



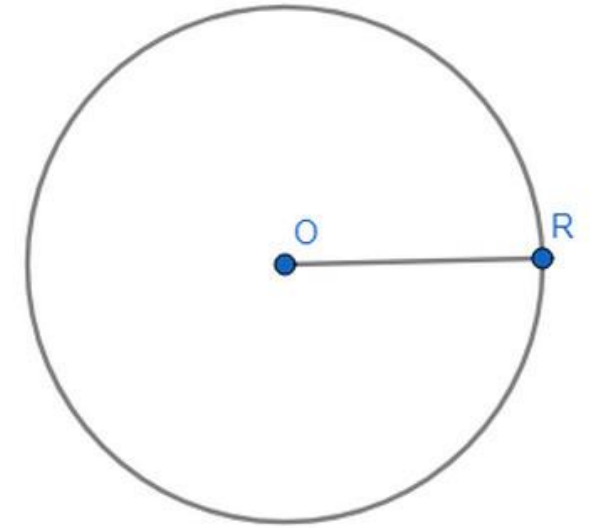
Thiết kế lớp đường tròn

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

Thiết kế lớp đối tượng CDuongTron



- Thuộc tính
 - + Tâm.
 - + Bán kính.



Thiết kế lớp đối tượng CDuongTron



```
11.class CDuongTron
12.{
13.    private:
14.        CDiem I;
15.        float R;
16.    public:
```

Thiết kế lớp đối tượng CDuongTron



- Thuộc tính
 - + Tâm.
 - + Bán kính.
- 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.

Thiết kế lớp đối tượng CDuongTron



```
11.class CDuongTron
12.{
13.    private:
14.        CDiem I;
15.        float R;
16.    public:
17.        // Nhóm phương thức khởi tạo
18.        // Nhóm phương thức cung cấp thông tin
19.        // Nhóm phương thức cập nhật thông tin
20.        // Nhóm phương thức kiểm tra
21.        // Nhóm phương thức xử lý
```

Thiết kế lớp đối tượng CDuongTron



— Nhóm phương thức khởi tạo

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

Thiết kế lớp đối tượng CDuongTron



```
11. // Nhóm các phương thức khởi tạo
12. void Nhap();
13. friend istream& operator >>(istream&,
                                CDuongTron&);
14. void KhoiTao();
15. void KhoiTao(const CDiem&, float);
16. void KhoiTao(const CDuongTron&);
17. CDuongTron();
18. CDuongTron(const CDiem&, float);
19. CDuongTron(const CDuongTron&);
```

Thiết kế lớp đối tượng CDuongTron



- Nhóm phương thức cung cấp thông tin
 - + Phương thức Xuất.
 - + Toán tử ra.
 - + Phương thức cung cấp tâm.
 - + Phương thức cung cấp bán kính.

Thiết kế lớp đối tượng CDuongTron



```
11. //Nhom cac phuong thuc cung cap thong tin
12. void Xuat();
13. friend ostream& operator << (ostream&,
                                CDuongTron&);
14. CDiem getI();
15. float getR();
```

Thiết kế lớp đối tượng CDuongTron



- Nhóm phương thức cập nhật thông tin
 - + Toán tử gán.
 - + Phương thức cập nhật tâm.
 - + Phương thức cập nhật bán kính.

Thiết kế lớp đối tượng CDuongTron



```
11. | // Nhóm các phương thức cập nhật thông tin
12. | CDuongTron& operator =(const CDuongTron&);
13. | void setI(const CDiem&);
14. | void setR(float);
```

Thiết kế lớp đối tượng CDuongTron



— Nhóm phương thức kiểm tra

- + Kiểm tra hai đường tròn có trùng nhau không?
- + Kiểm tra hai đường tròn có cắt nhau không?
- + Kiểm tra hai đường tròn có tiếp xúc trong không?
- + Kiểm tra hai đường tròn có tiếp xúc ngoài không?
- + Kiểm tra hai đường tròn có nằm trong không?
- + Kiểm tra hai đường tròn có nằm ngoài không?
- + Kiểm tra hai đường tròn có đồng tâm không?
- + Kiểm tra hai đường tròn có đồng tâm không?

