

# **ĐỒ HỌA 2D**

# **QUI TRÌNH HIỂN THỊ**

# Các bước hiển thị

---

**Input** : Mô hình các đối tượng

**Output** : Hình ảnh

bước 1 : Biến đổi các đối tượng

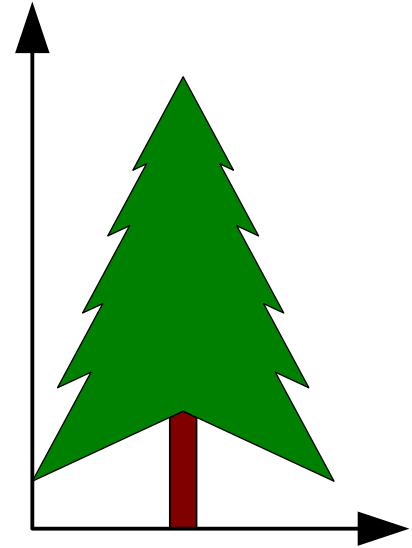
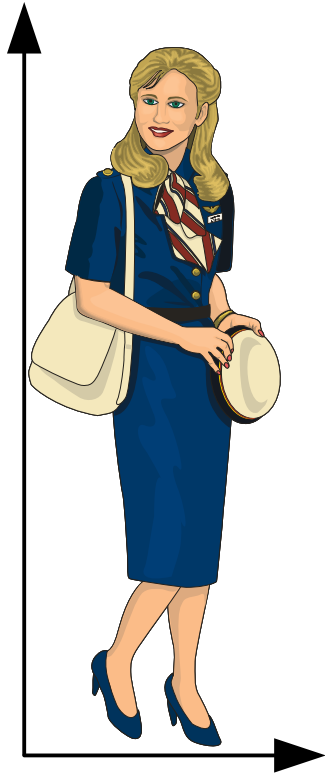
bước 2 : Xén các đối tượng

bước 3 : Chuyển hệ tọa độ

bước 4 : Vẽ các đối tượng.

