TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM

KHOA TOÁN – TIN HỌC

----ꕥ🕮ꕥ----

BÁO CÁO BÀI TẬP LỚN

MÔN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Thành viên nhóm:

VŨ QUANG MINH – 18110150

NGUYỄN PHÚC KHANG – 18110113

Bài tập đã chọn: 3

Lớp: 18TTH1

Giáo viên hướng dẫn: ***NGUYỄN NGỌC LONG***

Mục lục

I)Chủ đề báo cáo

II)Các đối tượng trong bài tập

1)Sơ đồ các đối tượng trong bài tập

2)Lớp Point

3)Lớp CShape

4)Lớp Ellipse

5)Lớp hình tròn

6)Lớp hình chữ nhật

7)Lớp hình vuông

8)Lớp Tam giác

9)Lớp Đa Giác

III)Cách thức hoạt động của các đối tượng trên Window Application

1)Chế độ Radio Button

2)Các thao tác di chuyển, phóng to, thu nhỏ

3)Các thao tác vẽ, tô màu , tô màu viền, tô màu phần giao.

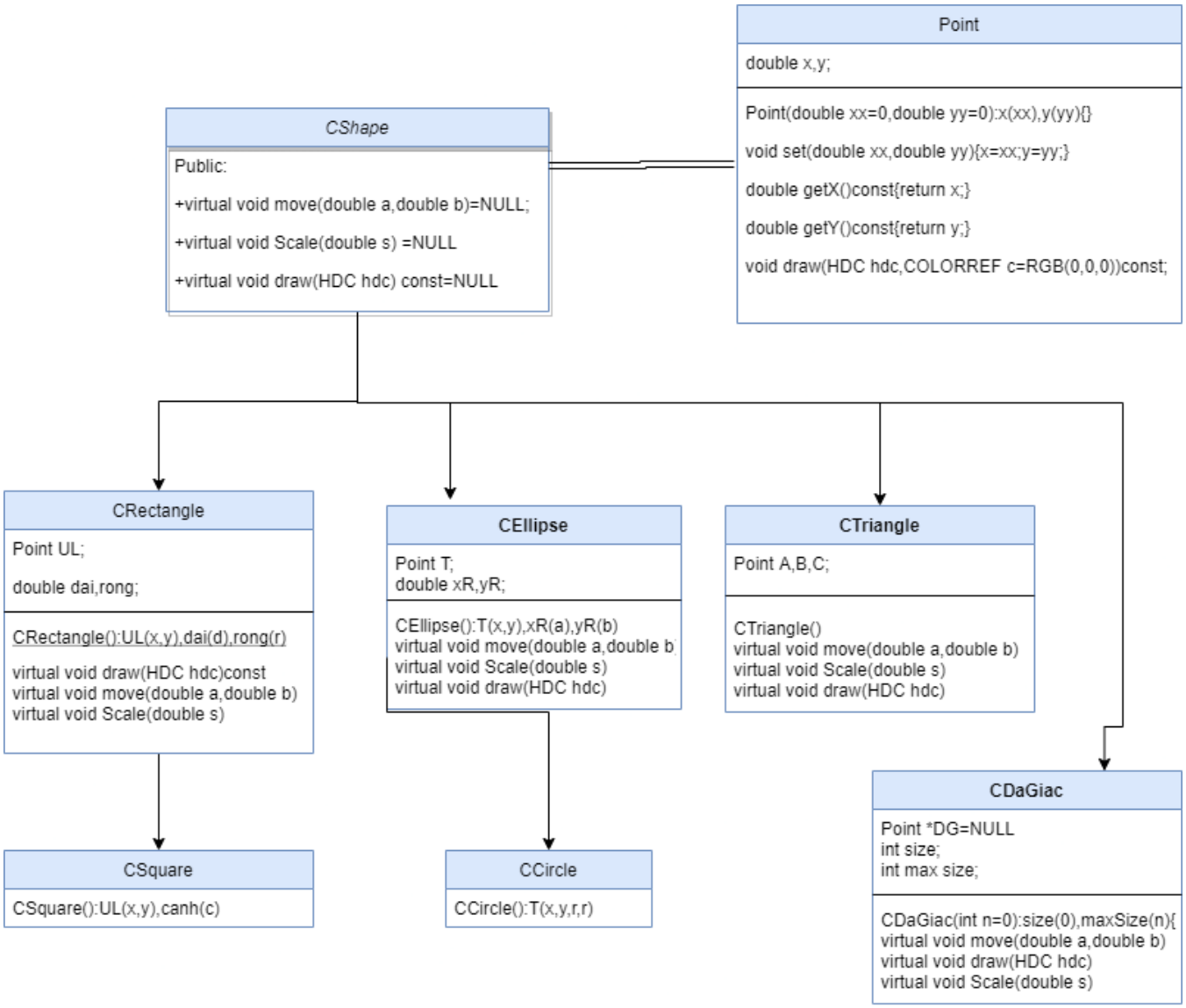
I) Chủ đề báo cáo:

Bài 3 : Cho các loại đối tượng: hình tròn, hình ellipse, đa giác, hình chữ nhật, hình vuông, hình tam giác. Viết chương trình ứng dụng cho phép tạo (hoặc nhập) hai hình thuộc một trong các hình kể trên. In thông báo cho biết hai hình có giao nhau không, nếu có tô màu phần giao và tô đậm đường biên của phần giao. Người sử dụng có thể bấm các phím mũi tên để di chuyển một trong hai hình, phím +, -, để phóng to thu nhỏ một trong hai hình. Có thể lập trình trong môi trường Windows.

Trong bài tập này đã làm cho chúng ta hiểu rõ các thức vẽ hình cách thức di chuyển, thao tác phóng to, thu nhỏ, tô màu, tô màu viền, đổi màu 2 phần giao nhau, in ra thông báo 2 phần giao nhau. Ngoài ra chúng ta sẽ còn biết thêm về chế độ Radio Buton.

II) Các đối tượng trong bài tập

1) Sơ đồ các đối tượng trong bài tập



2) Đối tượng Point

Đối tượng Point gồm *Point.h* và Point.cpp

Trong đối tượng Point có các phần *set(đọc giá trị x, đọc giá trị y)*, *getX()* trả về biến x, *getY()* trả về biến y, move để di chuyển điểm, *draw(để vẽ).*

Trong lớp *Point.h* có

```
class Point
{
private:
            double x, y;
public:
            Point(double xx=0,double yy=0):x(xx),y(yy){}
            void set(double xx, double yy) { x = xx; y = yy; }
            double getX()const { return x; }
            double getY()const { return y; }
            void move(double dx, double dy) { x += dx; y += dy; }
            void draw(HDC hdc, COLORREF c = RGB(0, 0, 0))const;
};
```

Trong phần Point.cpp

```
void Point::draw(HDC hdc, COLORREF c)const
      {
            SetPixel(hdc, int(x), int(y), c);
      }
```

3) Đối tượng CShape

Trong bài tập chúng ta sẽ dùng CShape mượn thư viện từ class Point làm đa hình giải quyết các hình nói chung. Bắt đầu là giải quyết các hình ellipse, hình chữ nhật, hình đa giác và hình tròn. Rồi từ lớp ellipse sẽ xây dựng lớp hình tròn kế thừa lớp ellipse và hình vuông kế thừa từ lớp hình chữ nhật.

Trong đối tượng CShape chúng ta sẽ dùng phương thức ảo để vẽ, phóng to, thu nhỏ và di chuyển,..

Sau đây là lớp CShape:

```
class CShape
{
public:
virtual void draw(HDC hdc) const = NULL;
virtual void move(double a, double b) = NULL;
virtual void Scale(double s) = NULL;
//virtual bool isInside() const = NULL;
};
```

4) Đối tượng Hình Ellipse

Trong bài tập đối tượng hình Ellipse sẽ được gọi là CEllipse phần CEllipse sẽ được lấy Public từ lớp CShape và mượn thư viện của hàm *Point.h*. Còn phần Private sẽ lấy 1 điểm T là tâm đối xứng 2 trục của hình ellipse và lấy bán kính 2 trục là trục lớn và trục nhỏ. Để phóng to hình ellipse chúng ta sẽ phóng to trục lớn trục nhỏ.