Thiết kế lớp đối tượng CDuongTron



```
11. //Nhom cac phuong thuc kiem tra
12. int isTrung(const CDuongTron&);
13. int isCat(const CDuongTron&);
14. int isTiepXucTrong(const CDuongTron&);
15. int isTiepXucNgoai(const CDuongTron&);
16. int isNamTrong(const CDuongTron&);
17. int isNamNgoai(const CDuongTron&);
18. int isDongTam(const CDuongTron&);
19. int isDonVi(const CDuongTron&);
```

Thiết kế lớp đối tượng CDuongTron



- Nhóm phương thức xử lý
 - + Tính chu vi
 - + Tính diện tích
 - + 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
 - + Phương thức phá hủy
 - + Tiêu chuẩn so sánh dựa vào bán kính

Thiết kế lớp đối tượng CDuongTron



```
11. // Nhóm các phương thức xử lý
12. float ChuVi();
13. float DienTich();
14. int operator == (CDuongTron);
15. int operator != (CDuongTron);
16. int operator > (CDuongTron);
17. int operator >= (CDuongTron);
18. int operator < (CDuongTron);
19. int operator <= (CDuongTron);
20. ~CDuongTron();
```

Thiết kế lớp đối tượng CDuongTron



- Thiết kế các thuộc tính của lớp phân số.
- Lớp CDuongTron có hai thuộc tính là tâm (I) với kiểu dữ liệu là CDiem và bán kính(R) với kiểu dữ liệu là số thực (float).

```
11.class CDuongTron
12.{
13.    private:
14.        CDiem I;
15.        float R;
16.        ...
17.};
```


Thiết kế lớp đối tượng CDuongTron



— Thiết kế các phương thức cung cấp thông tin.

```
11.class CDuongTron
12.{
13.    //Nhóm các phương thức cung cấp thông tin
14.    void Xuat();
15.    friend ostream& operator << (ostream&, CDuongTron&);
16.    CDiem getI();
17.    float getR();
18.};
```

Thiết kế lớp đối tượng CDuongTron



— Định nghĩa các phương thức cung cấp thông tin.

```
11. void CDuongTron::Xuat()  
12. {  
13. |   cout << "\\Tam: " << I;  
14. |   cout << "\\nBan Kinh: " << R;  
15. }
```

Thiết kế lớp đối tượng CDuongTron



— Định nghĩa các phương thức cung cấp thông tin.

```
11. ostream& operator << (ostream& os, CDuongTron& c)
12. {
13.     os << "\Tam: " << c.I;
14.     os << "\nBan Kinh: " << c.R;
15.     return os;
16. }
```

Thiết kế lớp đối tượng CDuongTron



— Định nghĩa các phương thức cung cấp thông tin.

— Cách 01.

```
11.CDiem CDuongTron::getI()  
12.{  
13.|    return I;  
14.}
```

— Cách 02.

```
11.CDiem CDuongTron::getI()  
12.{  
13.|    return this->I;  
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.

Thiết kế lớp đối tượng CDuongTron



— Định nghĩa các phương thức cung cấp thông tin.

— Cách 01.

```
11.float CDuongTron::getR()
```

```
12.{
```

```
13.|    return R;
```

```
14.}
```

15. Cách 02.

```
16.float CDuongTron::getR()
```

```
17.{
```

```
18.|    return thR;
```

```
19.}
```

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.

Thiết kế lớp đối tượng CDuongTron



— Thiết kế các phương thức cập nhật thông tin.

```
11.class CDuongTron
12.{
13.    // Nhóm các phương thức cập nhật thông tin
14.    CDuongTron& operator =(const CDuongTron&);
15.    void setI(const CDiem&);
16.    void setR(float);
17.};
```

Thiết kế lớp đối tượng CDuongTron



— Định nghĩa các phương thức cập nhật thông tin.

