通識計算機程式設計期末考參考解答, 6/22/2012

1. 參考下一頁的UML類別圖，撰寫C#敘述達成下列要求: (假設**using System;**敘述已經包含於程式中)，均不需處理例外情況

   (a) 撰寫結構 **Point**，包含兩個公有 **int** 變數 **x**、**y**，代表點座標(*x*,*y*)。另寫正規建構式分別設定 **x**、**y** 之值 (3%)

   Ans.

```
struct Point
{
    public int x;
    public int y;
    public Point(int x, int y)
    {
        this.x = x;
        this.y = y;
    }
}
```

   (b) 撰寫類別程式 **Triangle** 代表三角形，其中宣告 **Point** 陣列變數 **vertex** 代表三個頂點；另有預設建構式設定三個頂點座標爲(0,0), (1,0), (0,1)；正規建構式輸入三個頂點座標(x_1, y_1)、(x_2, y_2)、(x_3, y_3)。同時以公式$\frac{1}{2}\left|x_1y_2 - x_2y_1 + x_2y_3 - x_3y_2 + x_3y_1 - x_1y_3\right|$撰寫成員函式 **Area**，另撰寫屬性 **Vertex** 傳回 vertex 之值。(6%)

   Ans.

```
class Triangle
{
    private Point[] vertex = new Point[3];
    public Triangle()
    {
        vertex[0] = new Point(0, 0);
        vertex[1] = new Point(1, 0);
        vertex[2] = new Point(0, 1);
    }
    public Triangle(int x1, int y1, int x2, int y2,
      int x3, int y3)
```

```
    {
        vertex[0] = new Point(x1, y1);
        vertex[1] = new Point(x2, y2);
        vertex[2] = new Point(x3, y3);
    }
    public double Area()
    {
        return 0.5*Math.Abs( vertex[0].x*vertex[1].y -
                             vertex[0].y*vertex[1].x +
                             vertex[1].x*vertex[2].y -
                             vertex[1].y*vertex[2].x +
                             vertex[2].x*vertex[0].y -
                             vertex[2].y*vertex[0].x);
    }
    public Point[] Vertex
    {
        get { return vertex; }
    }
}
```

(c) 撰寫類別 **RTriangle** 繼承類別 **Triangle**，另外具有私有成員變數 **w**、**h**，代表底和高。並撰寫其預設建構式，設直角頂點座標(0,0)，**w** 和 **h** 爲 1，注意此與父類別之預設建構式之關連 (6%)

(d) 撰寫類別 **RTriangle** 之正規建構式，參數爲直角頂點座標(x, y)及底 w 與高 h，呼叫父類別建構式，將三頂點設爲(x, y)、($x+w$, y)、(x, $y+h$) 並撰寫屬性 **W** 及 **H**，傳回成員變數 w、h (6%)

(e) 以關鍵字 **new** 及公式 $\frac{1}{2}w*h$ 撰寫類別 **RTriangle** 的成員函式 **Area**，隱藏(hide)其父類別的同名成員函式 (3%)

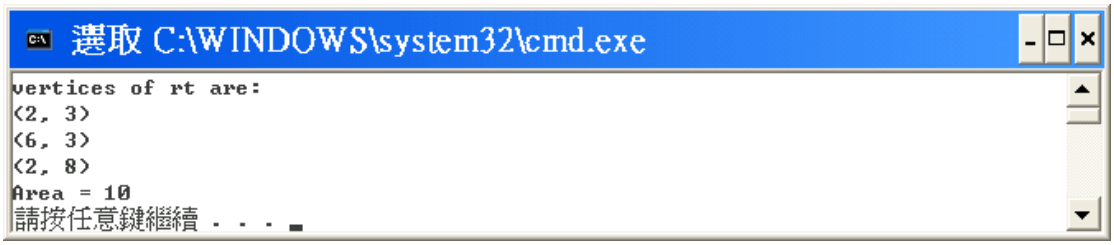
Ans.

```
class RTriangle : Triangle
{
    private int w;
    private int h;
    public RTriangle() : base()
    {
        w = 1;
        h = 1;
```

```
    }
    public RTriangle(int x, int y, int w, int h) :
       base( x, y, x+w, y, x, y+h)
    {
        this.w = w;
        this.h = h;
    }
    public int W
    {
        get { return w; }
    }
    public int H
    {
        get { return h; }
    }
    public new double Area()
    {
        return 0.5 * w * h;
    }

}
```

