Thiết kế lớp hình cầu trong mặt phẳng Oxyz

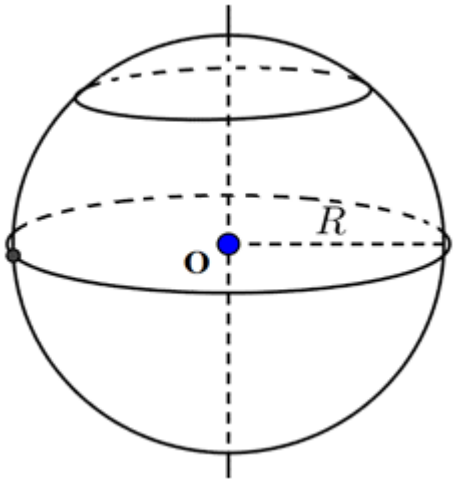
1. Hồ Thái Ngọc
2. ThS. Võ Duy Nguyên
3. TS. Nguyễn Tấn Trần Minh Khang

Thiết kế lớp đối tượng điểm CHinhCau

– Thuộc tính

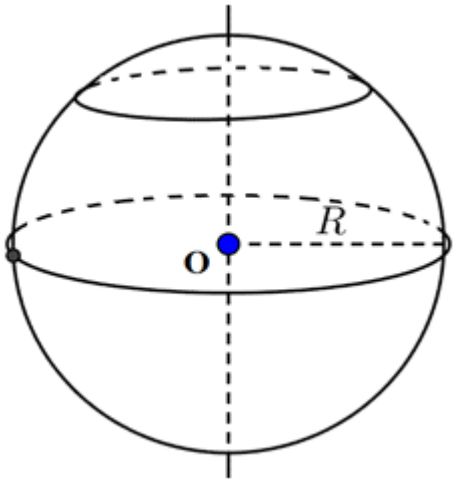
+ Tâm hình cầu.

+ Bán kính.



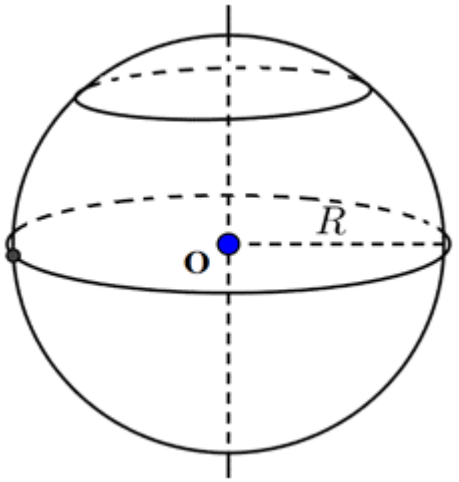
Thiết kế lớp đối tượng điểm CHinhCau

```
11.class CHinhCau
12.{
13.|    private:
14.|        CKG I;
15.|        float R;
16.|    public:
```



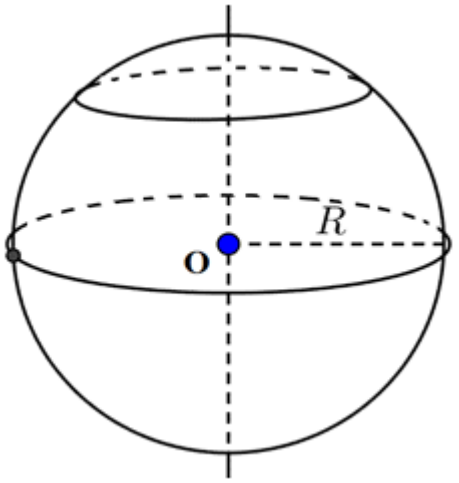
Thiết kế lớp đối tượng điểm CHinhCau

- Thuộc tính
  + Tâm hình cầu.
  + 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 điểm CHinhCau