Sau đây là lớp Ellipse:

```
class CEllipse :public CShape
{
     Point T;
     double xR, yR;
     public:
     CEllipse(double x, double y, double a, double b) : T(x, y), xR(a), yR(b)
{}
virtual void move(double a, double b) { T.move(a, b);  }
virtual void Scale(double s) { xR *= s; yR *= s; }
virtual  void  draw(HDC  hdc)  const  {  Ellipse(hdc,  T.getX()-xR,  T.getY()-yR,
T.getX() + xR, T.getY() + yR); }
};
```

5) Đối tượng Hình Tròn

Trong bài tập đối tượng hình tròn có tên class là Ccircle được kế thừa từ class CEllipse và dùng thư viện *Point.h*. Lớp hình tròn được xác định bởi tâm và bán kính hình tròn.

Sau đây là lớp hình tròn:

```
class CCircle : public CEllipse
{
public:
	CCircle(double xT = 300, double yT = 400, double r = 150) :
	CEllipse(xT, yT, r, r) {}
};
```

6) Đối tượng Hình Chữ Nhật

Trong bài tập đối tượng lớp Hình Chữ Nhật có tên class là CRectangle được kế thừa từ class CShape . Lớp hình chữ nhật được xác định bởi một điểm *trên bên trái* (UL)của hình chữ nhật và độ dài 2 cạnh của HCN. Để phóng to chúng ta sẽ lấy 2 cạnh nhân hệ số s, rồi lấy điểm phía trên bên trái dịch chuyển sang 1 đoạn là *rong * (1 - s) / 2, dai * (1 - s) / 2).*

Sau đây là lớp hình chữ nhật:

```
class CRectangle :public CShape
{
	Point UL;
	double dai, rong;
public:
	CRectangle(double x,double y,double d,double r):UL(x,y),dai(d),rong(r){}
	virtual void draw(HDC hdc) const { Rectangle(hdc,UL.getX(), UL.getY(),
			UL.getX()+dai, UL.getY()+rong);}
	virtual void move(double a, double b) { UL.move(a, b); }
	virtual void Scale(double s) { rong *= s; dai *= s;
	UL.move(rong * (1 - s) / 2, dai * (1 - s) / 2); }

};
```

7) Lớp Hình Vuông

Lớp hình vuông có tên class là CSquare, được kế thừa từ lớp Hình chữ nhật. CSquare được xây dựng trên thư viện *Point.h* và *Rectangle.h*.

Sau đây là lớp Hình Vuông:

```
class CSquare :public CRectangle
{
public:
	CSquare(double x, double y, double c) :CRectangle(x, y, c, c) {};
};
```

8) Lớp Tam Giác

Lớp tam giác có tên class là CTriangle, do tam giác chỉ có quy luật chung là 3 điểm nên lớp của nó phải được xây dựng trên tọa độ 3 điểm A, B, C. Lớp tam giác được xây dựng trên thư viện *Point.h*. Trong hàm draw của tam giác ta phải dùng hàm Polygon để vẽ 3 điểm của tam giác. Để phóng to chúng ta sẽ phóng lần lượt 3 điểm của tam giác.

Sau đây là lớp tam giác:

```
class CTriangle:public CShape
{
private:
Point A, B, C;
public:
CTriangle(double xA, double yA, double xB,
	double yB, double xC, double yC) : A(xA, yA), B(xB, yB), C(xC, yC){}
virtual void draw(HDC hdc) const {
	POINT apt[] = { A.getX(),A.getY(),B.getX(),B.getY(),C.getX(),C.getY() };
		Polygon(hdc, apt, 3);
	}
virtual void move(double a, double b) {
		A.move(a, b);
		B.move(a, b);
		C.move(a, b);
}
virtual void Scale(double s)
{
	A.set(A.getX() * s, A.getY() * s);
```

```
		B.set(B.getX() * s, B.getY() * s);
		C.set(C.getX() * s, C.getY() * s);
	}
	};
```

9) Lớp Đa Giác

Lớp đa giác có tên class CDaGiac. Lớp đa giác là 1 lớp đặc biệt nhất trong 6 hình trong danh sách. Xây dựng nó cũng khá đặc biệt. Do không biết được kích thước thật của 1 đa giác. Nên lúc đầu do không biết được kích thước của đa giác là bao nhiêu, nên lúc đầu sẽ cho size=0, sau đó sẽ cho thêm 1 biến Max_size. Khi thao tác thêm kích thước cho Đa giác thì sẽ sài hàm addPoint để nhập điểm cho đa giác sao cho số điểm < Max_side.