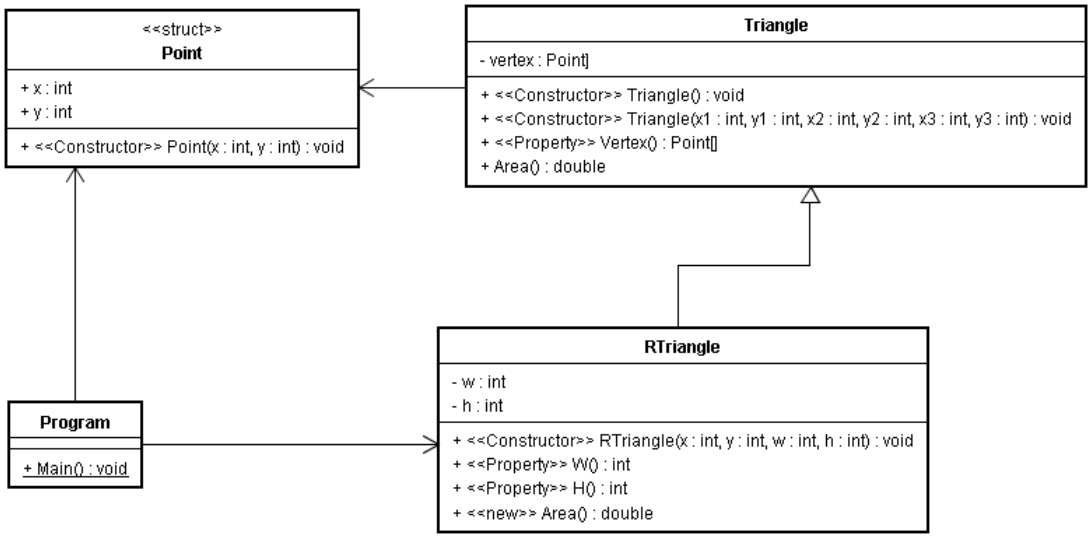
(f) 寫一段測試主程式，設定 **RTriangle** 物件 **rt** 之直角頂點座標爲(2, 3)，**w**、**h** 爲 4、5，再利用它與繼承來的屬性及結構之公開變數，輸出其三頂點座標與本身計算的面積值如下圖 (6%)



Ans.

```
        RTriangle rt = new RTriangle(2, 3, 4, 5);
        Point[] vertex = rt.Vertex;
        Console.WriteLine("vertices of rt are:");
        Console.WriteLine("({0}, {1})", vertex[0].x,
            vertex[0].y);
```

```
        Console.WriteLine("({0}, {1})", vertex[1].x,
            vertex[1].y);
        Console.WriteLine("({0}, {1})", vertex[2].x,
            vertex[2].y);
        Console.WriteLine("Area = " + rt.Area());
```



2. 找出以下程式片段之錯誤，並予更正.

(a) (3%) 一個錯誤

```
class Test1
{
    private int i;
    public Test1()
    {
        i = 1;
    }
    public int I
    {
        get { return i; }
    }
}
class Program
{
    static void Main(string[] args)
    {
```

```
        Test1 test1 = new Test1();
        Console.WriteLine(test1.i);
    }
}
```

Ans.

i 爲 **Test1** 類別的私有成員變數，不可由外部程式取用.

主程式須改爲

```
    static void Main(string[] args)
    {
        Test1 test1 = new Test1();
        Console.WriteLine(test1.I);
    }
```

(b) (3%) 一個錯誤

```
class Test2
{
    private int i;
    public Test2()
    {
        i = 1;
    }
    public Test2(int i)
    {
        i = 3;
    }
    public int I
    {
        get { return i; }
    }
}
```

Ans.