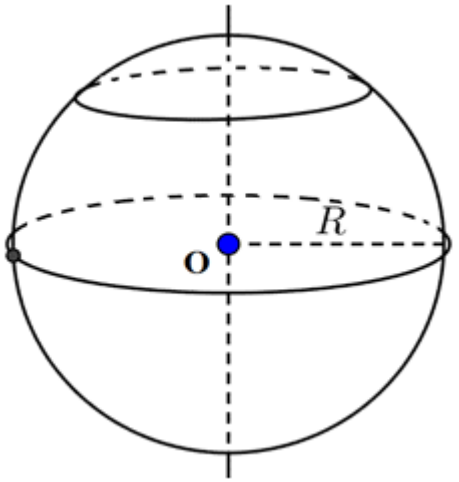
```
11.class CHinhCau
12.{
13.    private:
14.        CDiemKG I;
15.        float R;
16.    public:
17.        // Nhóm phương thức khởi tạo
18.        // Nhóm phương thức cung cấp thông tin
19.        // Nhóm phương thức cập nhật thông tin
20.        // Nhóm phương thức kiểm tra
21.        // Nhóm phương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



Lớp đối tượng điểm CHinhCau

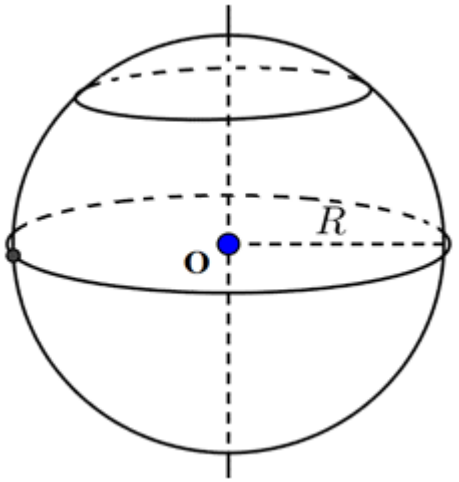
– Nhóm phương thức khởi tạo

+ Phương thức khởi tạo mặc định.
+ Phương thức khởi tạo sao chép.
+ Phương thức khởi tạo khi biết đầy đủ thông tir
+ 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 điểm CHinhCau

```
17.         // Nhóm phương thức khởi tạo
18.         void KhoiTao();
19.         void KhoiTao(CDiemKG, float);
20.         void KhoiTao(const CHinhCau&);
21.         CHinhCau();
22.         CHinhCau(CDiemKG, float);
23.         CHinhCau(const CHinhCau&);
24.         friend ostream& operator>>(ostream&,
    CHinhCau&);
25.         void Nhap();
```



Lớp đối tượng điểm CHinhCau

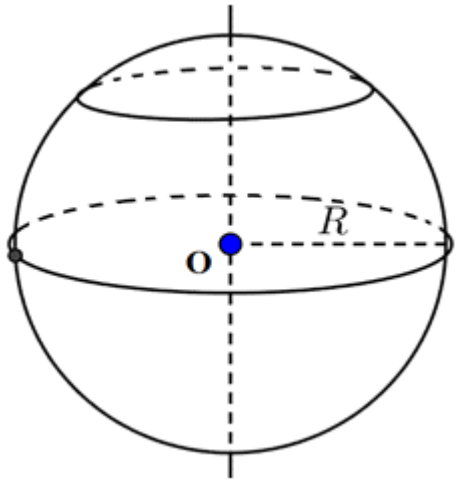
– Nhóm phương thức cung cấp thông tin

+ Phương thức Xuất.

+ Toán tử ra.

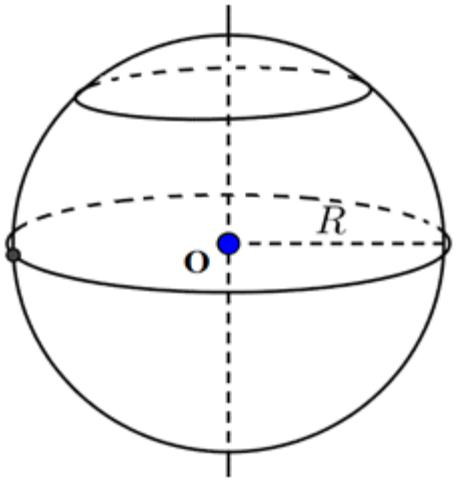
+ Phương thức cung cấp tâm hình cầu.

+ Phương thức cung cấp bán kính.



Thiết kế lớp đối tượng điểm CHinhCau

