通識計算機程式設計期末考參考解答

6/24/2016

1.

(a) (3%)

```
interface SwitchableDevice
{
  void TurnOn();
  void TurnOff();
  SwitchStatus Status();
  string Name();
}
```

(b) (6%)

```
public AirConditioner(string name)
{
  this.name = name;
  status = SwitchStatus.OFF;
  temperatureSetting = 28;
  rand = new Random();
}
```

(c) (6%)

```
public void TurnOn()
{
    if(DetectedTemperature() < temperatureSetting)
        return;
    if(status == SwitchStatus.OFF)
    {
        status = SwitchStatus.ON;
    }
}
```

(d) (3%)

```
public Lamp(string name)
{
  this.name = name;
  status = SwitchStatus.OFF;
}
```

```
public void TurnOn()
{
    if(status == SwitchStatus.OFF)
    {
        status = SwitchStatus.ON;
    }
}

public void TurnOff()
{
    if(status == SwitchStatus.ON)
    {
        status = SwitchStatus.OFF;
    }
}

public SwitchStatus Status()
{
    return status;
}

public string Name()
{
    return name;
}
```

(e) (6%)

```
public Button(SwitchableDevice device)
{
    status = ButtonStatus.RELEASED;
    this.device = device;
}
```

(f) (6%)

```
Button[] buttons = new Button[2];
Lamp lamp = new Lamp("Lamp_Porch");
AirConditioner air = new AirConditioner("AC_LivingRoom");
buttons[0] = new Button(lamp);
buttons[1] = new Button(air);
```

```
   buttons[0].Pressed();
   buttons[1].Pressed();
```

2. 找出以下程式片段之錯誤，並予更正.

(a) (3%) 一個錯誤

```
class Rectangle
{
    private int width;
    private int height;
    public Rectangle(int w, int h)
    {
        width = w;
        height = h;
    }
    public int Width
    {
        get { return width; }
    }
    public int Height
    {
        get { return height; }
    }
}
class Program
{
    static void Main(string[] args)
    {
        Rectangle rect = new Rectangle(4, 6);
   }
}
```

(b) (3%) 一個錯誤

```
struct ComplexNumber
{
    public double real;
    public double imag;
    public ComplexNumber()  // struct 不可以加入預設建構式
    {
        real = 0.0;
        imag = 0.0;
    }
    public ComplexNumber(double re, double im)
    {
        real = re;
        imag = im;
    }
}
```

(c) (3%) 一種錯誤。

```
   class Plane  // 須改為 interface
   {
       void Fly()
       {
         Console.WriteLine("Fly");
       }
   }
   class Ship // 須改為 interface
   {
       void Sail()
       {
         Console.WriteLine("Sail");
       }
   }
   class Floatplane : Plane, Ship  // 類別不可以多重繼承，介面才可以
   {
        public Floatplane()
       {
         Console.WriteLine("建立水上飛機");
       }
   }
```

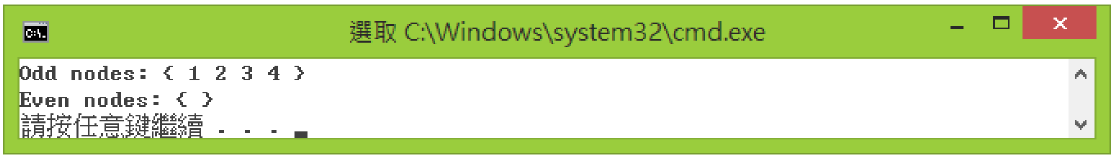
(d) (3%) 一種錯誤

```
class Shape
{
    private string type;
    public Shape(string type)
    {
        this.type = type;
    }
    virtual public void Draw()
    {
        Console.WriteLine("Drawing shape");
    }
}

class Rectangle : Shape
{
    private int width;
    private int height;
    public Rectangle(int w, int h) : base("rectangle")
    {
        width = w;
        height = h;
    }
    override public void Draw()  // 方可完成”多型”要求
    {
        Console.WriteLine("Drawing rectangle");
    }
}
class Circle : Shape
{
    private int radius;
    public Circle(int r) : base("circle")
    {
        radius = r;
    }
    override public void Draw()  // 方可完成”多型”要求
    {
        Console.WriteLine("Drawing circle");
    }
}
class Program
{
    static void Main(string[] args)
    {
        Shape[] shapes = new Shape[2];
        shapes[0] = new Rectangle(8, 6);
        shapes[1] = new Circle(5);

        // 應該達成”多型”的效果
        for(int i=0; i<2; ++i)
        {
            shapes[i].Draw();
        }
    }
```