# Mô hình đối tượng

---



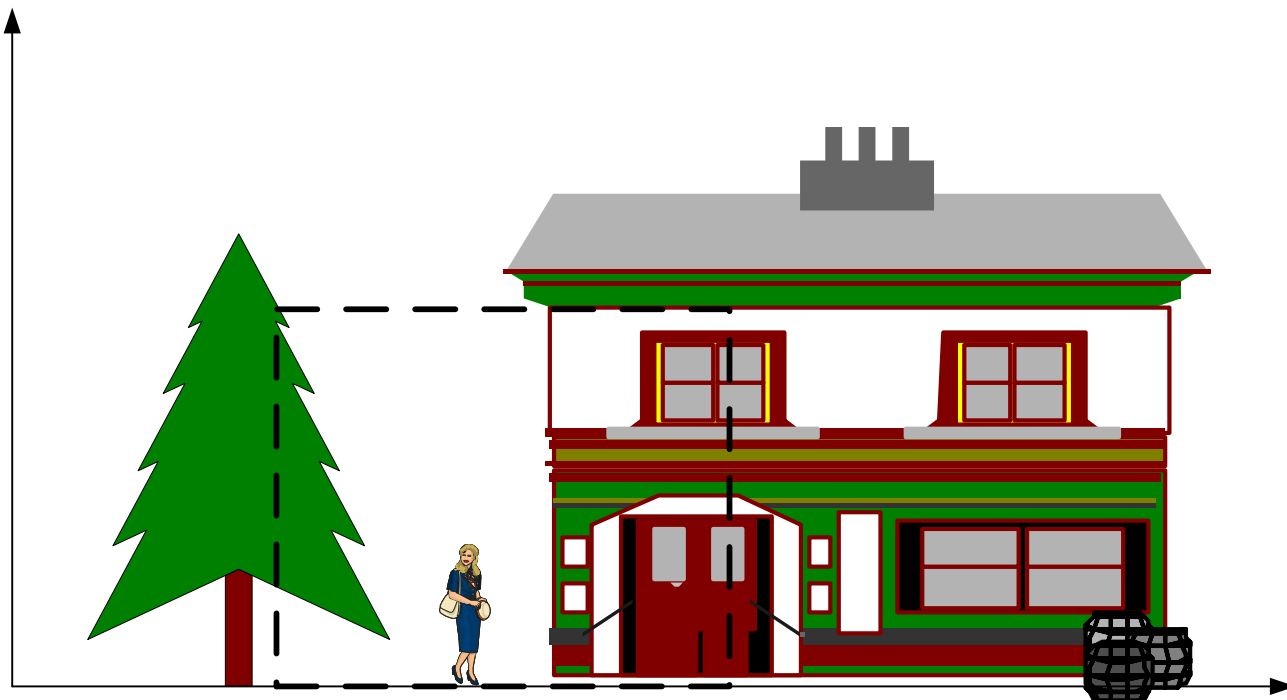
# bước 1 : biến đổi

---



## bước 2 + 3 : xén + chuyển tọa độ

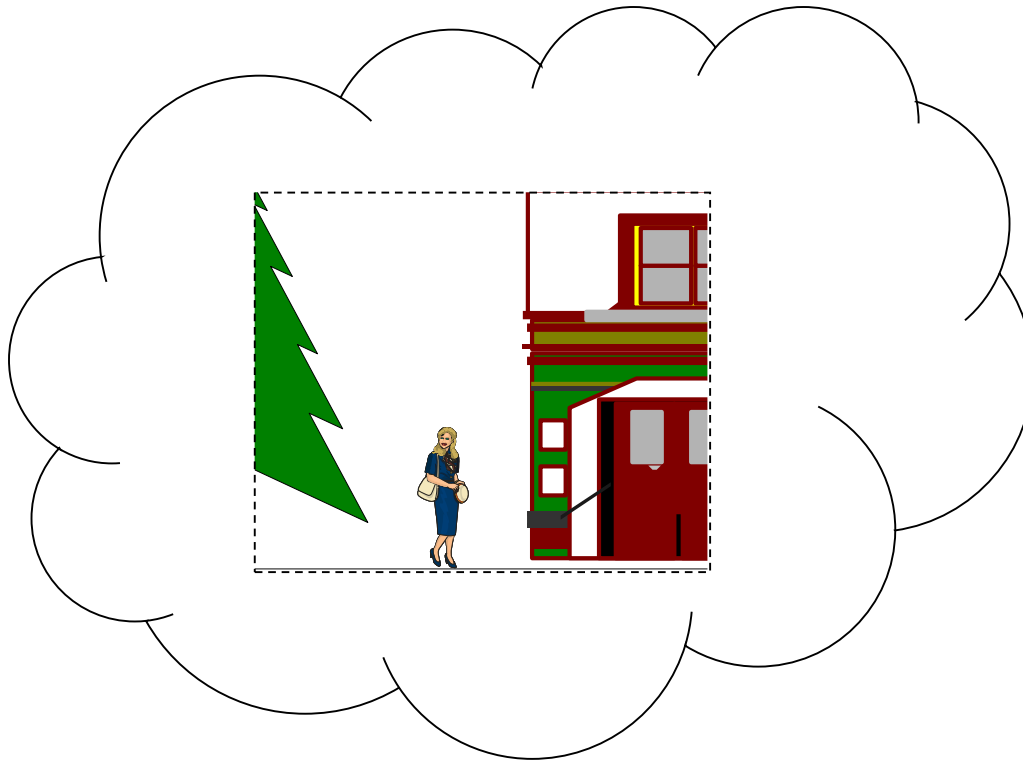
---



## bước 4 : vẽ

---

Kết quả



# MÔ HÌNH

# Đối tượng được biểu diễn như thế nào ?

---

## Các đối tượng cơ sở

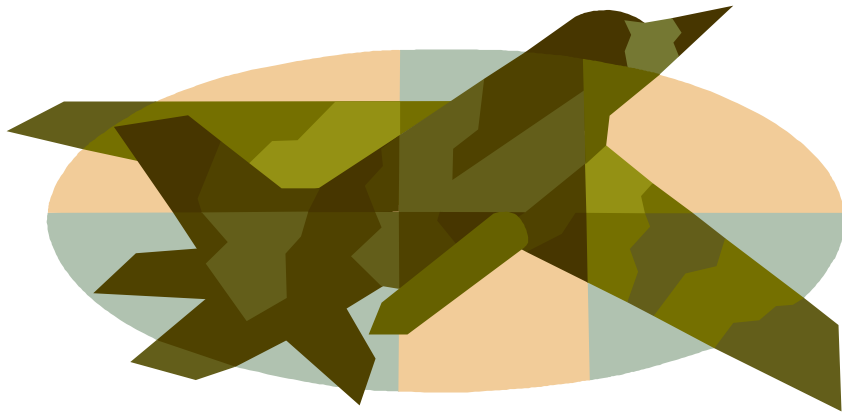
- Điểm
- Đoạn thẳng
- Đa giác
- Đường cong (bezier, spline, bspline ...)

