

Chương 9

THIẾT KẾ LỚP NGAY

1. ThS. Nguyễn Hữu Lợi
2. ThS. Nguyễn Văn Toàn
3. TS. Nguyễn Duy Khánh
4. TS. Nguyễn Tấn Trần Minh Khang

1. ĐẶT VẤN ĐỀ

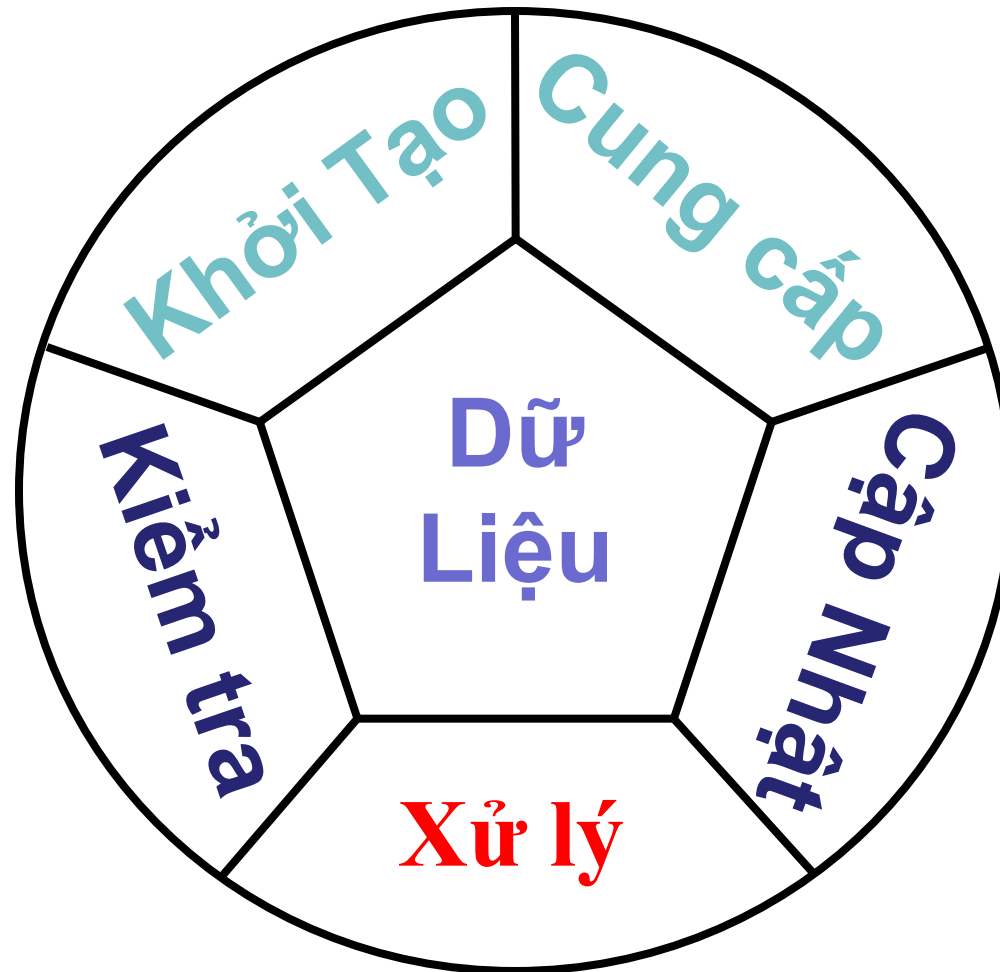
1. Đặt vấn đề

—Hãy thiết kế và xây dựng lớp CNgay trong toán học.



2. PHÂN LOẠI PHƯƠNG THỨC

2. Phân loại phương thức



2. Phân loại phương thức

- Các phương thức (method) của một lớp đối tượng (class) được chia thành 5 loại như sau:
 - + Nhóm các phương thức khởi tạo.
 - + Nhóm các phương thức cung cấp thông tin.
 - + Nhóm các phương thức cập nhật thông tin.
 - + Nhóm các phương thức kiểm tra.
 - + Nhóm các phương thức xử lý.