(e) (3%) 一種錯誤

```
class Rectangle
{
    private int width;
    private int height;
    private static int nRectangles = 0; // 不屬於任一產生的物件
    public Rectangle()
    {
        width = 1;
        height = 1;
        ++nRectangles;
    }
    public Rectangle(int w, int h)
    {
        width = w;
        height = h;
        ++nRectangles;
    }
    public int Width
    {
        get { return width; }
    }
    public int Height
    {
        get { return height; }
    }
    public static int NRectangles()  // 不屬於任一產生的物件
    {
        return nRectangles;
    }
}
class Program
{
    static void Main(string[] args)
    {
        Rectangle rec1 = new Rectangle();
        Rectangle rec2 = new Rectangle(4, 6);
        // 應該顯示一共產生了多少個矩形
        Console.WriteLine(Rectangle.NRectangles());
    }
}
```

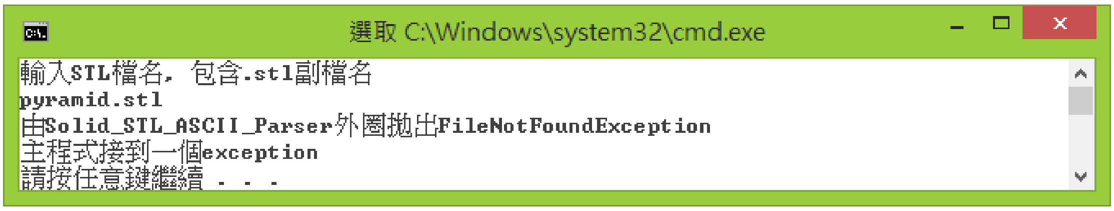
3. (12%)



```
Odd nodes: { 1 2 3 4 }
Even nodes: { }
請按任意鍵繼續 . . .
```

4.

(a) (3%) 檔案 **pyramid.stl** 尚未建立。

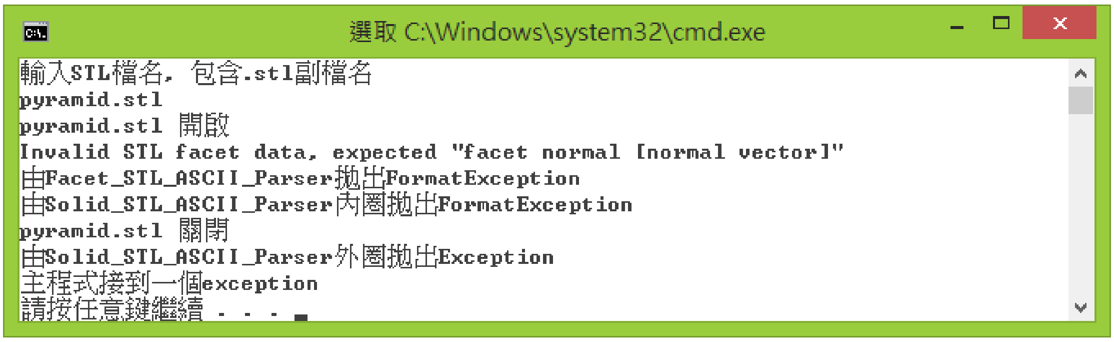


```
輸入STL檔名, 包含.stl副檔名
pyramid.stl
由Solid_STL_ASCII_Parser外圍拋出FileNotFoundException
主程式接到一個exception
請按任意鍵繼續 . . .
```