**Các đối tượng phức tạp** được tạo thành từ các đối tượng cơ sở



# Tập tin dữ liệu

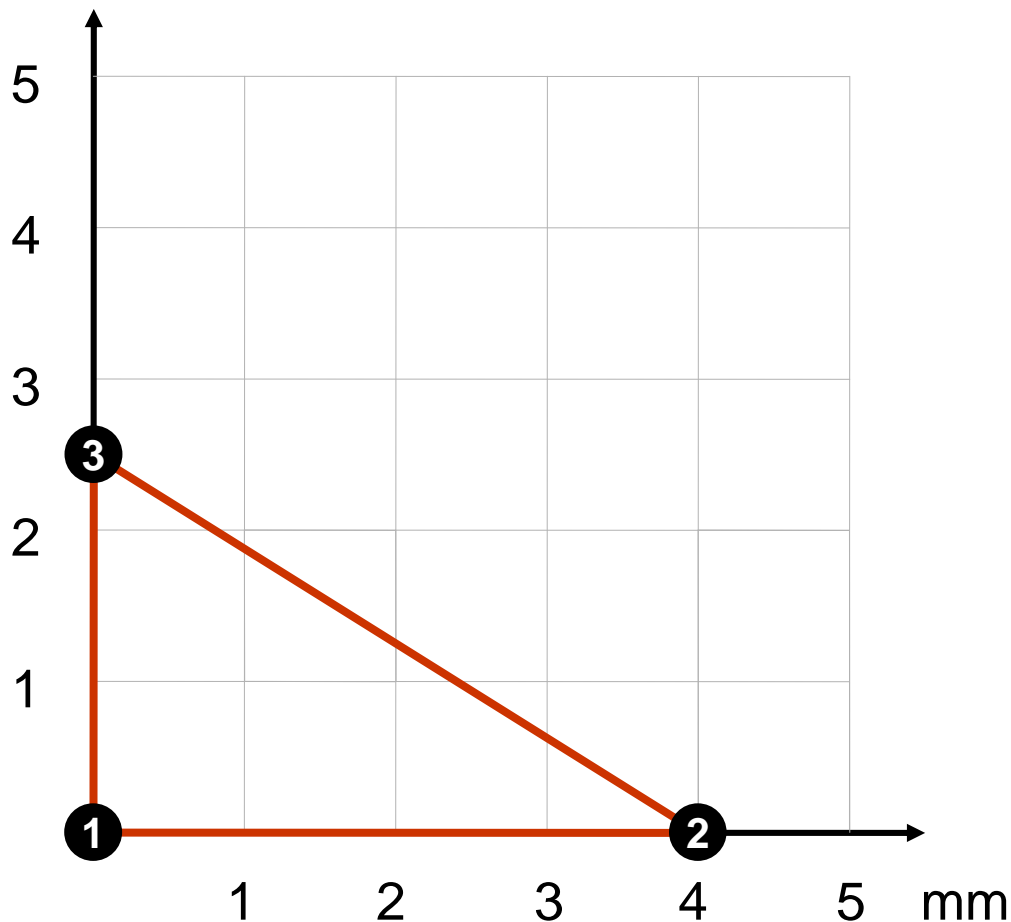
---



.TXT

- đối tượng cơ sở 1
- đối tượng cơ sở 2
- 
- 
- 
-

# Biểu diễn bằng đoạn thẳng



**line**  $x_1$   $y_1$   $x_2$   $y_2$   $r$   $g$   $b$

TAM\_GIAC.TXT

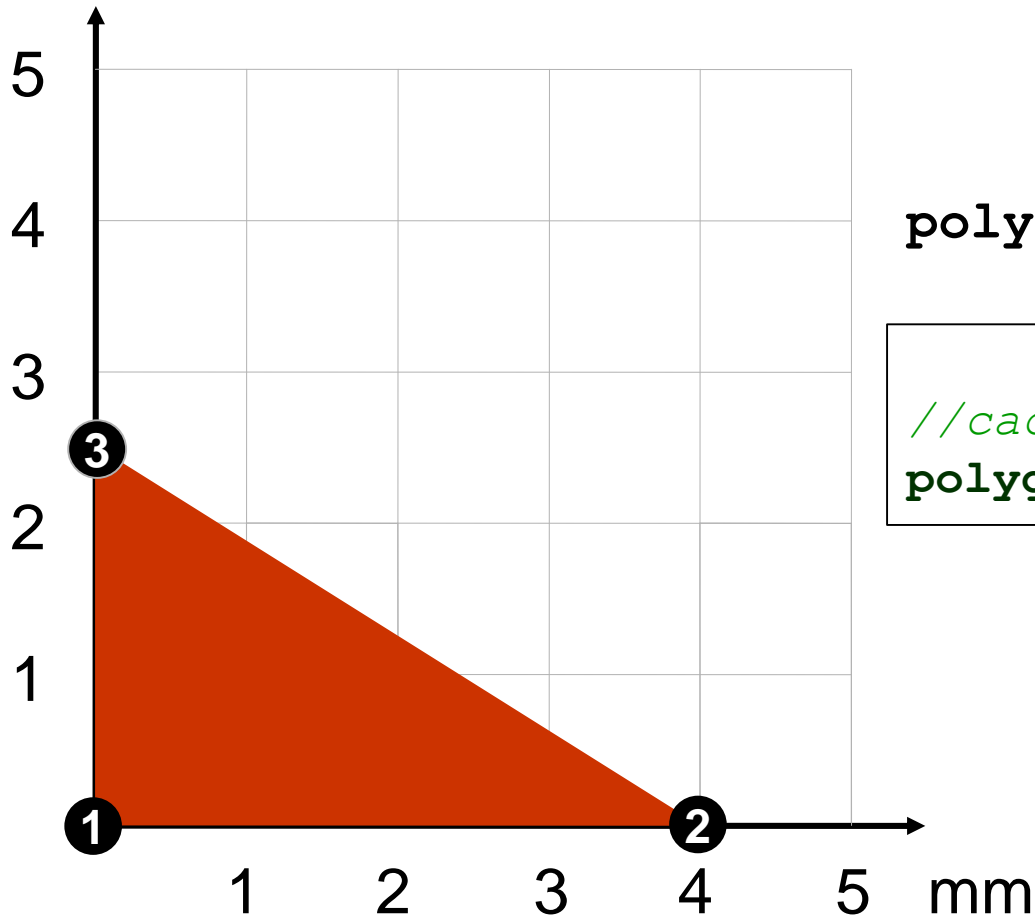
*//cac doan thang*

**line** 0 0 4 0 255 0 0

**line** 4 0 0 2.5 255 0 0

**line** 0 2.5 0 0 255 0 0

# Biểu diễn bằng đa giác



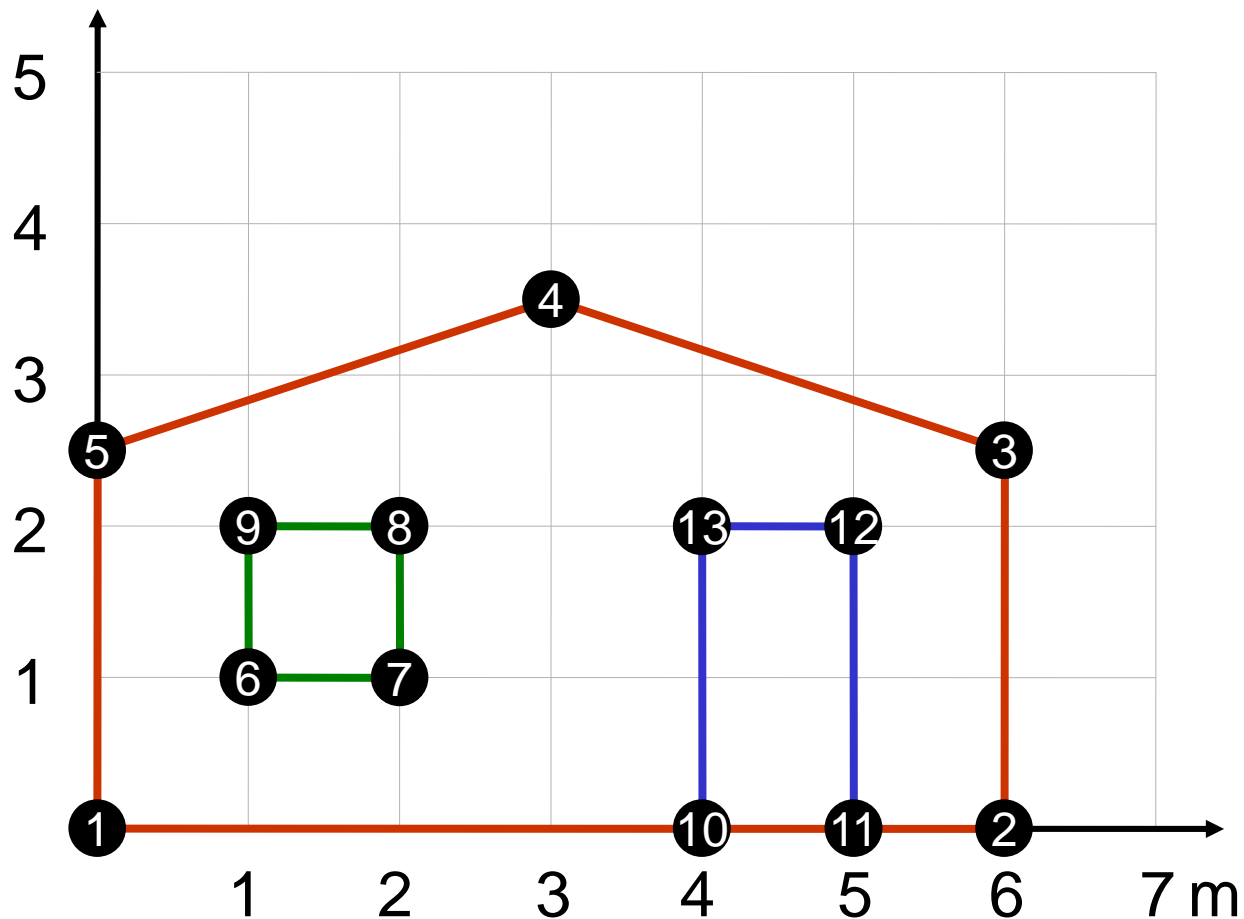
**polygon** *n*  $x_1$   $y_1$  ...  $x_n$   $y_n$  *r* *g* *b*