2. Phân loại phương thức

— Nhóm các phương thức khởi tạo

- + Khởi tạo (**init**) thông tin cho đối tượng.
- + Các phương thức thiết lập (**constructors**) thuộc nhóm các phương thức khởi tạo.
- + Toán tử nhập (operator >>) thuộc nhóm các phương thức khởi tạo.
- + Các loại phương thức khởi tạo.
 - Phương thức khởi tạo mặc định.
 - Phương thức khởi tạo dựa vào đối tượng khác cùng thuộc về lớp.
 - Phương thức khởi tạo khi biết đầy đủ thông tin.

2. Phân loại phương thức

- Nhóm các phương thức cung cấp thông tin
 - + Các phương thức cung cấp thông tin của một lớp đối tượng (**class**) giữ nhiệm vụ cung cấp các thông tin (**attributes, properties, information, data**) của những đối tượng (**objects**) thuộc về lớp cho thế giới bên ngoài.
 - + Toán tử xuất (operator <<) thuộc nhóm các phương thức cung cấp thông tin.
 - + Trong một lớp có bao nhiêu thuộc tính (**attributes, properties**) thì sẽ có ít nhất bấy nhiêu phương thức cung cấp thông tin.
 - + Thông thường các phương thức cung cấp thông tin bắt đầu bằng cụm từ `get`.

2. Phân loại phương thức

- Nhóm các phương thức cập nhật thông tin
 - + Các phương thức cập nhật thông tin của một lớp đối tượng (**class**) giữ nhiệm vụ cập nhật các thông tin (**attributes, properties, information**) của những đối tượng (**objects**) thuộc về lớp khi có sự thay đổi (sự hiệu chỉnh, sự biến động).
 - + Trong một lớp có bao nhiêu thuộc tính (**attributes, properties**) thì sẽ có ít nhất bấy nhiêu phương thức cập nhật thông tin.
 - + Thông thường các phương thức cập nhật thông tin bắt đầu bằng cụm từ **set**.

2. Phân loại phương thức

- Nhóm các phương thức kiểm tra
 - + Các phương thức kiểm tra của một lớp đối tượng (**class**) giữ nhiệm vụ kiểm tra (**test, check**) tính hợp lệ của những thông tin (**attributes, properties, information**) thuộc về các đối tượng, kiểm tra một tính chất nào đó của đối tượng.
 - + Các phương thức toán tử so sánh (relational operators, comparison operators) thuộc nhóm các phương thức kiểm tra.
 - + Thông thường các phương thức kiểm tra bắt đầu bằng cụm từ `is`.

2. Phân loại phương thức

- Nhóm các phương thức xử lý.
 - + Các phương thức xử lý của một lớp đối tượng (class) giữ nhiệm vụ thực hiện các xử lý, tính toán cho lớp đối tượng.
 - + Các toán tử số học (arithmetic operators), toán tử gán (operator =) thuộc nhóm các phương thức xử lý.
 - + Phương thức phá hủy (destructor) thuộc nhóm các phương thức xử lý.

3. THIẾT KẾ LỚP NGÀY

3. Thiết kế lớp ngày

- Thiết kế các thuộc tính của lớp ngày.
- Lớp CNgày có ba thuộc tính là ngày (*ng*), tháng (*th*) và năm (*nm*) với kiểu dữ liệu là số nguyên (*int*).

```

11.class CNgày
12.{
13.    private:
14.        int ng;
15.        int th;
16.        int nm;
17.    ...
18.};

```

3. Thiết kế lớp ngày

- Thiết kế các phương thức cung cấp thông tin.

```

11.class CNgay
12.{
13.    public:
14.        // Nhóm phương thức cung cấp thông tin
15.        void Xuat();
16.        friend ostream& operator << (ostream&, CNgay&);
17.        int getNgay();
18.        int getThang();
19.        int getNam();
20.        ...
21.};

```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức cung cấp thông tin.

```
11. void CNgay::Xuat()
12. {
13. |   cout << "Ngày " << ng << " Tháng " << th << " Năm " << nm;
14. }
```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức cung cấp thông tin.