(b) (3%) 檔案 **pyramid.stl** 已在正確位置，且內容為

```
solid pyramid
facet normal 0.0 0.0 -1.0
```

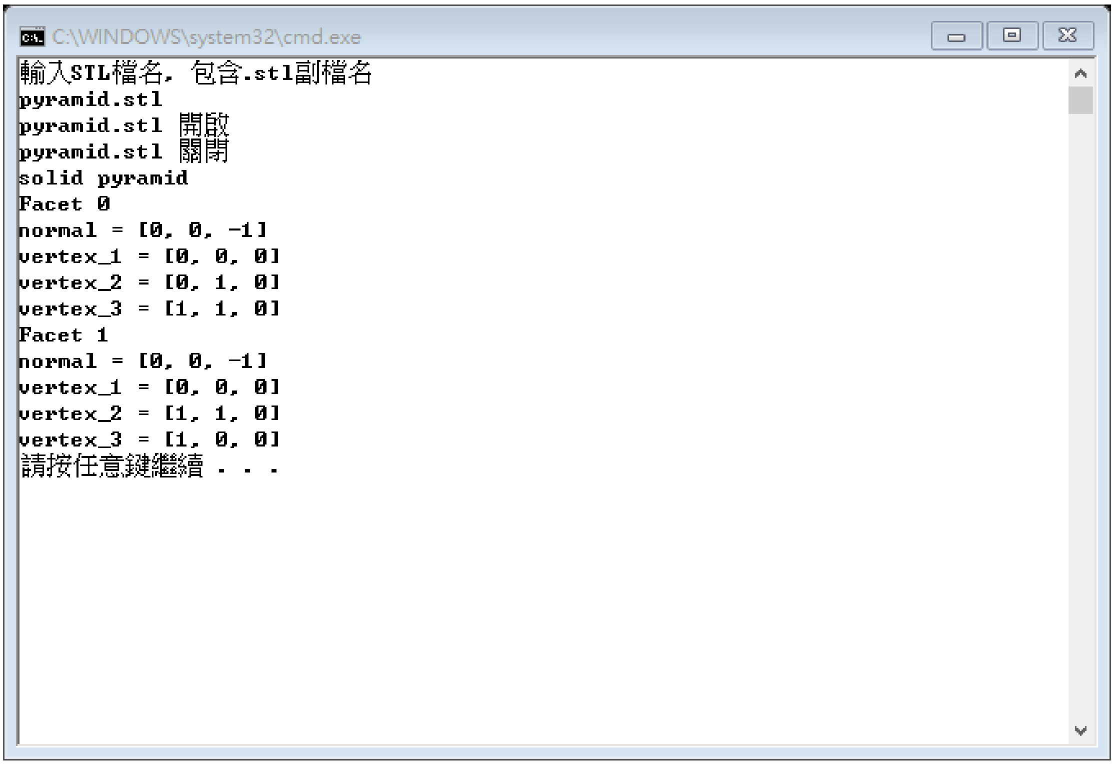
```
outer loop
vertex 0.0 0.0 0.0
vertex 0.0 1.0 0.0
vertex 1.0 1.0 0.0
endloop
endfacet
outer loop
vertex 0.0 0.0 0.0
vertex 1.0 1.0 0.0
vertex 1.0 0.0 0.0
endloop
endfacet
endsolid
```



```
輸入STL檔名，包含.stl副檔名
pyramid.stl
pyramid.stl 開啟
Invalid STL facet data, expected "facet normal [normal vector]"
由Facet_STL_ASCII_Parser拋出FormatException
由Solid_STL_ASCII_Parser內圈拋出FormatException
pyramid.stl 關閉
由Solid_STL_ASCII_Parser外圈拋出Exception
主程式接到一個exception
請按任意鍵繼續 . . .
```

(c) (3%) 檔案 **pyramid.stl** 已在正確位置，且內容為

```
solid pyramid
facet normal 0.0 0.0 -1.0
outer loop
vertex 0.0 0.0 0.0
vertex 0.0 1.0 0.0
vertex 1.0 1.0 0.0
endloop
endfacet
facet normal 0.0 0.0 -1.0
outer loop
vertex 0.0 0.0 0.0
vertex 1.0 1.0 0.0
vertex 1.0 0.0 0.0
endloop
endfacet
```