TAM\_GIAC.TXT

*//cac da giac*

**polygon** 3 0 0 4 0 0 2.5 255 0 0

# Ví dụ



NHA.TXT

*//cac doan thang*

**line** 0 0 6 0 255 0 0

**line** 6 0 6 2.5 255 0 0

**line** 6 2.5 3 3.5 255 0 0

**line** 3 3.5 0 2.5 255 0 0

**line** 0 2.5 0 0 255 0 0

**line** 1 1 2 1 0 255 0

**line** 2 1 2 2 0 255 0

**line** 2 2 1 2 0 255 0

**line** 1 2 1 1 0 255 0

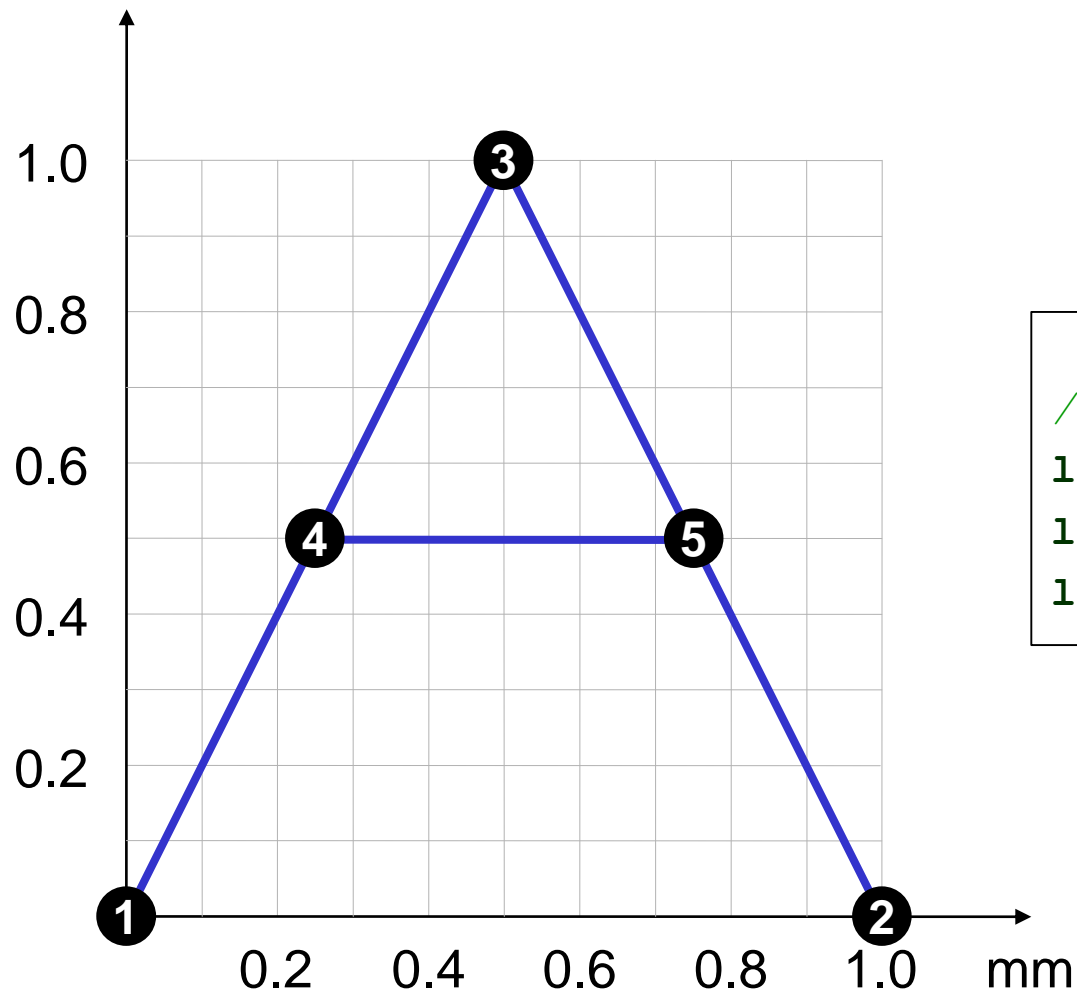
**line** 4 0 5 0 0 0 255

**line** 5 0 5 2 0 0 255

**line** 5 2 4 2 0 0 255

**line** 4 2 4 0 0 0 255

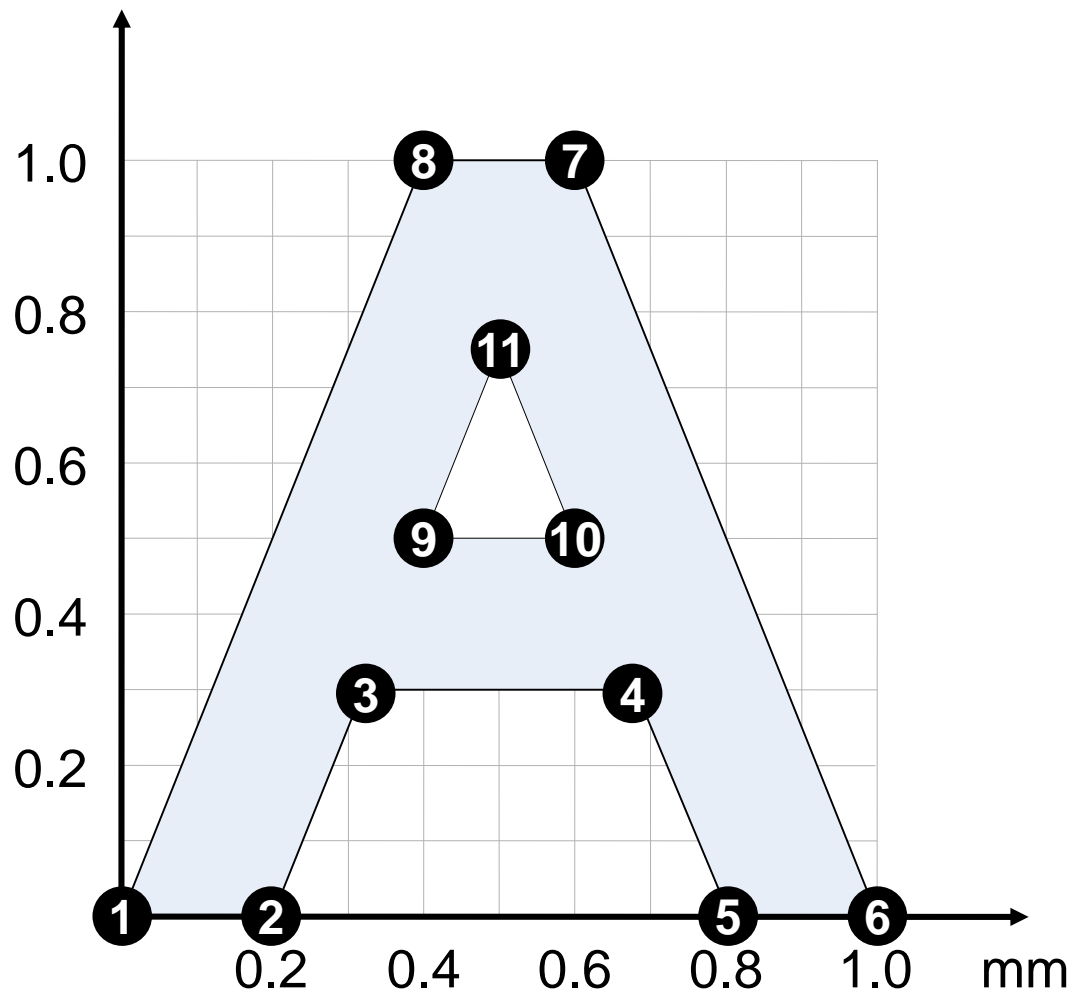
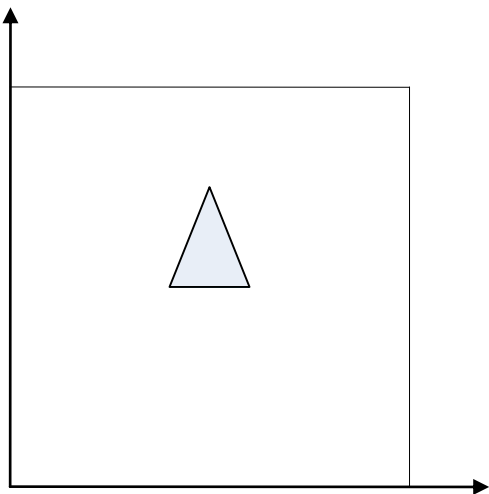
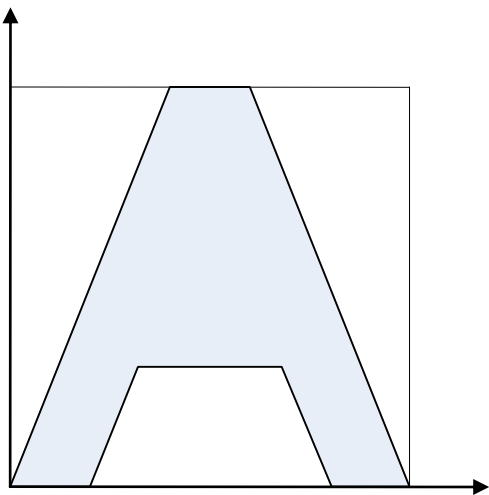
# Ví dụ



A.TXT

```
// cac doan thang  
line 0 0 0.5 1          0 0 0  
line 1 0 0.5 1          0 0 0  
line 0.25 0.5 0.75 0.5 0 0 0
```

# Ví dụ



# Cấu trúc dữ liệu

---

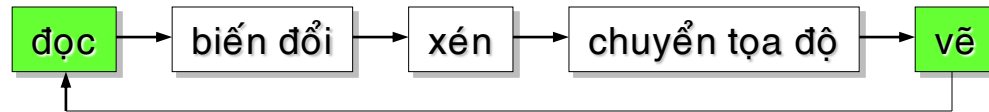
*// Lưu tọa độ của một điểm*

```
struct TPoint2D {  
    double x, y;  
};
```

*// Lưu thông tin của một đa giác*

```
struct TPolygon2D {  
    int n;  
    TPoint2D point[100];  
};
```