Sau đây là lớp Đa Giác:

```
	class CDaGiac:public CShape
	{
	protected:
		Point* DG = NULL;
		int size;
		int maxSize;
	public:
		CDaGiac(int n=0):size(0),maxSize(n){
			DG = new Point[maxSize];
		}
		virtual void move(double a, double b) {
		for (int i = 0; i < size; i++)
			{
				DG[i].move(a, b);
			}
		}
		virtual void draw(HDC hdc)const;
		virtual void Scale(double s);
		void addPoint(Point T);
```

```
};
```

Sau đây là *DaGiac.cpp*:

```
void CDaGiac::Scale(double s){
	for (int i = 0; i < size;i++){
		DG[i].set((DG[i].getX()) * s, (DG[i].getY()) * s);
	}
}
void CDaGiac::draw(HDC hdc)const{
	POINT* apt = new POINT[size+1];
	for (int i = 0; i < size; i++) {
		apt[i].x = DG[i].getX();
		apt[i].y = DG[i].getY();
	}
	Polygon(hdc, apt, size);
	//*/
}


void CDaGiac::addPoint(Point T) {
	DG[size] = T;
	size++;
}
```

Để vẽ hình Đa Giác trên Window Application chúng ta sẽ nhập sẵn các điểm. Rồi chúng ta sẽ dùng ép kiểu *dynamic_cast* để cho Đa Giác vào lớp CShape xong rồi chúng ta sẽ addpoint các điểm đã nhập.

```
Point A(100, 200);
Point B(300, 200);
Point C(400, 300);
Point D(50, 300);
Point E(60, 250);
```

```
	dynamic_cast<CDaGiac*>(aS[g])->addPoint(A);
	dynamic_cast<CDaGiac*>(aS[g])->addPoint(B);
	dynamic_cast<CDaGiac*>(aS[g])->addPoint(C);
	dynamic_cast<CDaGiac*>(aS[g])->addPoint(D);
	dynamic_cast<CDaGiac*>(aS[g])->addPoint(E);
```

III) Cách thức hoạt động của các đối tượng trên Window Application

1) Chế độ Radio Button

a) Phần Khai Báo

Trong bài này chúng ta sẽ dùng chế độ Radio Button để chọn đối tượng tùy thích, nhược điểm của Radio Button là chỉ được chọn một đối tượng trong danh sách. Nên chúng ta sẽ dùng 2 bảng Radio Button để chọn 2 hình thỏa mãn yêu cầu đề bài.

Do chọn 2 bảng Trong Radio Button nên chúng ta sẽ *dùng 2 con trỏ*.

Trong phần khai báo chúng ta sẽ tạo 2 đối tượng *H_Ellipse, H_Circle, H_Rectangle, H_Square, H_Triangle, H_Polygon* là đối tượng 1, còn *H1_Ellipse, H1_Circle, H1_Rectangle, H1_Square, H1_Triangle, H1_Polygon* là dối tượng 2.

Sau đây là phần khai báo:

```
	enum BUTTON_ID { H_Ellipse, H_Circle, H_Rectangle, H_Square, H_Triangle, H_Polygon,
	    H1_Ellipse, H1_Circle, H1_Rectangle, H1_Square, H1_Triangle, H1_Polygon
	};
	struct ButtonMap
	{
	    BUTTON_ID id;
	    const TCHAR *name;
	};
	static ButtonMap aMapTab[] =
	{
	    H_Ellipse, L"Ellipse",
```

```
        H_Circle, L"Circle",
        H_Rectangle, L"Rectangle",
        H_Square, L"Square",
        H_Triangle,L"Triangle",
        H_Polygon,L"Polygon",
        H1_Ellipse, L"Ellipse",
        H1_Circle, L"Circle",
        H1_Rectangle, L"Rectangle",
        H1_Square, L"Square",
        H1_Triangle,L"Triangle",
        H1_Polygon,L"Polygon"
    };
    #define dim(a)(sizeof(a)/size(a[0]))
```