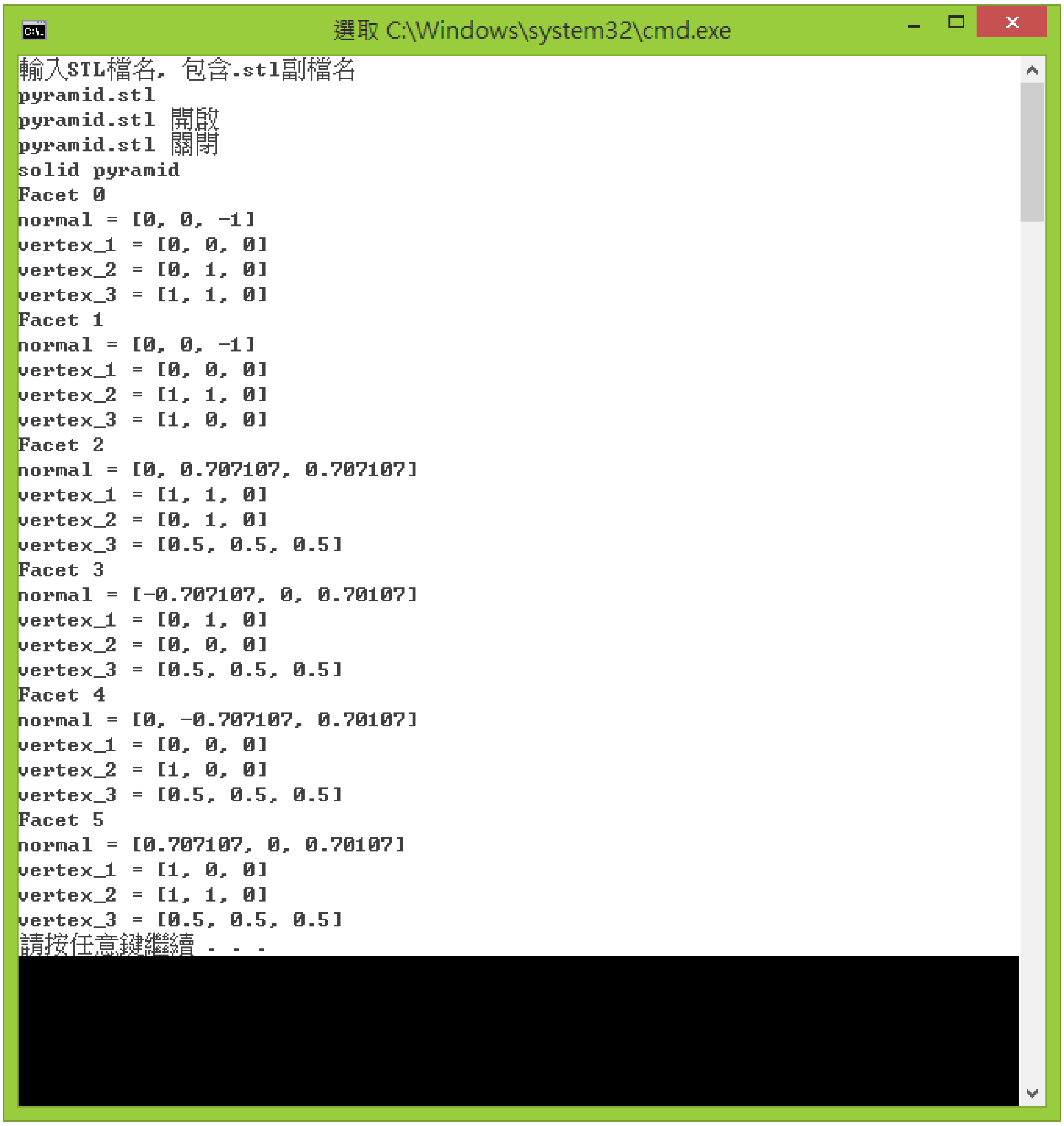


```
輸入STL檔名, 包含.stl副檔名
pyramid.stl
pyramid.stl 開啟
pyramid.stl 關閉
solid pyramid
Facet 0
normal = [0, 0, -1]
vertex_1 = [0, 0, 0]
vertex_2 = [0, 1, 0]
vertex_3 = [1, 1, 0]
Facet 1
normal = [0, 0, -1]
vertex_1 = [0, 0, 0]
vertex_2 = [1, 1, 0]
vertex_3 = [1, 0, 0]
請按任意鍵繼續 . . .
```