— Cách 01.

```
11.int CNgay::GetNgay()  
12.{  
13.|    return ng;  
14.}
```

— Cách 02.

```
11.int CNgay::getNgay()  
12.{  
13.|    return this->ng;  
14.}
```

Bên trong thân phương thức của một lớp đối tượng, **this là một con trỏ đối tượng thuộc về lớp mà phương thức đó thuộc về**, con trỏ đối tượng this giữ địa chỉ của đối tượng đang gọi thực hiện phương thức. Hơn nữa, ***this chính là đối tượng đang gọi thực hiện phương thức.**

3. Thiết kế lớp ngày

— Định nghĩa các phương thức cung cấp thông tin.

— Cách 01.

```
11.int CNgay::GetThang()  
12.{  
13.|    return th;  
14.}
```

— Cách 02.

```
11.int CNgay::getThang()  
12.{  
13.|    return this->th;  
14.}
```

Bên trong thân phương thức của một lớp đối tượng, **this** là một con trỏ đối tượng thuộc về lớp mà phương thức đó thuộc về, con trỏ đối tượng **this** giữ địa chỉ của đối tượng đang gọi thực hiện phương thức. Hơn nữa, ***this** chính là đối tượng đang gọi thực hiện phương thức.

3. Thiết kế lớp ngày

— Định nghĩa các phương thức cung cấp thông tin.

— Cách 01.

```
11.int CNgay::GetNam()  
12.{  
13.|    return nm;  
14.}
```

— Cách 02.

```
11.int CNgay::getNam()  
12.{  
13.|    return this->nm;  
14.}
```

Bên trong thân phương thức của một lớp đối tượng, **this** là một con trỏ đối tượng thuộc về lớp mà phương thức đó thuộc về, con trỏ đối tượng **this** giữ địa chỉ của đối tượng đang gọi thực hiện phương thức. Hơn nữa, ***this** chính là đối tượng đang gọi thực hiện phương thức.

3. Thiết kế lớp ngày

— Định nghĩa các phương thức cung cấp thông tin.

```
11. ostream& operator << (ostream& os, CNgay& x)
12. {
13.     os << "Ngày " << x.ng << " Tháng " << x.th << " Năm " << x.nm;
14.     return os;
15. }
```

3. Thiết kế lớp ngày

- Thiết kế các phương thức cập nhật thông tin.

```

11.class CNgay
12.{
13.    public:
14.        // Nhóm phương thức cập nhật thông tin
15.        CNgay& operator =(const CNgay&);
16.        void setNgay(int);
17.        void setThang(int);
18.        void setNam(int);
19.    ...
20.};

```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức cập nhật thông tin.

```

11. CNgay& CNgay::operator=(const CNgay& x)
12. {
13.     ng = x.ng;
14.     th = x.th;
15.     nm = x.nm;
16.     return *this;
17. }

```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức cập nhật thông tin.

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

```
12.{
```

```
13.    |    ng = x.ng;
```

```
14.    |    th = x.th;
```

```
15.    |    nm = x.nm;
```

```
16.    |    return *this;
```

```
17.}
```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức cập nhật thông tin.

```
19. void CNgay::SetNam(int nmnm)
20. {
21. |     nm = nmnm;
22. }
```

3. Thiết kế lớp ngày

— Thiết kế các phương thức kiểm tra.

```

11.class CNgay
12.{
13.    public:
14.        // Nhóm phương thức kiểm tra
15.        int IsHopLe();
16.        int IsNhuan();
17.        ...
18.};

```


3. Thiết kế lớp ngày

— Định nghĩa các phương thức kiểm tra.

```

11.int CNgay::IsHopLe()
12.{
13.    if (ng < 1 || ng > SoNgayToiDaTrongThang())
14.        return 0;
15.    if (th < 1 || th > 12)
16.        return 0;
17.    if (nm < 1)
18.        return 0;
19.    return 1;
20.}