建構式參數與成員變數同名時，需加 **this** 分別

修改建構式爲

```
public Test2(int i)
{
```

```
      this.i = 3;
}
```

(c) (3%).一項錯誤

```
class Test3
{
    private int i;
    public Test3()
    {
        i = 1;
    }
    public int I
    {
        get { return i; }
    }
}

class A : Test3
{
    private int j;
    public A() : base()
    {
        j = 2;
    }
    public int sum()
    {
        return i + j;
    }
}
```

Ans.

類別 **A** 的成員函式 **sum** 中，不可使用父類別 **Test3** 的私有成員變數 **i.**

需將 **i** 改爲 **protected** 或將 **sum** 改爲

```
    public int sum()
    {
        return I + j;
    }
```

(d) (3%) 一個錯誤

```
class Test4
{
    private int i;
    public Test4()
    {
        i = 1;
    }
    public virtual int Triple()
    {
        return 3 * i;
    }
}
class B : Test4
{
    private int j;
    public B() : base()
    {
        j = 2;
    }
    // 欲完成多型
    public int Triple()
    {
        return 3*j;
    }
}
```

類別 **B** 中之成員函式 **Triple** 需覆寫類別 **Test4** 中之同名成員函式即

```
    public override int Triple()
    {
        return 3*j;
    }
```

(e) (3%) 一個錯誤

```
interface C
```

```
{
    int Func1();
    void Func2();
}

class Test5 : C
{
    private int i;
    public Test5()
    {
        i = 1;
    }
    public int Func1()
    {
        return 2 * i;
    }
}
```

Ans.

實作介面時，其中的所有函式均需實作

故類別**Test5**應改爲

```
class Test5 : C
{
    private int i;
    public Test5()
    {
        i = 1;
    }
    public int Func1()
    {
        return 2 * i;
    }
    public void Func2() { }
}
```

3. 試寫出下列程式的輸出 (12%)

// 程式 CPFinal2012Problem3

```
using System;

namespace CPFinal2012Problem3
{
    class Program
    {
        static void Main(string[] args)
        {
            Item[] list = new Item[3];
            list[0] = new Book("001", "哈利波特:消失的密室",
                "J. K. 羅琳", "皇冠", 2001);
            list[1] = new Video("002", "變形金剛", "麥可貝",
                "席亞拉伯夫", "得利影視", 2009);
            list[2] = new Book( "003", "墨水心",
                "柯奈莉亞.馮克", "大田", 2005);
            foreach( Item item in list)
            {
                item.Dump();
            }
        }
    }

    abstract public class Item
    {
        protected string seqNo;
        protected string title;
        protected int year;
        public Item(string seqNo, string title, int year)
        {
            this.seqNo = seqNo;
            this.title = title;
            this.year = year;
        }
        abstract public void Dump();
    }

    public class Book : Item
    {
```

```
    private string majorAuthor;
    private string publisher;
    public Book(string seqNo, string title,
    string majorAuthor, string publisher, int year) :
        base(seqNo, title, year)
    {
        this.majorAuthor = majorAuthor;
        this.publisher = publisher;
    }
    public override void Dump()
    {
        Console.Write(seqNo + "\t");
        Console.Write(title + "\t");
        Console.Write(majorAuthor + "\t");
        Console.Write(publisher + "\t");
        Console.WriteLine(year);
    }
}

public class Video : Item
{
    private string majorDirector;
    private string majorActor;
    private string distributor;
    public Video(string seqNo, string title,
        string majorDirector, string majorActor,
        string distributor, int year)
        : base(seqNo, title, year)
    {
        this.majorDirector = majorDirector;
        this.majorActor = majorActor;
        this.distributor = distributor;
    }
    public override void Dump()
    {
        Console.Write(seqNo + "\t");
        Console.Write(title + "\t");
        Console.Write(majorDirector + "\t");
```

```
            Console.Write(majorActor + "\t");
            Console.Write(distributor + "\t");
            Console.WriteLine(year);
        }
    }
}
```

Ans.

4. 試寫出以下程式在下列狀況時的輸出

(a) (3%) 檔案 Test.dat 尚未建立

Ans.