(d) (3%) 檔案 **pyramid.stl** 已在正確位置，且內容為

```
solid pyramid
facet normal 0.0 0.0 -1.0
outer loop
vertex 0.0 0.0 0.0
vertex 0.0 1.0 0.0
vertex 1.0 1.0 0.0
endloop
endfacet
facet normal 0.0 0.0 -1.0
outer loop
vertex 0.0 0.0 0.0
vertex 1.0 1.0 0.0
vertex 1.0 0.0 0.0
endloop
endfacet
facet normal 0.0 0.707107 0.707107
outer loop
vertex 1.0 1.0 0.0
vertex 0.0 1.0 0.0
vertex 0.5 0.5 0.5
```

```
endloop
endfacet
facet normal -0.707107 0.0 0.70107
outer loop
vertex 0.0 1.0 0.0
vertex 0.0 0.0 0.0
vertex 0.5 0.5 0.5
endloop
endfacet
facet normal 0.0 -0.707107 0.70107
outer loop
vertex 0.0 0.0 0.0
vertex 1.0 0.0 0.0
vertex 0.5 0.5 0.5
endloop
endfacet
facet normal 0.707107 0.0 0.70107
outer loop
vertex 1.0 0.0 0.0
vertex 1.0 1.0 0.0
vertex 0.5 0.5 0.5
endloop
endfacet
endsolid
```



```
輸入STL檔名，包含.stl副檔名
pyramid.stl
pyramid.stl 開啟
pyramid.stl 關閉
solid pyramid
Facet 0
normal = [0, 0, -1]
vertex_1 = [0, 0, 0]
vertex_2 = [0, 1, 0]
vertex_3 = [1, 1, 0]
Facet 1
normal = [0, 0, -1]
vertex_1 = [0, 0, 0]
vertex_2 = [1, 1, 0]
vertex_3 = [1, 0, 0]
Facet 2
normal = [0, 0.707107, 0.707107]
vertex_1 = [1, 1, 0]
vertex_2 = [0, 1, 0]
vertex_3 = [0.5, 0.5, 0.5]
Facet 3
normal = [-0.707107, 0, 0.70107]
vertex_1 = [0, 1, 0]
vertex_2 = [0, 0, 0]
vertex_3 = [0.5, 0.5, 0.5]
Facet 4
normal = [0, -0.707107, 0.70107]
vertex_1 = [0, 0, 0]
vertex_2 = [1, 0, 0]
vertex_3 = [0.5, 0.5, 0.5]
Facet 5
normal = [0.707107, 0, 0.70107]
vertex_1 = [1, 0, 0]
vertex_2 = [1, 1, 0]
vertex_3 = [0.5, 0.5, 0.5]
請按任意鍵繼續 . . .
```

5. (6%)

```
private void DrawAllSquares(int n, int increment, Graphics g)
{
    // 請依照如下說明完成此處應有的程式敘述 (6%)
    // 畫出n個同心正方形
    // 輸入參數: n 代表要畫的同心正方形數
    //           increment 代表每一層正方形左上角座標比外一層正方形
    //                     左上角座標，在x與y方向都增加的pixel數
    //           g 是視窗程式提供的繪圖工具
    Point topLeft = topLeft0;
    int length = length0;
    for (int i = 0; i < n; ++i)
    {
       square = new Square(length, topLeft);
       square.Draw(g);
       topLeft.x += increment;
       topLeft.y += increment;
       length -= 2 * increment;
    }
}
```