```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức kiểm tra.

```

11.int CNgay::IsNhuan()
12.{
13.    if (nm % 4 == 0 && nm % 100 != 0)
14.        return true;
15.    if (nm % 400 == 0)
16.        return true;
17.    return false;
18.}

```

3. Thiết kế lớp ngày

— Thiết kế các phương thức xử lý.

```

11.class CNgay
12.{
13.    public:
14.        // Nhóm phương thức xử lý
15.        int operator == (const CNgay&);
16.        int operator != (const CNgay&);
17.        int operator > (const CNgay&);
18.        int operator >= (const CNgay&);
19.        int operator < (const CNgay&);
20.        int operator <= (const CNgay&);
21.        ...
22.};

```

3. Thiết kế lớp ngày

— Thiết kế các phương thức xử lý.

```

11.class CNgay
12.{
13.    ...
14.        int SoNgayToiDaTrongThang();
15.        int SoNgayToiDaTrongNam();
16.        int SoThuTuTrongNam();
17.        int SoThuTu();
18.        int SoSanh(CNgay);
19.        int KhoangCach(CNgay);
20.    ...
21.};
    
```

3. Thiết kế lớp ngày

— Thiết kế các phương thức xử lý.

```
11.class CNgay
12.{
13.    ...
14.    CNgay& KeTiep();
15.    CNgay& HomQua();
16.    CNgay& TruocDo(int);
17.    void XuatThu();
18.    ~CNgay();
19.    ...
20.};
```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức xử lý.

```

11.int CNgay::operator >(const CNgay& x)
12.{
13.    if (nm > x.nm)
14.        return 1;
15.    if (th > x.th)
16.        return 1;
17.    if (ng > x.ng)
18.        return 1;
19.    return 0;
20.}

```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức xử lý.

```

11.int CNgay::operator <(const CNgay& x)
12.{
13.    if (nm < x.nm)
14.        return 1;
15.    if (th < x.th)
16.        return 1;
17.    if (ng < x.ng)
18.        return 1;
19.    return 0;
20.}

```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức xử lý.

```

11.int CNgay::operator >=(const CNgay& x)
12.{
13.    if (nm >= x.nm)
14.        return 1;
15.    if (th >= x.th)
16.        return 1;
17.    if (ng >= x.ng)
18.        return 1;
19.    return 0;
20.}

```


3. Thiết kế lớp ngày

— Định nghĩa các phương thức xử lý.

```

11.int CNgay::operator <=(const CNgay& x)
12.{
13.    if (nm <= x.nm)
14.        return 1;
15.    if (th <= x.th)
16.        return 1;
17.    if (ng <= x.ng)
18.        return 1;
19.    return 0;
20.}

```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức xử lý.

```

11.int CNgay::operator !=(const CNgay& x)
12.{
13.    if (nm != x.nm)
14.        return 1;
15.    if (th != x.th)
16.        return 1;
17.    if (ng != x.ng)
18.        return 1;
19.    return 0;
20.}

```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức xử lý.

```
11.int CNgay::operator ==(const CNgay& x)
12.{
13.    if (nm == x.nm)
14.        return 1;
15.    if (th == x.th)
16.        return 1;
17.    if (ng == x.ng)
18.        return 1;
19.    return 0;
20.}
```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức xử lý.

```

11.int CNgay::SoNgayToiDaTrongThang()
12.{
13.    int NgayThang[12]={31,28,31,30,31,30,31,31,30,31,30,31 };
14.    if (IsNhuan())
15.        NgayThang[1] = 29;
16.    return NgayThang[th - 1];
17.}

```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức xử lý.

```
11.int CNgay::SoNgayToiDaTrongNam()
12.{
13.    if (this->IsNhuan() == 1)
14.        return 366;
15.    return 365;
16.}
```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức xử lý.

```

11.int CNgay::SoThuTuTrongNam()
12.{
13.    int stt = 0;
14.    for (int i = 1; i <= th - 1; i++)
15.    {
16.        CNgay temp = { 1,i,nm };
17.        stt = stt + temp.SoNgayToiDaTrongThang();
18.    }
19.    return (stt + ng);
20.}