# Cài đặt



```
void Read_Draw_2D(CDC *pDC, char *filename)
{
    ifstream f(filename);
    if(f.fail()) return;
    while(!f.eof())
    {
        char object[32];
        f >> object;
        if(strcmpi(object, "LINE") == 0)
        {
            TPoint2D P1, P2;
            int r, g, b;
            // Doc doan thang
            f >> P1.x >> P1.y >> P2.x >> P2.y >> r >> g >> b;
            // Ve doan thang
            pDC->MoveTo(P1.x, P1.y);
            pDC->LineTo(P2.x, P2.y);
        }
    }
}
```

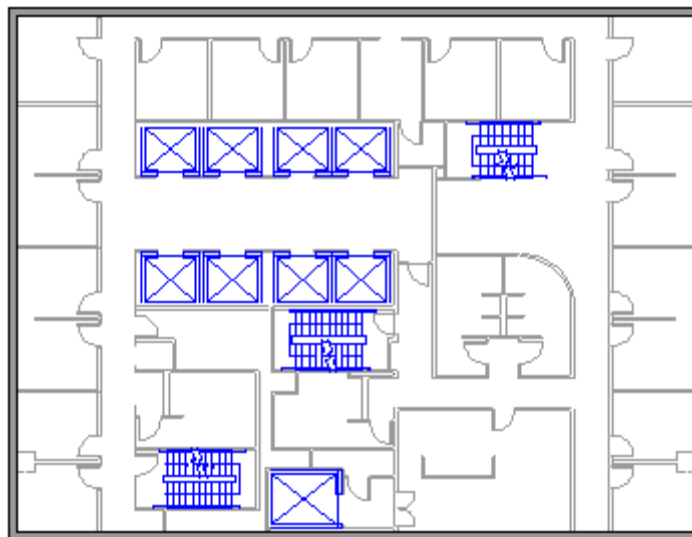