6. (25%)

```
// Problem6.Program
using System;
using System.Collections.Generic;
namespace Problem6
{
     class Program
     {
         static void Main(string[] args)
        {
          Console.WriteLine(
             "輸入 3D部件原始STL檔名，包括副檔名\".stl\" ");
          string inputFileName = Console.ReadLine();
          Solid_STL_ASCII_Parser parser = new
             Solid_STL_ASCII_Parser(inputFileName);
          Solid solid = parser.STL_Solid;

          Console.WriteLine(
             "輸入 伸縮後之新3D部件的STL檔名，包括副檔名\".stl\" ");
          string outputFileName = Console.ReadLine();
          Solid_STL_ASCII_Writer writer = new
             Solid_STL_ASCII_Writer(outputFileName);

          Console.WriteLine(
             "輸入 x， y， z個方向的伸縮比例，各比例間以一個空白分開");
          string line = Console.ReadLine();
```

```
                string[] terms = line.Split(' ');
                double sx = double.Parse(terms[0]);
                double sy = double.Parse(terms[1]);
                double sz = double.Parse(terms[2]);

                solid.Scale(sx, sy, sz);
                writer.Write(solid);
            }
        }
    }

// Problem6. Solid_STL_ASCII_Parser
//略去與第4題重複之程式碼

// Problem6.Facet_STL_ASCII_Parser
//略去與第 4 題重複之程式碼

// Problem6.Solid
namespace Problem6
{
    public class Solid
    {
        //略去與第4題重複之程式碼

        public void Scale(double sx, double sy, double sz)
        {
            foreach(Facet facet in facets)
            {
                facet.Scale(sx, sy, sz);
            }
        }
}

// Problem6.Facet
using System;

namespace Problem6
{
    public class Facet
    {
        //略去與第4題重複之程式碼

        public void Scale(double sx, double sy, double sz)
        {
            r1.x *= sx;
            r1.y *= sy;
            r1.z *= sz;

            r2.x *= sx;
            r2.y *= sy;
            r2.z *= sz;

            r3.x *= sx;
            r3.y *= sy;
            r3.z *= sz;
        }
    }
}

// Problem6.Solid_STL_ASCII_Writer
using System;
using System.Collections.Generic;
using System.IO;

namespace Problem6
{
    class Solid_STL_ASCII_Writer
    {
        private StreamWriter writer;
        public Solid_STL_ASCII_Writer(string fileName)
        {
            try
            {
                writer = new StreamWriter(fileName);
            }
            catch(Exception e)
            {
                Console.WriteLine(e);
            }
        }

        public void Write(Solid solid)
        {
            writer.WriteLine("solid " + solid.Name);
            Facet_STL_ASCII_Writer facetOutput =
              new Facet_STL_ASCII_Writer(writer);
            List<Facet> facets = solid.Facets();
            foreach (Facet facet in facets)
            {
                facetOutput.Write(facet);
            }
            writer.WriteLine("endsolid");
            writer.Close();
        }

    }
}

// Problem6.Facet_STL_ASCII_Writer
using System;
using System.Collections.Generic;
using System.IO;
namespace Problem6
{
    public class Facet_STL_ASCII_Writer
    {
        private StreamWriter writer;
        public Facet_STL_ASCII_Writer(StreamWriter writer)
        {
            this.writer = writer;
        }
        public void Write(Facet facet)
        {
            Vector3D normal = facet.Normal;
            writer.WriteLine("facet normal {0} {1} {2}",
              normal.x, normal.y, normal.z);
            writer.WriteLine("outer loop");
            Vector3D v = facet.Vertex_1;
            writer.WriteLine("vertex {0} {1} {2}", v.x, v.y, v.z);
            v = facet.Vertex_2;
            writer.WriteLine("vertex {0} {1} {2}", v.x, v.y, v.z);
            v = facet.Vertex_3;
            writer.WriteLine("vertex {0} {1} {2}", v.x, v.y, v.z);

            writer.WriteLine("endloop");
            writer.WriteLine("endfacet");
        }
    }
}
```