```
11. CDuongTron& CDuongTron::operator= (const CDuongTron& c)
```

```
12. {
```

```
13. |     *this = c;
```

```
14. |     return *this;
```

```
15. }
```

Thiết kế lớp đối tượng CDuongTron



— Định nghĩa các phương thức cập nhật thông tin.

```
11. void CDuongTron::setI(const CDiem& P)
```

```
12. {
```

```
13. |     I = P;
```

```
14. }
```

```
15. void CDuongTron::setR(float rr)
```

```
16. {
```

```
17. |     R = rr;
```

```
18. }
```


Thiết kế lớp đối tượng CDuongTron



— Thiết kế các phương thức kiểm tra.

```
11.class CDuongTron
12.{
13.    //Nhom cac phuong thuc kiem tra
14.    int isTrung(const CDuongTron&);
15.    int isCat(const CDuongTron&);
16.    int isTiepXucTrong(const CDuongTron&);
17.    int isTiepXucNgoai(const CDuongTron&);
18.    int isNamTrong(const CDuongTron&);
19.    int isNamNgoai(const CDuongTron&);
20.    ...
21.};
```

Thiết kế lớp đối tượng CDuongTron



— Thiết kế các phương thức kiểm tra.

```
11.class CDuongTron
12.{
13.    //Nhóm các phương thức kiểm tra
14.    ...
15.    int isDongTam(const CDuongTron&);
16.    int isDonVi(const CDuongTron&);
17.};
```

Thiết kế lớp đối tượng CDuongTron



— Định nghĩa các phương thức kiểm tra.

```
11.int CDuongTron::isTrung(const CDuongTron &c)
12.{
13.|    return (*this == c);
14.}
```

Thiết kế lớp đối tượng CDuongTron



— Định nghĩa các phương thức kiểm tra.

```
11.int CDuongTron::isCat(const CDuongTron& c)
12.{
13.    float kc = I.KhoangCach(c.I);
14.    return ((R - c.R) < kc && kc < (R + c.R));
15.}
```

Thiết kế lớp đối tượng CDuongTron



— Định nghĩa các phương thức kiểm tra.

```
11.int CDuongTron::isTiepXucTrong(const CDuongTron& c)
12.{
13.    float kc = I.KhoangCach(c.I);
14.    return (abs(R - c.R) == kc);
15.}
```

Thiết kế lớp đối tượng CDuongTron



— Định nghĩa các phương thức kiểm tra.

```
11.int CDuongTron::isTiepXucNgoai(const CDuongTron& c)
12.{
13.    float kc = I.KhoangCach(c.I);
14.    return ((R + c.R) == kc);
15.}
```

3. Thiết kế lớp phân số



— Định nghĩa các phương thức kiểm tra.

```
11.int CDuongTron::isNamTrong(const CDuongTron& c)
12.{
13.    float kc = I.KhoangCach(c.I);
14.    return (abs(R - c.R) < kc);
15.}
```

3. Thiết kế lớp phân số



— Định nghĩa các phương thức kiểm tra.

```
11.int CDuongTron::isNamNgoai(const CDuongTron& c)
12.{
13.    float kc = I.KhoangCach(c.I);
14.    return (R + c.R < kc);
15.}
```


3. Thiết kế lớp phân số



— Định nghĩa các phương thức kiểm tra.

```
11.int CDuongTron::isDongTam(const CDuongTron &c)
12.{
13.|    return (R==c.R);
14.}
```

3. Thiết kế lớp phân số



— Định nghĩa các phương thức kiểm tra.

```
11.int CDuongTron::isDonVi(const CDuongTron &c)
12.{
13.    |   CDiem goc(0, 0);
14.    |   CDuongTron temp(goc, 1);
15.    |   return (temp == c);
16.}
```

Thiết kế lớp đối tượng CDuongTron



— Thiết kế các phương thức khởi tạo.