```
26.         // Nhóm phương thức cung cấp thông tin
27.         void Xuat();
28.         friend ostream& operator << (ostream&,
    CHinhCau&);
29.         CDiemKG getI();
30.         float getR();
```



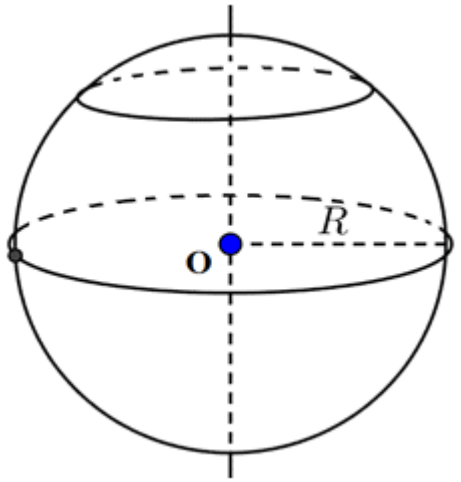
Lớp đối tượng điểm CHinhCau

– Nhóm phương thức cập nhật thông tin

+ Toán tử gán.

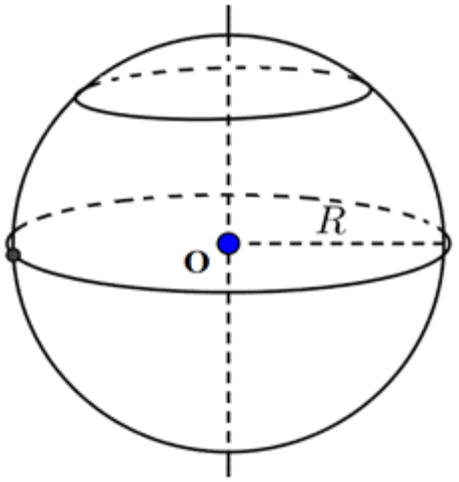
+ Phương thức cập nhật tâm hình cầu.

+ Phương thức cập nhật bán kính.



Thiết kế lớp đối tượng điểm CHinhCau

```
31.         // Nhóm phương thức cập nhật thông tin
32.         CHinhCau& operator = (const CHinhCau&);
33.         void setI(CDiemKG);
34.         void setR(float);
```



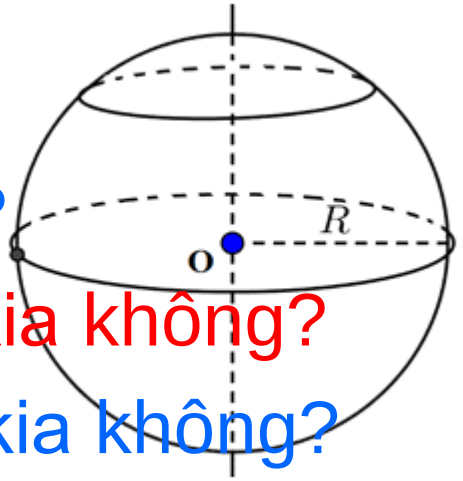
Lớp đối tượng điểm CHinhCau

– Nhóm phương thức kiểm tra

+ Kiểm tra hai hình cầu có trùng nhau không?

+ Kiểm tra hai hình cầu có tiếp xúc trong không?

+ Kiểm tra hai hình cầu có tiếp xúc ngoài không?



+ Kiểm tra hình cầu này có nằm trong hình cầu kia không?

+ Kiểm tra hình cầu này có nằm ngoài hình cầu kia không?

+ Kiểm tra hai hình cầu có giao nhau không?

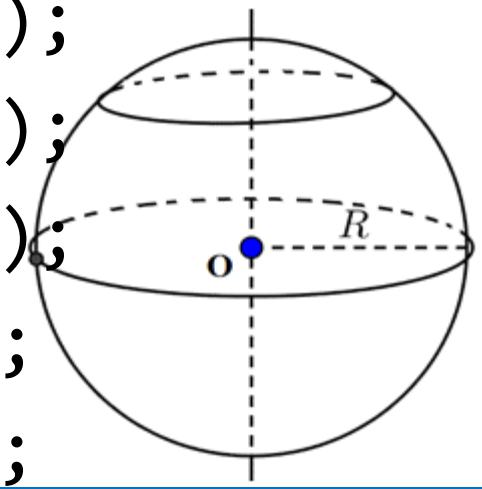
+ Kiểm tra điểm có ngoài trong hình cầu không?

+ Kiểm tra điểm có nằm ngoài hình cầu không?

+ Kiểm tra điểm có thuộc hình cầu không?

Thiết kế lớp đối tượng điểm CHinhCau