**ĐỒ HỌA 2D**

**BIẾN ĐỔI TỌA ĐỘ**

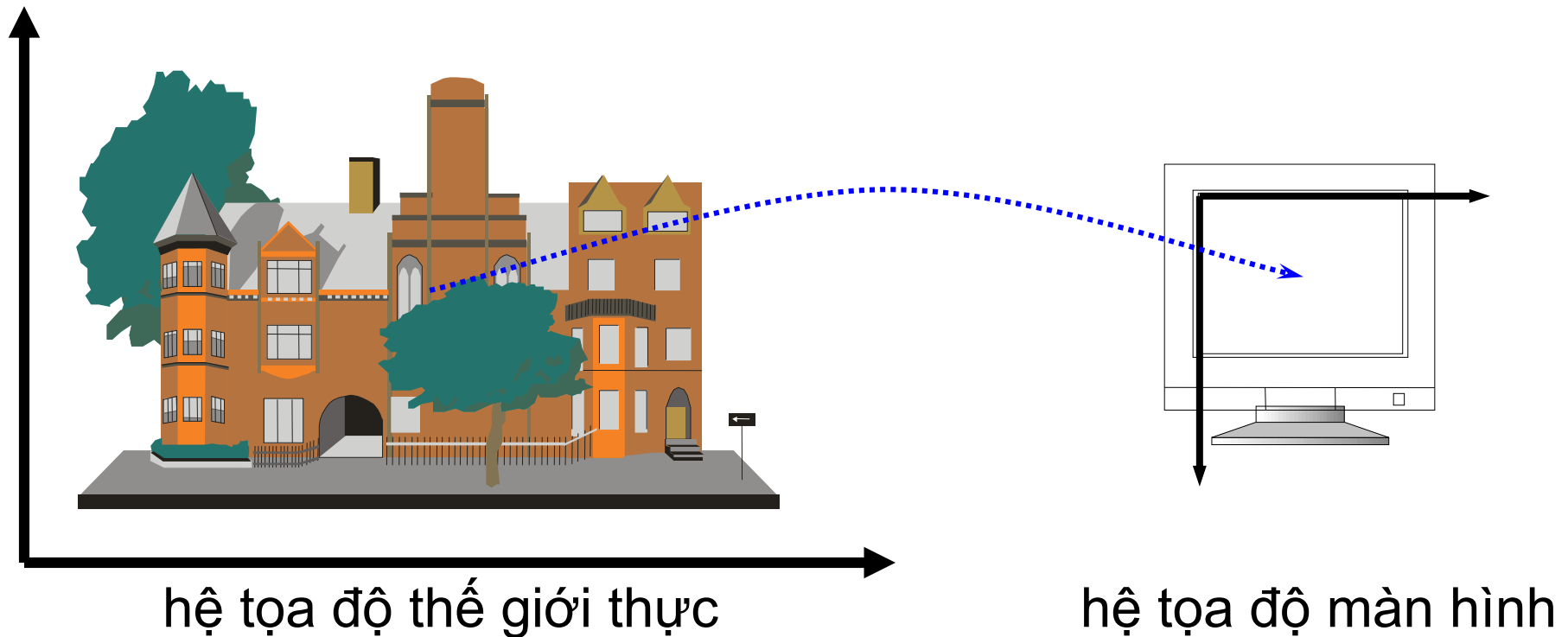
# Dẫn nhập

---

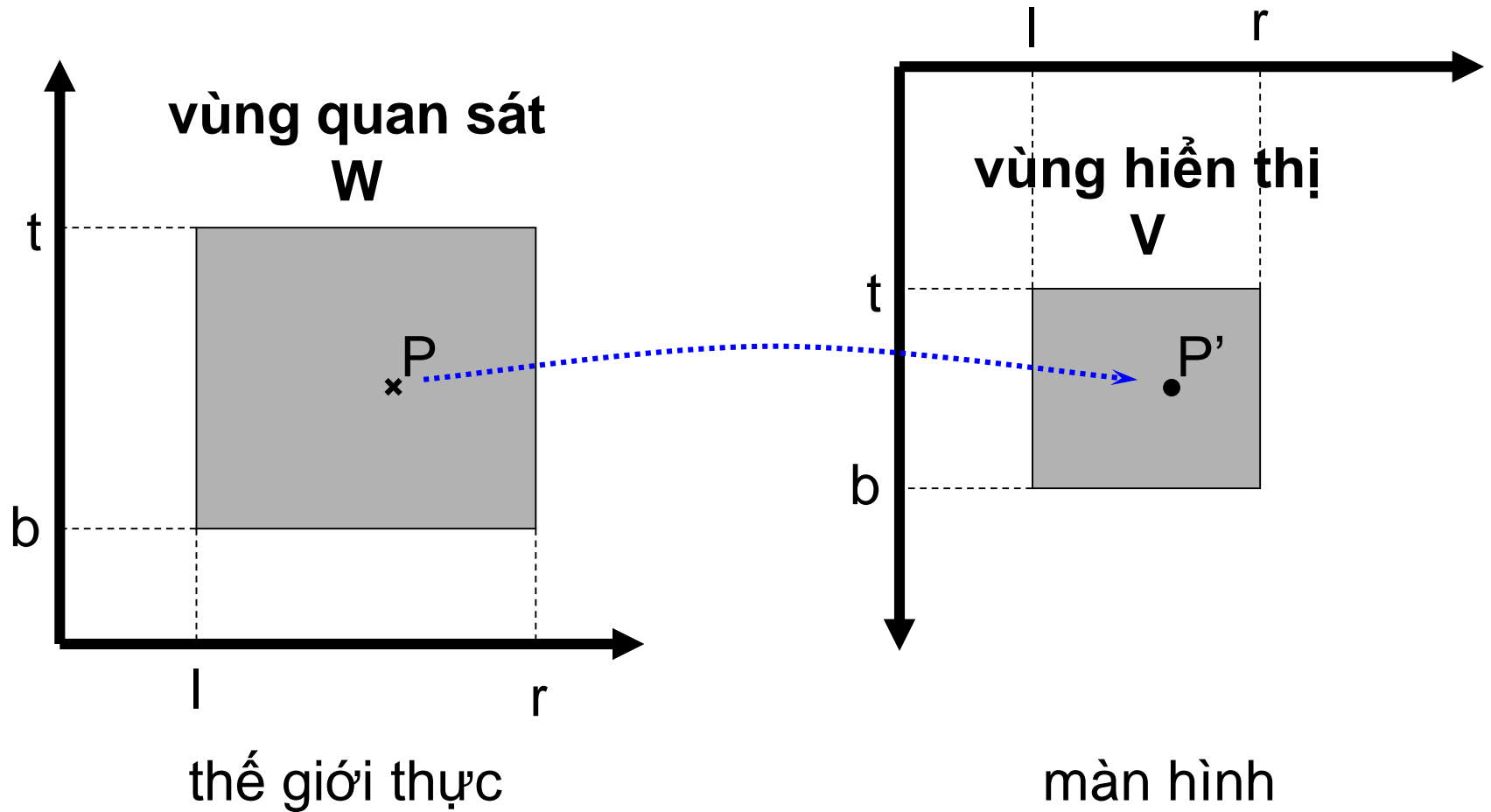


# Biến đổi tọa độ window – viewport là gì ?

Là phép chuyển đổi tọa độ thế giới thực hai chiều sang tọa độ màn hình.



# Công thức



# Công thức

---

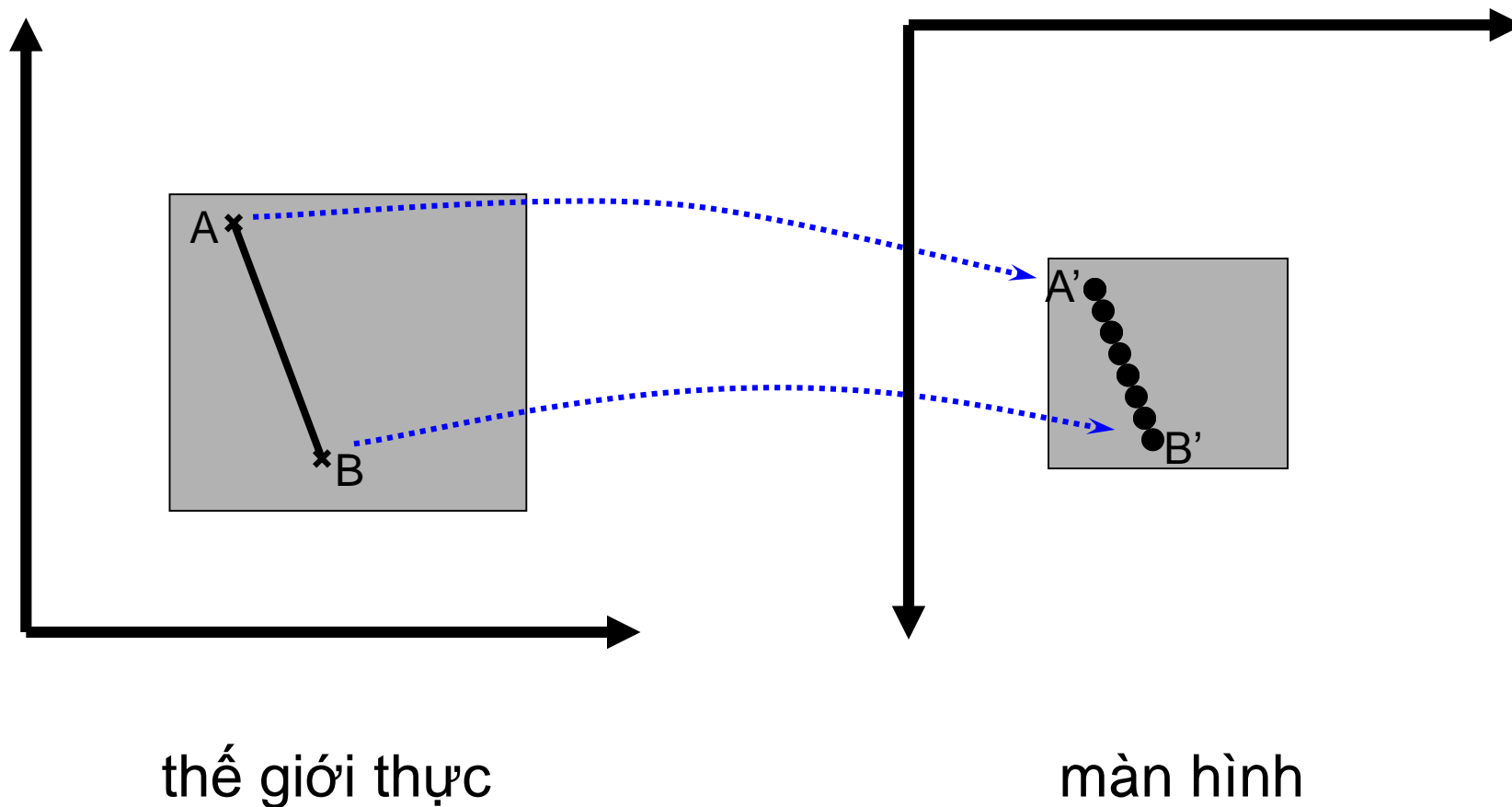
Thế giới thực  $\rightarrow$  màn hình

$$\left| \begin{aligned} P'_x &= \left( P_x - \frac{W_r + W_l}{2} \right) \frac{V_r - V_l}{W_r - W_l} + \frac{V_r + V_l}{2} \\ P'_y &= \left( P_y - \frac{W_t + W_b}{2} \right) \frac{V_t - V_b}{W_t - W_b} + \frac{V_t + V_b}{2} \end{aligned} \right.$$