Trong phần *LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)*

```
    static CShape* aS[] = {
            new CEllipse(400,250,200,100),
            new CCircle(300, 300, 150),
            new CRectangle(200,200,300,200),
            new CSquare(100,100,200),
            new CTriangle(100,200,200,400,300,200),
            new CDaGiac(4),
        };
        static int h = 0;
        static int h1 = 0;
        static int temp = h;
        const int n = sizeof(aS) / sizeof(aS[0]);
        int g = 5;
        static CShape* pS = aS[h];
```

```
        static CShape* pS1 = aS[h1];

        static BUTTON_ID iFigureRadio = H_Ellipse,iFigureRadio1=H_Ellipse;

        RECT rt;
```

Chúng ta đã có 2 con trỏ con trỏ pS sẽ trỏ đến các đối tượng 1 là*H_Ellipse,H_Circle,H_Rectangle,H_Square,H_Triangle,H_Polygon(đa giác)* còn con trỏ pS1 trỏ tới các đối tượng 2 là *H1_Ellipse, H1_Circle,H1_Rectangle,H1_Square,H1_Triangle,H1_Polygon(đa giác).*

b)Trong phần WM_CREATE

Trong phần *WM_CREATE* chúng ta sẽ tạo 2 bảng để thao tác chế độ Radio Button

Tạo bảng cho đối tượng 1:

```
    CreateWindow(TEXT("button"),TEXT("Ellipse"),WS_VISIBLE|WS_CHILD|BS_AUTORADIOBU
TTON, 20, 10, 100, 30,hWnd, (HMENU)aMapTab[0].id,hInst, NULL);

    CreateWindow(TEXT("button"),   TEXT("Circle"),   WS_VISIBLE   |   WS_CHILD   |
BS_AUTORADIOBUTTON, 20, 35, 100, 30, hWnd, (HMENU)aMapTab[1].id,hInst, NULL);

    CreateWindow(TEXT("button"),  TEXT("Rectangle"),  WS_VISIBLE   |   WS_CHILD   |
BS_AUTORADIOBUTTON, 20, 60, 100, 30, hWnd, (HMENU)aMapTab[2].id,hInst, NULL);

    CreateWindow(TEXT("button"),   TEXT("Square"),   WS_VISIBLE   |   WS_CHILD   |
BS_AUTORADIOBUTTON, 20, 85, 100, 30, hWnd, (HMENU)aMapTab[3].id,hInst, NULL);

    CreateWindow(TEXT("button"),   TEXT("Triangle"),   WS_VISIBLE   |   WS_CHILD   |
BS_AUTORADIOBUTTON, 20, 110, 100, 30, hWnd, (HMENU)aMapTab[4].id,hInst, NULL);

    CreateWindow(TEXT("button"),   TEXT("Polygon"),   WS_VISIBLE   |   WS_CHILD   |
BS_AUTORADIOBUTTON, 20, 135, 100, 30, hWnd, (HMENU)aMapTab[5].id,hInst, NULL);
```

Tạo bảng cho đối tượng 2:

```
    CreateWindow(TEXT("button"),   TEXT("Ellipse"),   WS_VISIBLE   |   WS_CHILD   |
BS_AUTORADIOBUTTON, 120, 10, 100, 30, hWnd, (HMENU)aMapTab[6].id, hInst, NULL);

    CreateWindow(TEXT("button"),   TEXT("Circle"),   WS_VISIBLE   |   WS_CHILD   |
BS_AUTORADIOBUTTON, 120, 35, 100, 30, hWnd, (HMENU)aMapTab[7].id, hInst, NULL);

    CreateWindow(TEXT("button"),  TEXT("Rectangle"),  WS_VISIBLE   |   WS_CHILD   |
BS_AUTORADIOBUTTON, 120, 60, 100, 30, hWnd, (HMENU)aMapTab[8].id, hInst, NULL);

    CreateWindow(TEXT("button"),   TEXT("Square"),   WS_VISIBLE   |   WS_CHILD   |
BS_AUTORADIOBUTTON, 120, 85, 100, 30, hWnd, (HMENU)aMapTab[9].id, hInst, NULL);

    CreateWindow(TEXT("button"),   TEXT("Triangle"),   WS_VISIBLE   |   WS_CHILD   |
BS_AUTORADIOBUTTON, 120, 110, 100, 30, hWnd, (HMENU)aMapTab[10].id, hInst, NULL);
```

```
      CreateWindow(TEXT("button"),   TEXT("Polygon"),   WS_VISIBLE   |   WS_CHILD   |
BS_AUTORADIOBUTTON, 120, 135, 100, 30, hWnd, (HMENU)aMapTab[11].id, hInst, NULL);

      GetClientRect(hWnd, &rt);
```