```
35.         // Nhóm phương thức kiểm tra
36.         int isTrungNhau(const CHinhCau&);
37.         int isTiepXucTrong(const CHinhCau&);
38.         int isTiepXucNgoai(const CHinhCau&);
39.         int isNamTrong(const CHinhCau&);
40.         int isNamNgoai(const CHinhCau&);
41.         int isGiaoNhau(const CHinhCau&);
42.         int isNamTrong(const CDiemKG&);
43.         int isNamNgoai(const CDiemKG&);
44.         int isThuoc(const CDiemKG&);
```



Lớp đối tượng điểm CHinhCau

– Nhóm phương thức xử lý

+ Toán tử so sánh bằng

+ Toán tử so sánh khác

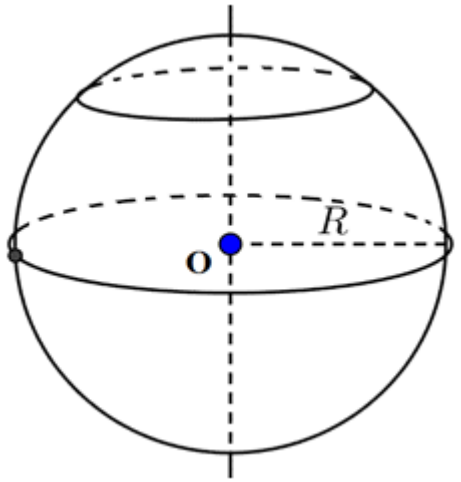
+ Toán tử so sánh lớn hơn

+ Toán tử so sánh nhỏ hơn

+ Toán tử so sánh lớn hơn bằng

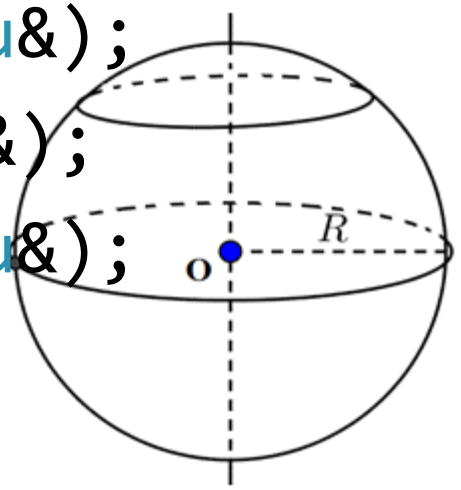
+ Toán tử so sánh nhỏ hơn bằng

+ Tiêu chuẩn so sánh dựa vào bán kính hình cầu (Bán kính lớn hơn thì lớn hơn)



Thiết kế lớp đối tượng điểm CHinhCau

```
42.         // Nhóm phương thức xử lý
43.         int operator == (const CHinhCau&);
44.         int operator != (const CHinhCau&);
45.         int operator > (const CHinhCau&);
46.         int operator >= (const CHinhCau&);
47.         int operator < (const CHinhCau&);
48.         int operator <= (const CHinhCau&);
```

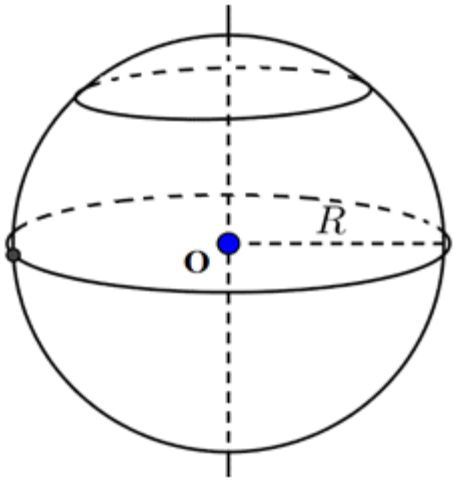


Lớp đối tượng điểm CHinhCau

– Nhóm phương thức xử lý

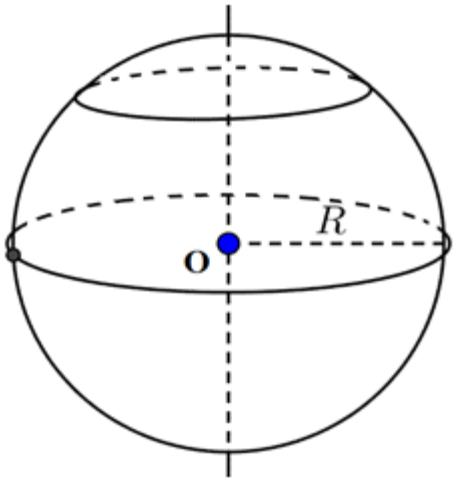
+ Tính diện tích mặt cầu.

+ Tính thể tích mặt cầu.



Thiết kế lớp đối tượng điểm CHinhCau

