1, "cua mach song song")
6.             type = int(input("Nhap loai (0. Noi tiep, 1. Song song, 2.R): "))
7.             if type == 0:
8.                 from CMachNoiTiep import CMachNoiTiep
9.                 temp = CMachNoiTiep()
10.            if type == 1:
11.                temp = CMachSongSong()
12.            if type == 2:
13.                from CDienTro import CDienTro
14.                temp = CDienTro()
15.            temp.Nhap()
16.            self._ds.append(temp)
```

CSongSong

n

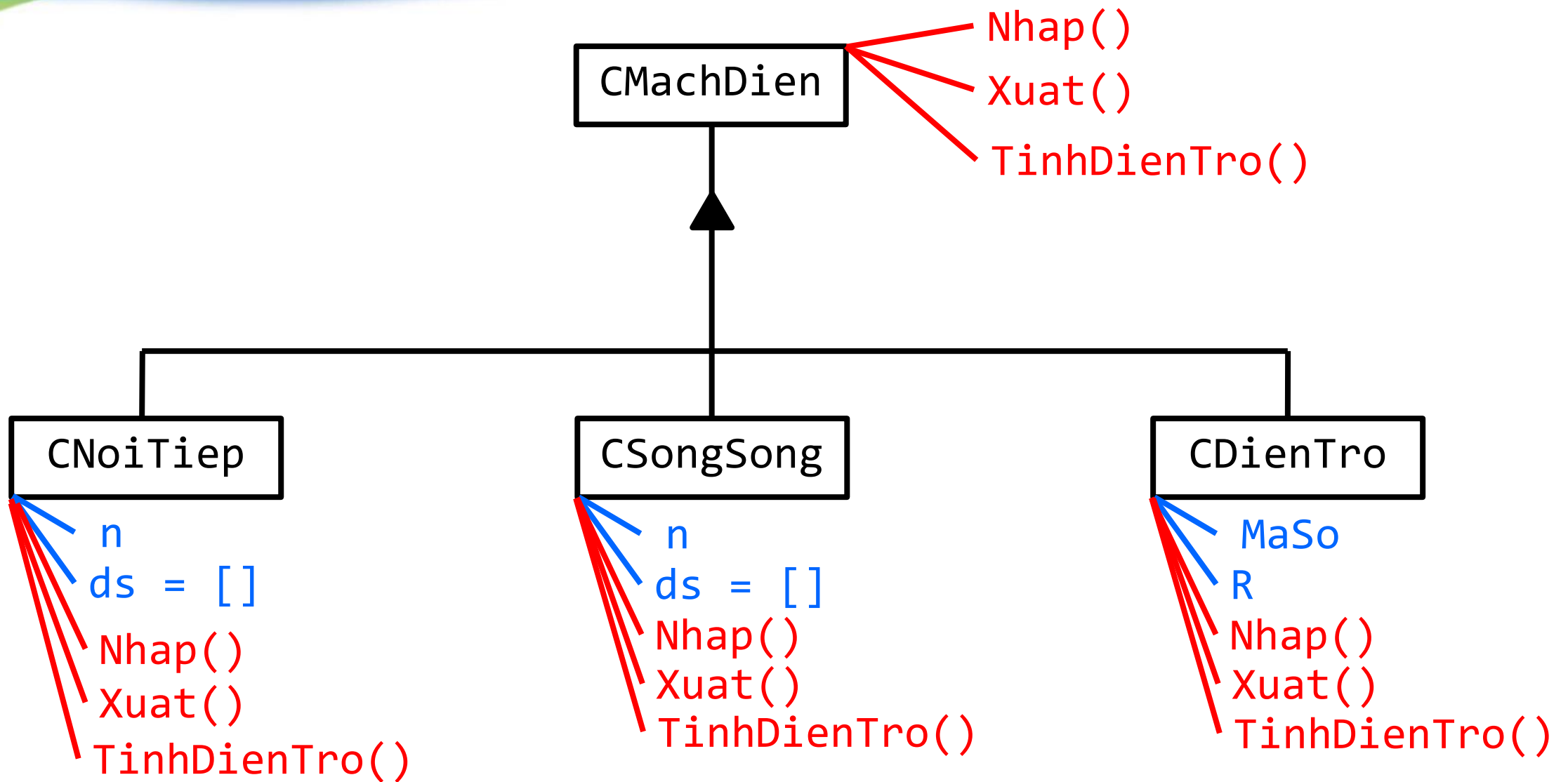
ds = []

Nhap()

Xuat()


TinhDienTro()





# Định nghĩa phương thức

```
1. class CMachDien:  
2.     def Xuat(self):  
3.         return
```



Định nghĩa  
phương thức  
Xuat

CMachDien

void Nhap()

void Xuat()

float TinhDienTro()

# Định nghĩa phương thức

```
1. class CDienTro(CMachDien):  
2.     ...  
3.     def Xuat(self):  
4.         print("Ma so: ", self._MaSo)  
5.         print("R = ", self._r)
```

CDienTro

MaSo (string)

R (float)

void Nhap()

void Xuat()

float TinhDienTro()

# Định nghĩa phương thức

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

CNoiTiep

n

CMachDien\* ds[100]

void Nhap()

void Xuat()

float TinhDienTro()

# Định nghĩa phương thức

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

CSongSong

n

ds = []

Nhap()

Xuat()

TinhDienTro()

# ĐỊNH NGHĨA HÀM MAIN

```
1. def main():
2.     type = int(input("Nhap loai (0. Noi tiep,
3.                     1. Song song, 2.Dien tro): "))
4.     if type == 0:
5.         c = CMachNoiTiep()
6.     if type == 1:
7.         c = CMachSongSong()
8.     if type == 2:
9.         c = CDienTro()
10.    c.Nhap()
11.    ...
```

```
1. def main():  
2.     ...  
3.     print("Mach dien:")  
4.     c.Xuat()  
  
5.     kq = c.TinhDienTro()  
6.     print("Dien tro tuong duong cua mach la:", kq)  
  
7. if __name__ == "__main__":  
8.     main()
```



**Cảm ơn quý vị đã lắng nghe**

**Nhóm tác giả**

**TS. Nguyễn Tấn Trần Minh Khang**