```
11.class CDuongTron
12.{
13.    // Nhóm các phương thức khởi tạo
14.    void Nhap();
15.    friend istream& operator >>(istream&, CDuongTron&);
16.    void KhoiTao();
17.    void KhoiTao(const CDiem&, float);
18.    void KhoiTao(const CDuongTron&);
19.};
```

Thiết kế lớp đối tượng CDuongTron



— Thiết kế các phương thức khởi tạo.

```
11.class CDuongTron
```

```
12.{
```

```
13.    // Nhóm các phương thức khởi tạo
```

```
14.    CDuongTron();
```

```
15.    CDuongTron(const CDiem&, float);
```

```
16.    CDuongTron(const CDuongTron&);
```

```
17.};
```

Thiết kế lớp đối tượng CDuongTron



— Định nghĩa các phương thức khởi tạo.

```
11. void CDuongTron::Nhap()  
12. {  
13.     cout << "\\Nhap tam duong tron: ";  
14.     cin >> I;  
15.     cout << "\\Nhap ban kinh: ";  
16.     cin>>R;  
17. }
```

Thiết kế lớp đối tượng CDuongTron



— Định nghĩa các phương thức khởi tạo.

```
11.istream& operator >>(istream& is, CDuongTron& c)
12.{
13.    cout << "\Nhap tam duong tron: ";
14.    is >> c.I;
15.    cout << "\Nhap ban kinh: ";
16.    is >> c.R;
17.    return is;
18.}
```

Thiết kế lớp đối tượng CDuongTron



- Định nghĩa các phương thức khởi tạo.

```
11. void CDuongTron::KhoiTao()  
12. {  
13. |   I = CDiem(0, 0);  
14. |   R = 0;  
15. }
```

- Phương thức khởi tạo mặc định, không nhận tham số đầu vào, các thông tin ban đầu của đối tượng được thiết lập mặc định như sau: Tâm là (0,0), bán kính 0

Thiết kế lớp đối tượng CDuongTron



- Định nghĩa các phương thức khởi tạo.

```
11. void CDuongTron::KhoiTao(const CDiem& P, float rr)
12. {
13. |   I = P;
14. |   R = rr;
15. }
```

- Phương thức khởi tạo khi biết tử, nhận hai tham số đầu vào là P , R các thông tin ban đầu của đối tượng được thiết lập như sau: tâm (I) lấy giá trị P , bán kính (R) lấy giá trị rr .

Thiết kế lớp đối tượng CDuongTron



- Định nghĩa các phương thức khởi tạo.

```
11. void CDuongTron::KhoiTao(const CDuongTron&c)
12. {
13. |     *this = c;
14. }
```

- Phương thức khởi tạo dựa vào đối tượng khác cùng thuộc về lớp, nhận một tham số đầu vào là `c` là đối tượng thuộc lớp `CDuongTron`, các thông tin ban đầu của đối tượng được thiết lập như sau: tâm (`I`) lấy giá trị `c.I`, bán kính (`R`) lấy giá trị `c.R`.

Thiết kế lớp đối tượng CDuongTron



- Định nghĩa các phương thức khởi tạo.

```
11. CDuongTron::CDuongTron()
```

```
12. {
```

```
13. |     I = CDiem(0, 0);
```

```
14. |     R = 0;
```

```
15. }
```

- Phương thức thiết lập mặc định, không nhận tham số đầu vào, các thông tin ban đầu của đối tượng được thiết lập mặc định như sau: tâm (I) lấy giá trị (0,0), bán kính(R) lấy giá trị 0.

Thiết kế lớp đối tượng CDuongTron



- Định nghĩa các phương thức khởi tạo.