```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức xử lý.

```

11.int CNgay::SoThuTu()
12.{
13.    int stt = 0;
14.    for (int i = 1; i <= nm - 1; i++)
15.    {
16.        CNgay temp = { 1,1,i };
17.        stt = stt + temp.SoNgayToiDaTrongNam();
18.    }
19.    return (stt + SoThuTuTrongNam());
20.}

```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức xử lý.

```
11.int CNgay::SoSanh(CNgay x)
```

```
12.{
```

```
13.    if (nm > x.nm)
```

```
14.        return 1;
```

```
15.    if (nm < x.nm)
```

```
16.        return -1;
```

```
17.    if (th > x.th)
```

```
18.        return 1;
```

```
19.    if (th < x.th)
```

```
20.        return -1;
```

```
21.    if (ng > x.ng)
```

```
22.        return 1;
```

```
23.    if (ng < x.ng)
```

```
24.        return -1;
```

```
25.    return 0;
```

```
26.}
```


3. Thiết kế lớp ngày

— Định nghĩa các phương thức xử lý.

```
11.int CNgay::KhoangCach(CNgay x)
12.{
13.    int a = SoThuTu();
14.    int b = x.SoThuTu();
15.    return abs(a - b);
16.}
```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức xử lý.

```
11. void CNgay::XuatThu()
12. {
13.     int stt = SoThuTu();
14.     switch (stt % 7)
15.     {
16.     case 0: cout<<"Chu nhat";
17.         break;
18.     case 1: cout<<"Thu hai";
19.         break;
```

```
20.     case 2: cout<<"Thu ba";
21.         break;
22.     case 3: cout<<"Thu tu";
23.         break;
24.     case 4: cout<<"Thu nam";
25.         break;
26.     case 5: cout<<"Thu sau";
27.         break;
28.     case 6: cout<<"Thu bay";
29.         break;
30. }
```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức xử lý.

11. CNgay& CNgay::KeTiep()

12. {

13. int NgayThang[12] =
 {31,28,31,30,31,30,31,
 31,30,31,30,31 };

14. if (IsNhuan())

15. NgayThang[1] = 29;

16. ng++;

17. if (ng > NgayThang[th-1])

18. {

19. th++;

20. if (th > 12)

21. {

22. nm++;

23. th = 1;

24. }

25. ng = 1;

26. }

27. return *this;

28. }

3. Thiết kế lớp ngày

— Định nghĩa các phương thức xử lý.

11. CNgay& CNgay::HomQua()

12. {

13. int NgayThang[12] =
 {31,28,31,30,31,30,31,
 31,30,31,30,31 };

14. if (IsNhuan())

15. NgayThang[1] = 29;

16. ng--;

17. if (ng == 0)

18. {

19. th--;

20. if (th == 0)

21. {

22. nm--;

23. th = 12;

24. }

25. ng = NgayThang[th- 1];

26. }

27. return *this;

28. }

3. Thiết kế lớp ngày

— Định nghĩa các phương thức xử lý.

```

11. CNgay& CNgay::TruocDo(int k)
12. {
13.     for (int i = 1; i <= k; i++)
14.         *this = this->HomQua();
15.     return *this;
16. }

```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức xử lý.

```
11. CNgay::~CNgay()
```

```
12. {
```

```
13. |     return;
```

```
14. }
```

3. Thiết kế lớp ngày

— Thiết kế các phương thức khởi tạo.

```

11.class CNgay
12.{
13.    ...
14.    // Nhóm phương thức khởi tạo
15.    void KhoiTao();
16.    void KhoiTao(int, int, int);
17.    void KhoiTao(const CNgay&);
18.    ...
19.};
    
```

3. Thiết kế lớp ngày

— Thiết kế các phương thức khởi tạo.

```

11.class CNgay
12.{
13.    ...
14.        void Nhap();
15.        CNgay();
16.        CNgay(int, int, int);
17.        CNgay(const CNgay&);
18.        friend istream& operator >> (istream&, CNgay&);
19.    ...
20.};
    
```


3. Thiết kế lớp ngày

— Thiết kế các phương thức khởi tạo.