Màn hình  $\rightarrow$  Thế giới thực

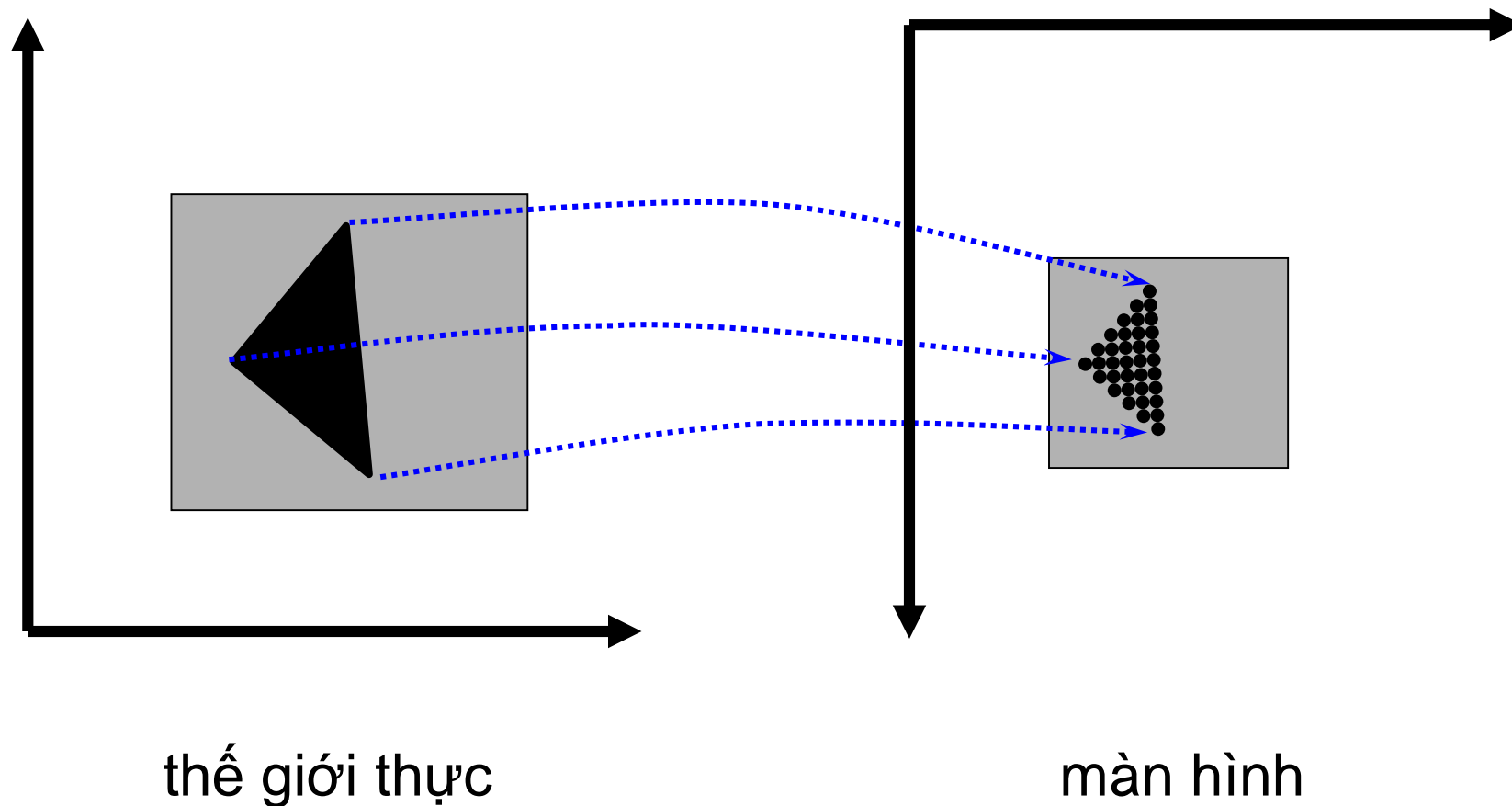
$$\left| \begin{aligned} P_x &= \left( P'_x - \frac{V_r + V_l}{2} \right) \frac{W_r - W_l}{V_r - V_l} + \frac{W_r + W_l}{2} \\ P_y &= \left( P'_y - \frac{V_t + V_b}{2} \right) \frac{W_t - W_b}{V_t - V_b} + \frac{W_t + W_b}{2} \end{aligned} \right.$$

# Công thức cho đoạn thẳng



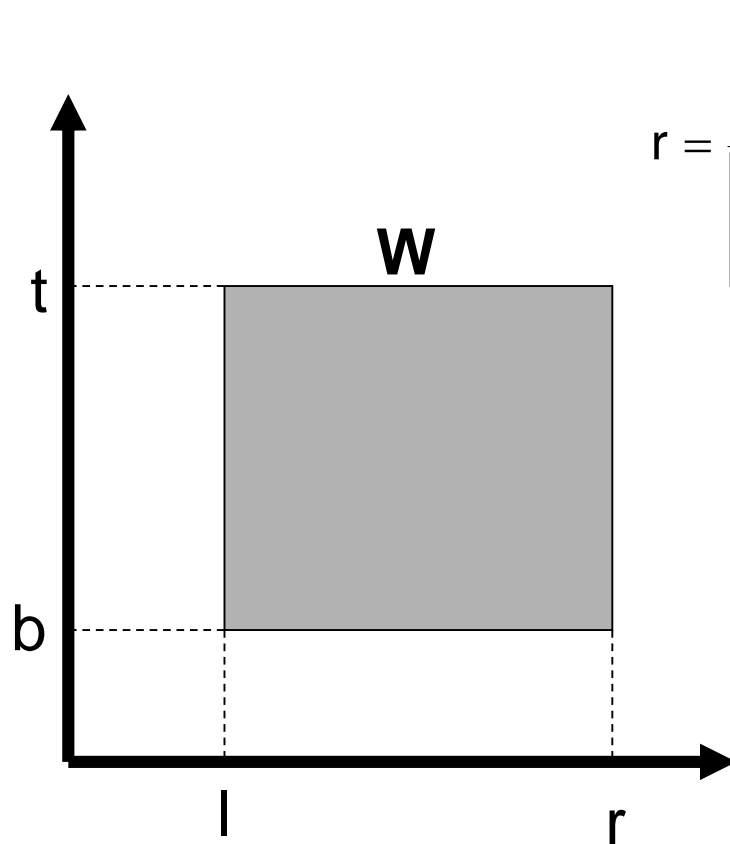
# Công thức cho đa giác

---



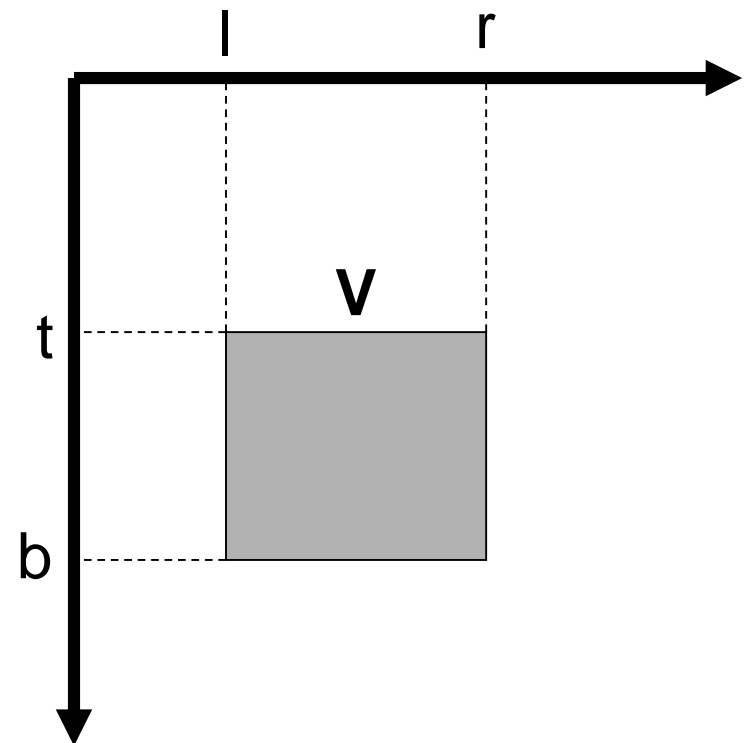
# Hệ số méo

Khi thiết lập cửa sổ **window**  $W$  và cửa sổ **viewport**  $V$  phải lưu ý đến hệ số méo  $r$



thế giới thực

$$r = \frac{\left| \frac{V_r - V_l}{W_r - W_l} \right|}{\left| \frac{V_t - V_b}{W_t - W_b} \right|}$$