c)Trong phần WM_COMMAND

Nếu như phần *WM_CREATE* là front end thì phần WM_COMMAND là front end. Phần *WM_COMMAND* sẽ giúp 2 con trỏ, trỏ đến đúng đối tượng mà chúng ta cần trỏ tới.

Sau đây là phần *WM_COMMAND* đối với đối tượng 1:

```
Case  H_Ellipse:case  H_Circle:case  H_Rectangle:case  H_Square:case  H_Triangle:case
H_Polygon:

      if (iFigureRadio != wmId)

         {

           iFigureRadio = (BUTTON_ID)wmId;

           InvalidateRect(hWnd, NULL, TRUE);

           CheckRadioButton(hWnd, H_Circle, H_Polygon, iFigureRadio);

           h = iFigureRadio - H_Ellipse;

           pS = aS[h];

           InvalidateRect(hWnd, NULL, TRUE);

         }

      Break;
```

Còn đây đối với đối tượng 2

```
case H1_Ellipse:case H1_Circle:case H1_Rectangle:case H1_Square:case H1_Triangle:case
H1_Polygon:

      if (iFigureRadio1 != wmId)

          {

              iFigureRadio1 = (BUTTON_ID)wmId;

              InvalidateRect(hWnd, NULL, TRUE);

              CheckRadioButton(hWnd, H1_Circle, H1_Polygon, iFigureRadio1);

              h1 = iFigureRadio1 - H1_Ellipse;

              pS1 = aS[h1];

              InvalidateRect(hWnd, NULL, TRUE);
```

```
	}
Break;
```

2) Các thao tác di chuyển, phóng to, thu nhỏ

Các thao tác di chuyển của đối tượng 1 sẽ năm trong phần *WM_KEYDOWN*, các thao tác di chuyển đối tượng 2 sẽ nằm trong *WM_CHAR*.

Các thao tác phóng to, thu nhỏ cũng nằm trong *WM_CHAR*.

Đối với đối tượng 1:

Để di chuyển đối tượng 1 sẽ dùng các phím trên bàn phím là ↑,↓,←,→ lần lượt để di chuyển lên trên, xuống dưới, qua trái, qua phải.

Các phím +, - để phóng to, thu nhỏ.

Đối với đối tượng 2:

Để di chuyển dối tượng 2 sẽ dùng các phím trên bàn phím là 8,2,4,6 lần lượt để di chuyển lên trên, xuống dưới, qua trái, qua phải.

Các phím *,/ để phóng to, thu nhỏ.

3) Các thao tác vẽ, tô màu , tô màu viền, tô màu phần giao.

Các thao tác vẽ, tổ màu, tô màu viền, tô màu phần giao sẽ nằm trên WM_PAINT.

```
	HBRUSH hbr,hbr1;
		HPEN hpen;
	hbr = CreateSolidBrush(RGB(0, 255, 255));
		hbr1 = CreateSolidBrush(RGB(255, 0, 0));
		hpen = CreatePen(PS_DOT, 4, RGB(255, 255, 0));
		SelectObject(hdc, hpen);
		SetROP2(hdc, R2_NOTXORPEN);
		SelectObject(hdc, hbr);
		pS->draw(hdc);
		SelectObject(hdc, hbr1);
		pS1->draw(hdc);
```

Thao tác vẽ đối tượng 1 là *pS->draw(hdc)* vẽ đối tượng 2 là *pS1->draw(hdc)*

Thao tác tô màu dùng *hbrush*, tô màu viền dùng *hpen*, đổi màu phần giao là *setRPO2*.