(b) (3%) 檔案 Test.dat 已在正確位置，且內容爲

2
B645331 90
B645332 88

Ans.

(c) (3%) 檔案 Test.dat 已在正確位置，且內容爲

2
B645331 90
B645332 *8

Ans.

(d) (3%) 檔案 Test.dat 已在正確位置，且內容爲

```
2
B645331 102
B645332 88
```

Ans.


```
// 程式 CPFinal2012Problem4
using System;
using System.IO;


namespace CPFinal2012Problem4
{
    class Program
    {
        static void Main(string[] args)
        {
            string fileName = "Test.dat";
            int average = Test(fileName);
            Console.WriteLine("average = " + average);
        }
        static int Test(string fileName)
        {
            int result = 0;
            try
```

```
{
    StreamReader input = new
       StreamReader(fileName);
    Console.WriteLine("檔案開啓");

    try
    {
        int nDataPerLine =
          int.Parse(input.ReadLine());
        int i;
        string line;
        string[] dataString = new
           string[nDataPerLine];
        string regNo;
        int score;
        int sum = 0;
        int nStudents = 0;
        while (!input.EndOfStream)
        {
            line = input.ReadLine();
            dataString = line.Split(' ');
            regNo = dataString[0];
            score = int.Parse(dataString[1]);
            if(score < 0 || score > 100) throw
               new ScoreOutOfRangeException(
                  regNo, score);
            sum += score;
            nStudents++;
        }
        result = sum / nStudents;
    }
    catch (FormatException e)
    {
        Console.WriteLine("格式錯誤");
    }
    catch(ScoreOutOfRangeException e)
    {
        Console.WriteLine(e);
```

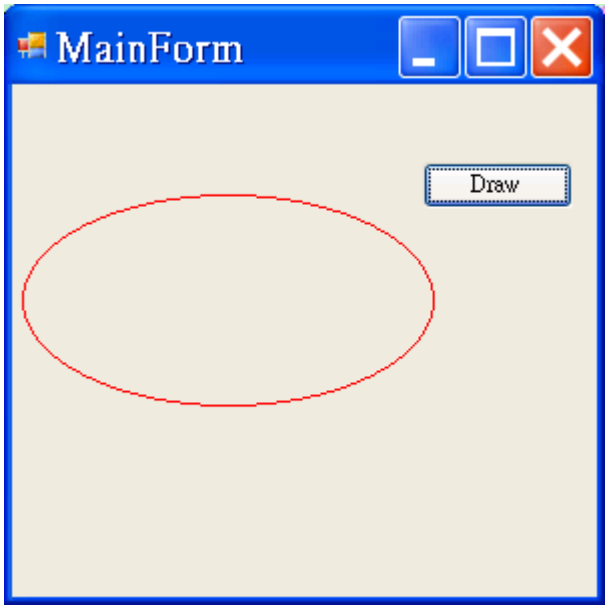
```
            }
            catch (Exception e)
            {
                Console.WriteLine(
                    "函式Test內層try-catch捕捉到例外");
                Console.WriteLine(
                    "函式Test內層try-catch再拋出例外");
                throw e;
            }
            finally
            {
                input.Close();
                Console.WriteLine("檔案關閉");
            }
        }
        catch (FileNotFoundException e)
        {
            Console.WriteLine("檔案不存在");
        }
        catch (Exception e)
        {
            Console.WriteLine(
                 "函式Test外層try-catch捕捉到例外");
        }
        return result;
    }
}

class ScoreOutOfRangeException : ApplicationException
{
    private string message;
    public ScoreOutOfRangeException(string regNo,
       int score : base()
    {
        message = "學生" + regNo + "之分數" + score +
        "不在0與100之間";
    }
    public override string ToString()
```

```
        {
            return message;
        }
    }
}
```

5. 依據以下描述及程式框架，完成指定程式。你在答案卷只需寫下程式註解標示的部份。(6%)