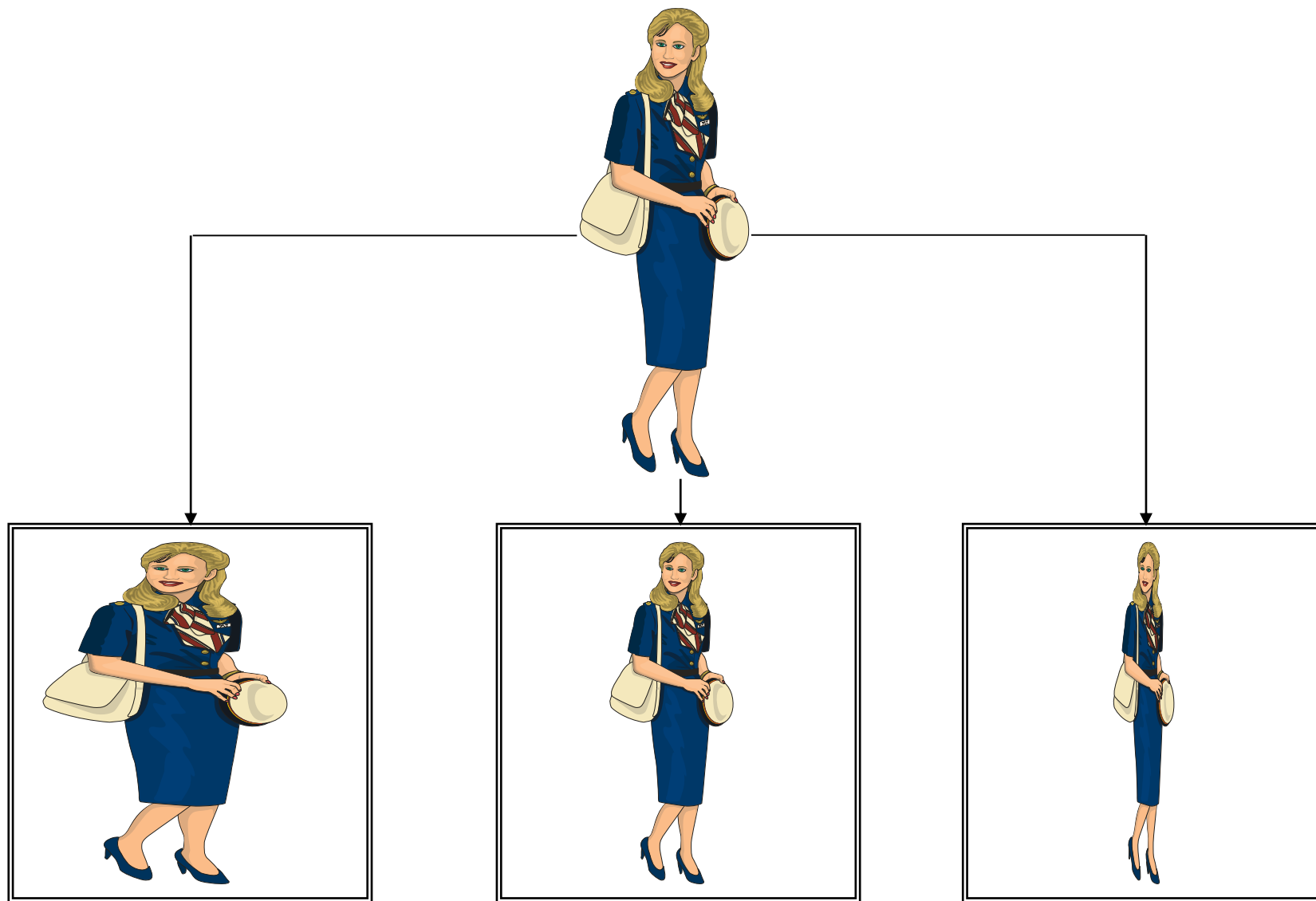


màn hình



# Hệ số méo

---



# Cấu trúc dữ liệu

---

*// Lưu thông tin cửa sổ quan sát*

```
struct TWindow2D {  
    double l, r, b, t;  
};
```

*// Lưu thông tin cửa sổ hiển thị*

```
struct TViewport2D {  
    int l, r, b, t;  
};
```

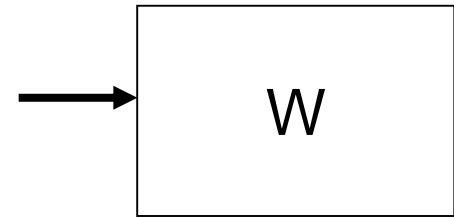
*// Khai báo các cửa sổ*

```
TWindow2D W;  
TViewport2D V;
```

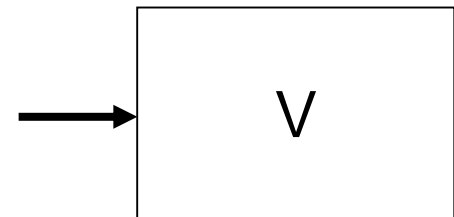
# Cài đặt

---

```
void SetWindow2D(double l, double r, double b, double t)
{
    W.l = l;
    W.r = r;
    W.b = b;
    W.t = t;
}
```



```
void SetViewport2D(int l, int r, int b, int t)
{
    V.l = l;
    V.r = r;
    V.b = b;
    V.t = t;
}
```



# Cài đặt

```
CPoint ConvertWorldToScreen2D(double x, double y)
{
    CPoint Q;

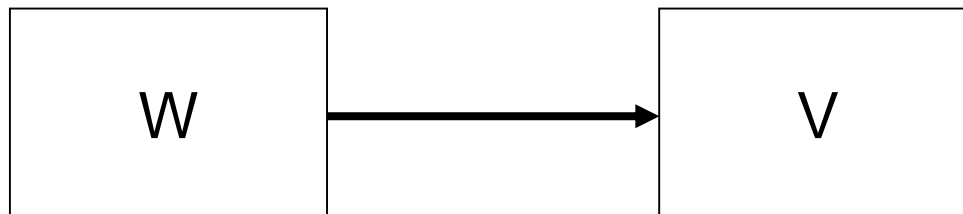
    Q.x=(x-(W.r+W.l)/2)*(V.r-V.l)/(W.r-W.l)+(V.r+V.l)/2;
    Q.y=(y-(W.t+W.b)/2)*(V.t-V.b)/(W.t-W.b)+(V.t+V.b)/2;

    return Q;
}
```

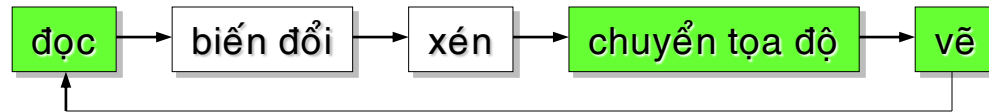
```
CPoint ConvertWorldToScreen2D(TPoint2D P)
{
    CPoint Q;

    Q.x=(P.x-(W.r+W.l)/2)*(V.r-V.l)/(W.r-W.l)+(V.r+V.l)/2;
    Q.y=(P.y-(W.t+W.b)/2)*(V.t-V.b)/(W.t-W.b)+(V.t+V.b)/2;

    return Q;
}
```



# Cài đặt



```
void Read_Convert_Draw_2D(CDC *pDC, char *filename)
{
    ifstream f(filename);
    if(f.fail()) return;
    while(!f.eof())
    {
        char object[32];
        f >> object;
        if(strcmpi(object, "LINE") == 0)
        {
            TPoint2D P1, P2;
            CPoint Q1, Q2;
            int r, g, b;
            // Doc doan thang
            f >> P1.x >> P1.y >> P2.x >> P2.y >> r >> g >> b;
            // Chuyen toa do
            Q1 = ConvertWorldToScreen2D(P1);
            Q2 = ConvertWorldToScreen2D(P2);
            // Ve doan thang
            pDC->MoveTo(Q1);
            pDC->LineTo(Q2);
        }
    }
}
```