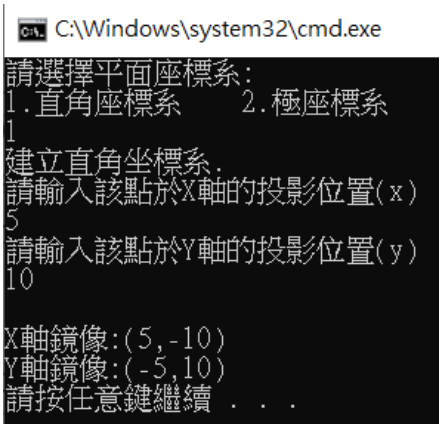
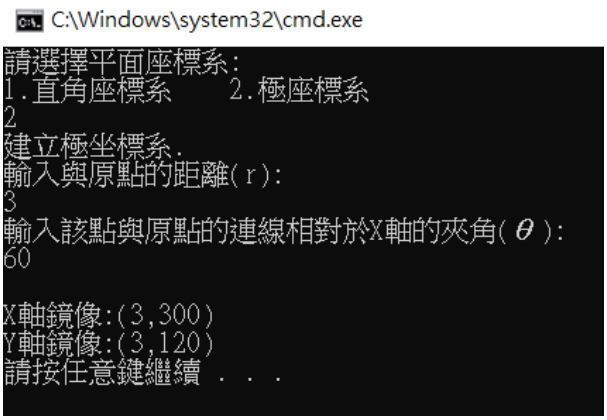
作業：座標轉換

- 請撰寫三個類別。第一個類別為平面座標系類別，第二個為直角座標系類別，第三個為極座標系類別，平面座標系類別為另外兩個類別的父類別。
- 所有座標皆以整數表示，座標系必須具備鏡像的功能，且必須支援多型技術。





Program.cs

```
using System;

namespace 平面座標系
{
    class Program
    {
        static void Main(string[] args)
        {
            CoordinateSystem coordinatesystem = new CoordinateSystem();
            Console.WriteLine("請選擇平面座標系:\n1.直角座標系\t2.極座標系");
            choosesystem mychoose =
(choosesystem)Convert.ToInt32(Console.ReadLine());
            switch (mychoose)
            {
                case choosesystem.rectangularcoordinate:
                    coordinatesystem = new rectangularcoordinate();
                    Console.WriteLine("請輸入該點於X軸的投影位置(x)");
                    coordinatesystem.X = Convert.ToInt32(Console.ReadLine());
                    Console.WriteLine("請輸入該點於Y軸的投影位置(y)");
                    coordinatesystem.Y = Convert.ToInt32(Console.ReadLine());
```

```
                    Console.WriteLine();
                    break;
                case choosesystem.polarcoordinate:
                    coordinatesystem = new polarcoordinate();
                    Console.WriteLine("輸入與原點的距離(r):");
                    coordinatesystem.X = Convert.ToInt32(Console.ReadLine());
                    Console.WriteLine("輸入該點與原點的連線相對於X軸的夾角(θ):");
                    coordinatesystem.Y = Convert.ToInt32(Console.ReadLine());
                    Console.WriteLine();
                    break;
                default:
                    Console.WriteLine("輸入錯誤");
                    break;
            }

            coordinatesystem.Mirror(mirrorType.Xaxis);
            coordinatesystem.Mirror(mirrorType.Yaxis);
        }
    }
}
```

MyClass.cs

```
namespace 平面座標系
{
    enum choosesystem
    {
        none,
        rectangularcoordinate,
        polarcoordinate
    }
    enum mirrorType
    {
        Xaxis,
        Yaxis,
    }
    class CoordinateSystem
```

```
    {
        public CoordinateSystem()
        {
            m_X = 0;
            m_Y = 0;
        }
        public int m_X;
        public int m_Y;
        public int X
        {
            get
            {
                return m_X;
            }
            set
            {
                m_X = value;
            }
        }
        public int Y
        {
            get
            {
                return m_Y;
            }
            set
            {
                m_Y = value;
            }
        }
        virtual public void Mirror(mirrorType type)
        {
           // Console.Write("{0}使出", m_X);
        }
    }
}
```

Class1.cs

```
using System;
namespace 平面座標系
{
    class rectangularcoordinate : CoordinateSystem
    {
        public rectangularcoordinate()
        {
            Console.WriteLine("建立直角坐標系.");
        }
        override public void Mirror(mirrorType type)
        {
            base.Mirror(type);
            if (type == mirrorType.Xaxis) Console.WriteLine("X軸鏡像:({0},{1})",m_X,-
m_Y);


            else Console.WriteLine("Y軸鏡像:({0},{1})", -m_X, m_Y);
        }
    }
}
```

Class2.cs

```
using System;
namespace 平面座標系
{
    class polarcoordinate : CoordinateSystem
    {
        public polarcoordinate()
        {
            Console.WriteLine("建立極坐標系.");
        }
        override public void Mirror(mirrorType type)
        {
            base.Mirror(type);
            if (type == mirrorType.Xaxis) Console.WriteLine("X軸鏡像:({0},{1})", m_X,
360 - m_Y);
```

```
            else Console.WriteLine("Y軸鏡像:({0},{1})", m_X, 180 - m_Y);
        }
    }
}
```