程式描述:
建立如下視窗介面，按下 Draw 按鈕，即於主視窗顯示一個紅色橢圓。假設 Draw 按鈕則設爲 Button 物件 button1，恰包住橢圓的矩形左上角座標爲(5,55), 右下角座標爲(205, 105)。不需要考慮 OnPaint 的問題。




```
// 檔案 MainForm.cs
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
```

```
using System.Windows.Forms;

namespace CPFinal2012Problem5
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender,
           EventArgs e)
        {
            //*********************************************
               在此加入必要之程式碼
            //*********************************************
        }
    }
}
```

Ans.

```
            Graphics g = this.CreateGraphics();
            g.DrawArc(new Pen(Color.Red), 5, 55, 200, 50,
                0, 360);
            g.Dispose();
```

6. 撰寫一個 C#程式以模擬神奇寶貝(Pocket Monster)運動會中的格鬥賽。爲簡化問題，只要寫主控台程式，並在螢幕顯示每一回合神奇寶貝之行動與體力值，最後顯示先使對方體力下降到 0(含)以下的贏家即可。假設參賽的神奇寶貝有皮卡丘(Pikachiu)

 、妙蛙種子(**Bulbasaur**)  、傑尼龜(**Squirtle**)  ，

由使用者任意挑選兩隻出賽，每回合隨機決定由那一隻神奇寶貝攻擊(attack)，另一隻防禦(defense)。開始時各種神奇寶貝的體力值(hp)均爲 15；攻擊時，使對方

體力下降的數值(稱爲攻擊力 attack strength)如下：

皮卡丘： **`rand.Next() % 16 + 1`**

妙蛙種子：**`(rand1.Next() % 10) + (rand2.Next() % 5) + 1`**
(需要兩個亂數產生器)

傑尼龜： **`(rand.Next() % 20 + 5) % 10  + 1`**

而防禦時，可減少對方攻擊造成之體力損失(稱爲防禦力 `defense strength`)如下：

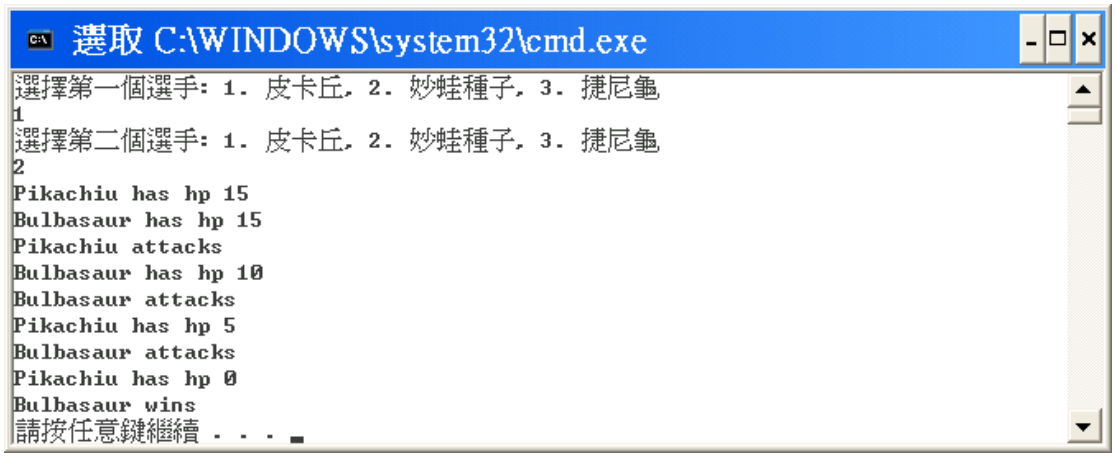
皮卡丘：**`(rand.Next() % 20 + 5) % 10  + 1`**

妙蛙種子：**`(rand1.Next() % 8) + (rand2.Next() % 3) + 1`**

傑尼龜：**`rand.Next() % 16 + 1`**

如果自身防禦力高於對方的攻擊力，則體力值不變。同一隻神奇寶貝攻擊和防禦共同使用一組亂數產生器。

其輸出可能如下圖(殘念!! 皮卡丘先盛後衰，輸了！好討厭的感覺呀!)：



以上述之測試場景撰寫程式，不需撰寫額外內容。不使用類別者，最高得 13 分；使用類別，不使用多型者，最高得 20 分；正確使用類別及多型者，最高得 25 分。
(25%)