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

```
12. {
```

```
13. |     ng = 1;
```

```
14. |     th = 1;
```

```
15. |     nm = 1;
```

```
16. }
```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức khởi tạo.

```
11. void CNgay::KhoiTao(int ngng, int thth, int nmnm)
12. {
13.     ng = ngng;
14.     th = thth;
15.     nm = nmnm;
16. }
```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức khởi tạo.

```
11. void CNgay::KhoiTao(const CNgay& x)
12. {
13.     ng = x.ng;
14.     th = x.th;
15.     nm = x.nm;
16. }
```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức khởi tạo.

```
11. void CNgay::Nhap()
12. {
13.     cout << "Nhap ngay: ";
14.     cin >> ng;
15.     cout << "Nhap thang: ";
16.     cin >> th;
17.     cout << "Nhap nam: ";
18.     cin >> nm;
19. }
```

3. Thiết kế lớp ngày

— Định nghĩa các phương thức khởi tạo.

```
11. CNgay::CNgay()
```

```
12. {
```

```
13. |     ng = 1;
```

```
14. |     th = 1;
```

```
15. |     nm = 1;
```

```
16. }
```

Phương thức thiết lập sao chép, không nhận một tham số đầu vào, các thông tin ban đầu của đối tượng được thiết lập như sau: ngày (ng) lấy giá trị 1, tháng (th) lấy giá trị 1, năm (nm) lấy giá trị 1.

3. Thiết kế lớp ngày

— Định nghĩa các phương thức khởi tạo.

```
11. CNgay::CNgay(int ngng, int thth, int nmnm)
12. {
13.     ng = ngng;
14.     th = thth;
15.     nm = nmnm;
16. }
```

Phương thức thiết lập sao chép, nhận ba tham số đầu vào là `ngng`, `thth`, `nmnm`, các thông tin ban đầu của đối tượng được thiết lập như sau: ngày (`ng`) lấy giá trị `ngng`, tháng (`th`) lấy giá trị `thth`, năm (`nm`) lấy giá trị `nmnm`.

3. Thiết kế lớp ngày

— Định nghĩa các phương thức khởi tạo.

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

```
12. {
```

```
13.     ng = x.ng;
```

```
14.     th = x.th;
```

```
15.     nm = x.nm;
```

```
16. }
```

Phương thức thiết lập sao chép, nhận một tham số đầu vào là x là đối tượng thuộc lớp `CNgay`, các thông tin ban đầu của đối tượng được thiết lập như sau: ngày (ng) lấy giá trị $x.ng$, tháng (th) lấy giá trị $x.th$, năm (nm) lấy giá trị $x.nm$.

3. Thiết kế lớp ngày

— Định nghĩa các phương thức khởi tạo.

```
11.istream& operator >> (istream& is, CNgay& x)
12.{
13.    cout << "Nhap ngay: ";
14.    is >> x.ng;
15.    cout << "Nhap thang: ";
16.    is >> x.th;
17.    cout << "Nhap nam: ";
18.    is >> x.nm;
19.    return is;
20.}
```