```
11. CDuongTron::CDuongTron(const CDiem& P, float rr)
```

```
12. {
```

```
13. |     R = rr;
```

```
14. |     I = P;
```

```
15. }
```

- Phương thức thiết lập khi biết tử, nhận 2 tham số đầu vào là P, R các thông tin ban đầu của đối tượng được thiết lập như sau: tâm (I) lấy giá trị P , bán kính (R) lấy giá trị mặc định là rr .

Thiết kế lớp đối tượng CDuongTron



— Định nghĩa các phương thức khởi tạo.

```
11. CDuongTron::CDuongTron(const CDuongTron& c)
```

```
12. {
```

```
13. |     *this = c;
```

```
14. }
```

— Phương thức thiết lập khi biết đầy đủ thông tin, nhận một tham số đầu vào là c các thông tin ban đầu của đối tượng được thiết lập như sau: tâm (I) lấy giá trị $c.I$, bán kính (R) lấy giá trị $c.R$.

Thiết kế lớp đối tượng CDuongTron



— Thiết kế các phương thức xử lý.

```
11.class CDuongTron
12.{
13.    // Nhóm các phương thức xử lý
14.    float ChuVi();
15.    float DienTich();
16.};
```

Thiết kế lớp đối tượng CDuongTron



— Thiết kế các phương thức xử lý.

```
11.class CDuongTron
12.{
13.    // Nhóm các phương thức xử lý
14.    int operator == (CDuongTron);
15.    int operator != (CDuongTron);
16.    int operator > (CDuongTron);
17.    int operator >= (CDuongTron);
18.    int operator < (CDuongTron);
19.    int operator <= (CDuongTron);
20.    ~CDuongTron();
21.};
```

Thiết kế lớp đối tượng CDuongTron



— Thiết kế các phương thức xử lý.

```
1. float CDuongTron::ChuVi()  
2. {  
3. |   return (float)pi * 2 * R;  
4. }
```

Thiết kế lớp đối tượng CDuongTron



— Thiết kế các phương thức xử lý.

```
11.float CDuongTron::DienTich()  
12.{  
13.|    return (float)pi * R * R;  
14.}
```


Thiết kế lớp đối tượng CDuongTron



— Thiết kế các phương thức xử lý.

```
11.int CDuongTron::operator == (CDuongTron c)
12.{
13.    if (c.I.getX() == I.getX() && c.I.getY() == I.getY() &&
        c.R == R)
14.        return 1;
15.    return 0;
16.}
```

Thiết kế lớp đối tượng CDuongTron



— Thiết kế các phương thức xử lý.

```
11.int CDuongTron::operator != (CDuongTron c)
12.{
13.    if (c.I.getX() == I.getX() && c.I.getY() == I.getY() &&
        c.R == R)
14.        return 0;
15.    return 1;
16.}
```

Thiết kế lớp đối tượng CDuongTron



— Thiết kế các phương thức xử lý.

```
11.int CDuongTron::operator > (CDuongTron c)
12.{
13.    if (R > c.R)
14.        return 1;
15.    return 0;
16.}
```

Thiết kế lớp đối tượng CDuongTron



— Thiết kế các phương thức xử lý.

```
11.int CDuongTron::operator >= (CDuongTron c)
12.{
13.    if (R >= c.R)
14.        return 1;
15.    return 0;
16.}
```

Thiết kế lớp đối tượng CDuongTron



— Thiết kế các phương thức xử lý.

```
11.int CDuongTron::operator < (CDuongTron c)
12.{
13.    if (R < c.R)
14.        return 1;
15.    return 0;
16.}
```

Thiết kế lớp đối tượng CDuongTron



— Thiết kế các phương thức xử lý.

```
11.int CDuongTron::operator <= (CDuongTron c)
12.{
13.    if (R <= c.R)
14.        return 1;
15.    return 0;
16.}
```

Thiết kế lớp đối tượng CDuongTron



— Thiết kế các phương thức xử lý.

```
11. CDuongTron::~CDuongTron()
```

```
12. {
```

```
13. |     return;
```

```
14. }
```



Cảm ơn quý vị đã lắng nghe

Nhóm tác giả

TS. Nguyễn Tấn Trần Minh Khang