```
49.         // Nhóm phương thức xử lý
50.         float DienTichMatCau();
51.         float TheTichHinhCau();
```



Thiết kế lớp hình cầu

- Định nghĩa các phương thức cung cấp thông tin.
- Cách 01.

```
11.CDiemKG CHinhCau::GetI()
12.{
13.|    return I;
14.}
```

- Cách 02.

```
11.CDiemKG CHinhCau::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 hình cầu

- Định nghĩa các phương thức cung cấp thông tin.
- Cách 01.

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

- Cách 02.

```
11.float CHinhCau::getR()
12.{
13.|    return R;
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 hình cầu

- Định nghĩa các phương thức cập nhật thông tin.

```
11.void CHinhCau::setI(CDiemKG II)
12.{
13.    I.setx(II.getx());
14.    I.sety(II.gety());
15.    I.setz(II.getz());
16.}

11.void CHinhCau::setR(float RR)
12.{
13.    R = RR;
14.}
```

Thiết kế lớp hình cầu

– Định nghĩa các phương thức kiểm tra.

```
11.int CHinhCau::isTrungNhau(const CHinhCau& PP)
12.{
13.    if (I == PP.I && R == PP.R)
14.        return 1;
15.    return 0;
16.}
```

Thiết kế lớp hình cầu

— Định nghĩa các phương thức kiểm tra.

```
11.int CHinhCau::isTiepXucTrong(const CHinhCau& PP)
12.{
13.    float distance = I.KhoangCach(PP.I) + R;
14.    if (distance == PP.R)
15.        return 1;
16.    return 0;
17.}
```

Thiết kế lớp hình cầu

- Định nghĩa các phương thức kiểm tra.

```
11.int CHinhCau::isTiepXucNgoai(const CHinhCau& PP)
12.{
13.    float distance = I.KhoangCach(PP.I);
14.    if (distance == R + PP.R)
15.        return 1;
16.    return 0;
17.}
```

Thiết kế lớp hình cầu

- Định nghĩa các phương thức kiểm tra.

```
11.int CHinhCau::isNamTrong(const CHinhCau& PP)
12.{
13.    float distance = I.KhoangCach(PP.I) + R;
14.    if (distance < 2 * PP.R)
15.        return 1;
16.    return 0;
17.}
```

Thiết kế lớp hình cầu

- Định nghĩa các phương thức kiểm tra.

```
11.int CHinhCau::isNamNgoai(const CHinhCau& PP)
12.{
13.    float distance = I.KhoangCach(PP.I) + R;
14.    if (distance > 2 * PP.R)
15.        return 1;
16.    return 0;
17.}
```

Thiết kế lớp hình cầu

- Định nghĩa các phương thức kiểm tra.

```
11.int CHinhCau::isGiaoNhau(const CHinhCau& PP)
12.{
13.    if (I.KhoangCach(PP.I) < (R + PP.R))
14.        return 1;
15.    return 0;
16.}
```

Thiết kế lớp hình cầu

- Định nghĩa các phương thức khởi tạo.

```
11.void CHinhCau::KhoiTao()
12.{
13.    I.KhoiTaoCDiemKG();
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 gọi phương thức khởi tạo mặc định, bán kinh lấy giá trị 0.