Ans.

```
using System;

namespace CPFinal2012Problem6
{
    class Program
    {
        static void Main(string[] args)
        {
            const int N_MONSTERS = 3;
            PocketMonster[] monsters = new
              PocketMonster[N_MONSTERS];
            monsters[0] = new Pikachiu(168);
            monsters[1] = new Bulbasaur(777, 545);
            monsters[2] = new Squirtle(66);
            Console.WriteLine("選擇第一個選手: 1. 皮卡丘,
               2. 妙蛙種子, 3. 捷尼龜");
            int i = int.Parse(Console.ReadLine());
            PocketMonster fighter1 = monsters[i - 1];
            Console.WriteLine("選擇第二個選手: 1. 皮卡丘,
               2. 妙蛙種子, 3. 捷尼龜");
            i = int.Parse(Console.ReadLine());
            PocketMonster fighter2 = monsters[i - 1];
            Console.WriteLine("{0} has hp {1}", fighter1.Name,
              fighter1.HP);
            Console.WriteLine("{0} has hp {1}", fighter2.Name,
              fighter2.HP);
            Random rand = new Random();
            while (fighter1.HP > 0 && fighter2.HP > 0)
            {
               if (rand.Next() % 2 == 0)
               {
                  Fight(fighter1, fighter2);
                  if (fighter2.HP <= 0)
                  {
                     Console.WriteLine(fighter1.Name +
                       " wins");
                     break;
                  }
```

```
            }
            else
            {
                Fight(fighter2, fighter1);
                if (fighter1.HP <= 0)
                {
                    Console.WriteLine(fighter2.Name +
                      " wins");
                    break;
                }
            }
        }

    }
    static void Fight(PocketMonster attacker,
                      PocketMonster defender)
    {
        Console.WriteLine(attacker.Name+" attacks");
        int attackStrength = attacker.AttackStrength();
        defender.Defend(attackStrength);
        Console.WriteLine("{0} has hp {1}", defender.Name,
          defender.HP);
    }
}
abstract class PocketMonster
{
    private string name;
    protected int hp = 15;
    public PocketMonster(string name)
    {
        this.name = name;
    }
    public string Name
    {
        get { return name; }
    }
    public int HP
    {
```

```
        get { return hp; }
    }
    abstract public int AttackStrength();
    abstract public void Defend(int attackStrength);
}

class Pikachiu : PocketMonster
{
    private Random rand;
    public Pikachiu(int seed)
        : base("Pikachiu")
    {
        rand = new Random(seed);
    }
    public override int AttackStrength()
    {
        return rand.Next() % 16 + 1;
    }
    public override void Defend(int attackStrength)
    {
        int defenseStrength = (rand.Next() % 20 + 5) % 10
          + 1;
        if (attackStrength > defenseStrength)
        {
            hp -= (attackStrength - defenseStrength);
        }
    }
}

class Bulbasaur : PocketMonster
{
    private Random rand1;
    private Random rand2;
    public Bulbasaur(int seed1, int seed2)
        : base("Bulbasaur")
    {
        rand1 = new Random(seed1);
        rand2 = new Random(seed2);
```

```
    }
    public override int AttackStrength()
    {
        return (rand1.Next() % 10) + (rand2.Next() % 5) +
               1;
    }
    public override void Defend(int attackStrength)
    {
        int defenseStrength = (rand1.Next() % 8) +
          (rand2.Next() % 3) + 1;
        if (attackStrength > defenseStrength)
        {
            hp -= (attackStrength - defenseStrength);
        }
    }
}

class Squirtle : PocketMonster
{
    private Random rand;
    public Squirtle(int seed)
        : base("Squirtle")
    {
        rand = new Random(seed);
    }
    public override int AttackStrength()
    {
        return (rand.Next() % 20 + 5) % 10 + 1;
    }
    public override void Defend(int attackStrength)
    {
        int defenseStrength = rand.Next() % 16 + 1;
        if (attackStrength > defenseStrength)
        {
            hp -= (attackStrength - defenseStrength);
        }
    }
}
```

```
}
```