4. REVIEW

```
11. class CNgay
```

```
12. {
```

```
13.     private:
```

```
14.         int ng;
```

```
15.         int th;
```

```
16.         int nm;
```

```
17.     public:
```

```
18.         // Nhóm khởi tạo
```

```
19.         void KhoiTao();
```

```
20.         void KhoiTao(int, int, int);
```

```
21.         void KhoiTao(const CNgay&);
```

```
22.         ...
```

```
23. };
```

```
11. class CNgay
```

```
12. {
```

```
13.     ...
```

```
14.     void Nhap();
```

```
15.     CNgay();
```

```
16.     CNgay(int, int, int);
```

```
17.     CNgay(const CNgay&);
```

```
18.     friend istream& operator >> (istream&, CNgay&);
```

```
19.     ...
```

```
20. };
```

```
11. class CNgay
```

```
12. {
```

```
13.     ...
```

```
14.     // Nhóm cung cấp thông tin
```

```
15.     void Xuat();
```

```
16.     friend ostream& operator << (ostream&, CNgay&);
```

```
17.     int getNgay();
```

```
18.     int getThang();
```

```
19.     int getNam();
```

```
20.     ...
```

```
21. };
```

```
11. class CNgay
```

```
12. {
```

```
13.     ...
```

```
14.     // Nhóm cập nhật thông tin
```

```
15.     CNgay& operator =(const CNgay&);
```

```
16.     void setNgay(int);
```

```
17.     void setThang(int);
```

```
18.     void setNam(int);
```

```
19.     ...
```

```
20. };
```

```
11. class CNgay
```

```
12. {
```

```
13.     ...
```

```
14.     // Nhóm xử lý
```

```
15.     int operator == (const CNgay&);
```

```
16.     int operator != (const CNgay&);
```

```
17.     int operator > (const CNgay&);
```

```
18.     int operator >= (const CNgay&);
```

```
19.     int operator < (const CNgay&);
```

```
20.     int operator <= (const CNgay&);
```

```
21.     ...
```

```
22. };
```

```
11. class CNgay
12. {
13.     ...
14.     int SoNgayToiDaTrongThang();
15.     int SoNgayToiDaTrongNam();
16.     int SoThuTuTrongNam();
17.     int SoThuTu();
18.     int SoSanh(CNgay);
19.     int KhoangCach(CNgay);
20.     CNgay& KeTiep();
21.     CNgay& HomQua();
22.     CNgay& TruocDo(int);
23.     void XuatThu();
24.     ~CNgay();
25. };
```

Lớp đối tượng ngày CNgay

— Thuộc tính

- + Ngày.
- + Tháng.
- + Năm.

— Phương thức

- + Nhóm phương thức khởi tạo.
- + Nhóm phương thức cung cấp thông tin.
- + Nhóm phương thức cập nhật thông tin.
- + Nhóm phương thức xử lý.
- + Nhóm phương thức kiểm tra.



Lớp đối tượng ngày CNgay

— Nhóm phương thức khởi tạo

- + Phương thức Nhập.
- + Toán tử vào.
- + Phương thức thiết lập mặc định.
- + Phương thức thiết lập sao chép.
- + Phương thức thiết lập khi biết đầy đủ thông tin.



Lớp đối tượng ngày CNgay

- Nhóm phương thức cung cấp thông tin
 - + Phương thức Xuất.
 - + Toán tử ra.
 - + Phương thức cung cấp ngày.
 - + Phương thức cung cấp tháng.
 - + Phương thức cung cấp năm.



Lớp đối tượng ngày CNgay

— Nhóm phương thức cập nhật thông tin

- + Toán tử gán.
- + Phương thức cập nhật ngày.
- + Phương thức cập nhật tháng.
- + Phương thức cập nhật năm.



Lớp đối tượng CNgay

– Nhóm phương thức xử lý

- + 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

+ Tiêu chuẩn so sánh dựa vào lần lượt năm->tháng->ngày

- Năm lớn hơn thì lớn hơn
- Tháng lớn hơn thì lớn hơn
- Ngày lớn hơn thì lớn hơn



Lớp đối tượng CNgay

— Nhóm phương thức xử lý

- + Số ngày tối đa trong tháng.
- + Số ngày tối đa trong năm.
- + Số thứ tự trong năm.
- + Số thứ tự tính từ ngày 01/01/01.
- + So sánh hai ngày trong năm.
- + Khoảng cách giữa hai ngày.
- + Tìm ngày kế tiếp.
- + Tìm ngày kế tiếp trong năm.
- + Tìm ngày hôm qua trong năm.
- + Tìm ngày trước đó k ngày.
- + Xuất thứ.
- + Phương thức phá hủy



Lớp đối tượng CNgay

— Nhóm phương thức kiểm tra

+ Kiểm tra ngày có hợp lệ hay không?

+ Kiểm tra năm có phải là năm nhuận hay không?



Cảm ơn quý vị đã lắng nghe

Nhóm tác giả

TS. Nguyễn Tấn Trần Minh Khang