Thiết kế lớp hình cầu

- Định nghĩa các phương thức khởi tạo.

```
11.void CHinhCau::KhoiTao(CDiemKG II, float RR)
12.{
13.    I = II;
14.    R = RR;
15.}
```

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

Thiết kế lớp hình cầu

- Định nghĩa các phương thức khởi tạo.

```
11.void CHinhCau::KhoiTao(const CHinhCau& PP)
12.{
13.    I = PP.I;
14.    R = PP.R;
15.}
```

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

Thiết kế lớp hình cầu

- Định nghĩa các phương thức khởi tạo.

```
11.CHinhCau::CHinhCau()
12.{
13.|    R = 0;
14.}
```

- 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`) thiết lập mặc định, bán kính (`R`) lấy giá trị 0.

Thiết kế lớp hình cầu

- Định nghĩa các phương thức khởi tạo.

```
11.CHinhCau::CHinhCau(CDiemKG P, float RR)
12.{
13.    I = P;
14.    R = RR;
15.}
```

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

Thiết kế lớp hình cầu

- Định nghĩa các phương thức khởi tạo.

```
11.CHinhCau::CHinhCau(const CHinhCau& OO)
12.{
13.    I = OO.I;
14.    R = OO.R;
15.}
```

- Phương thức thiết lập sao chép, nhận một tham số đầu vào là `x` là đối tượng thuộc lớp `CHinhCau`, các thông tin ban đầu của đối tượng được thiết lập như sau: tâm (`I`) lấy giá trị `OO.I`, bán kính (`R`) lấy giá trị `OO.R`.

Thiết kế lớp hình cầu

- Định nghĩa các phương thức khởi tạo.

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

Thiết kế lớp hình cầu

- Định nghĩa các phương thức khởi tạo.

```
11.ostream& operator << (ostream& os, CHinhCau& OO)
12.{
13.    os << OO.I;
14.    cout << "Ban kinh duong tron la: ";
15.    os << OO.R;
16.    return os;
17.}
```

Thiết kế lớp hình cầu

– Định nghĩa các phương thức xử lý.

```
11.int CHinhCau::operator > (const CHinhCau& PP)
12.{
13.    if (R > PP.R)
14.        return 1;
15.    return 0;
16.}
```

Thiết kế lớp hình cầu

- Định nghĩa các phương thức xử lý.

```
11.int CHinhCau::operator < (const CHinhCau& PP)
12.{
13.    if (R < PP.R)
14.        return 1;
15.    return 0;
16.}
```

Thiết kế lớp hình cầu

— Định nghĩa các phương thức xử lý.

```
11.int CHinhCau::operator == (const CHinhCau& PP)
12.{
13.    if (R == PP.R)
14.        return 1;
15.    return 0;
16.}
```

Thiết kế lớp hình cầu

- Định nghĩa các phương thức xử lý.

```
11.int CHinhCau::operator != (const CHinhCau& PP)
12.{
13.    if (R != PP.R)
14.        return 1;
15.    return 0;
16.}
```

Thiết kế lớp hình cầu

- Định nghĩa các phương thức xử lý.

```
11.int CHinhCau::operator >= (const CHinhCau& PP)
12.{
13.    if (R >= PP.R)
14.        return 1;
15.    return 0;
16.}
```

Thiết kế lớp hình cầu

- Định nghĩa các phương thức xử lý.

```
11.int CHinhCau::operator <= (const CHinhCau& PP)
12.{
13.    if (R <= PP.R)
14.        return 1;
15.    return 0;
16.}
```

Thiết kế lớp hình cầu

- Định nghĩa các phương thức xử lý.

```
11.CHinhCau& CHinhCau::operator = (CHinhCau b)
12.{
13.    I = b.I;
14.    R = b.R;
15.    return *this;
16.}
```

Thiết kế lớp hình cầu

– Định nghĩa các phương thức xử lý.

```
11.float CHinhCau::DienTichMatCau()
12.{
13.    return (4 * 3.14 * R * R);
14.}
```

Thiết kế lớp hình cầu

– Định nghĩa các phương thức xử lý.

```
11.float CHinhCau::TheTichHinhCau()
12.{
13.    return (4 * 3.14 * R * R * R / 3);
14.}
```

Cảm ơn quí vị đã lắng nghe

Nhóm tác giả

Hồ Thái Ngọc

ThS. Võ Duy Nguyên

TS. Nguyễn Tấn Trần Minh Khang