

## 通識計算機程式設計期末考試題參考解答, 6/21/2007

1.

- (a) 撰寫類別程式 `Ellipse` 代表橢圓，其中宣告私有 `double` 變數 `a`、`b` 分別代表半長軸與半短軸，另有建構式設定 `a`、`b` 之值，屬性 `accessMajor`、`accessMinor` 分別傳回與設定 `a`、`b` 之值 (9%)

答：

```
public class Ellipse
{
    private double a; // 半長軸
    private double b; // 半短軸
    public Ellipse(double a, double b)
    {
        this.a = a;
        this.b = b;
    }
    public double accessMajor
    {
        get { return a; }
        set { a = value; }
    }
    public double accessMinor
    {
        get { return b; }
        set { b = value; }
    }
}
```

- (b) 撰寫類別程式 `Circle` 代表圓，繼承 `Ellipse`，在傳入半徑參數 `r` 之建構式中呼叫父類別建構式設定 `a`、`b` 之值為 `r`，增加屬性 `accessR` 傳回 `a` 與設定 `a`、`b` 值 (6%)

答：

```
public class Circle : Ellipse
{
    public Circle(double r) : base( r, r ) {}
    public double accessR
    {
```

```

        get { return base.accessMajor; }
        set
        {
            base.accessMajor = value;
            base.accessMinor = value;
        }
    }
}

```

- (c) 撰寫抽象類別 `Shape`，其中定義公有抽象函式 `GetArea()`，用以計算面積，傳回 `double` 數值 (3%)

答：

```

abstract class Shape
{
    public abstract double GetArea();
}

```

- (d) 改寫(a)的類別程式 `Ellipse`，繼承 `Shape`，並覆寫(override)函式 `GetArea()` (3%) [注意橢圓面積為 $\pi ab$ ]

答：

```

public class Ellipse : Shape
{
    private double a; // 半長軸
    private double b; // 半短軸
    public Ellipse(double a, double b)
    {
        this.a = a;
        this.b = b;
    }
    public double accessMajor
    {
        get { return a; }
        set { a = value; }
    }
    public double accessMinor
    {
        get { return b; }
        set { b = value; }
    }
}

```

```

        }
    public override double GetArea()
    {
        return Math.PI*a*b;
    }
}

```

- (e) 在類別程式 `Ellipse` 的建構式與屬性中，檢查輸入參數是否不大於 0，  
若此種狀況發生，`throw` 一個 `ArgumentException` 例外物件 (3%)

答：

```

public class Ellipse : Shape
{
    private double a; // 半長軸
    private double b; // 半短軸
    public Ellipse(double a, double b)
    {
        if (a <= 0 || b <= 0)
            throw new ArgumentException(
                "半長軸與半短軸須為正數");
        this.a = a;
        this.b = b;
    }
    public double accessMajor
    {
        get { return a; }
        set
        {
            if (value <= 0 )
                throw new ArgumentException(
                    "半長軸須為正數");
            a = value;
        }
    }
    public double accessMinor
    {
        get { return b; }
        set
        {

```

```

        if (value <= 0)
            throw new ArgumentException(
                "半短軸須為正數");
        b = value;
    }
}
public override double GetArea()
{
    return Math.PI*a*b;
}
}

```

- (f) 寫一段測試程式，自鍵盤讀入 a、b 之值，再以之建立一個橢圓物件 e。  
 用 try-catch 擋截 FormatException、ArgumentException、  
 Exception 例外物件，並於例外發生時，要求使用者重新輸入，直到  
 正確為止 (6%)

答：

```

class Program
{
    static void Main(string[] args)
    {
        bool failed = true;
        do
        {
            try
            {
                Console.WriteLine("輸入橢圓半長軸a");
                double a =
                    double.Parse(Console.ReadLine());
                Console.WriteLine("輸入橢圓半短軸b");
                double b =
                    double.Parse(Console.ReadLine());
                Ellipse e = new Ellipse(a, b);
                Console.WriteLine("橢圓e半長軸a = " +
                    e.accessMajor);
                Console.WriteLine("橢圓e半短軸b = " +
                    e.accessMinor);
                failed = false;
            }
            catch (FormatException)
            {
                Console.WriteLine("輸入格式錯誤，請重新輸入");
            }
            catch (ArgumentException)
            {
                Console.WriteLine("半長軸或半短軸必須為正數，請重新輸入");
            }
            catch (Exception)
            {
                Console.WriteLine("未知錯誤，請重新輸入");
            }
        }
        while (failed);
    }
}

```

```

        }
        catch (FormatException ex)
        {
            Console.WriteLine(ex);
        }
        catch (ArgumentException ex)
        {
            Console.WriteLine(ex);
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex);
        }
    } while (failed);
}
}

```

2. 找出以下程式片段之錯誤，並予更正.

(a) (6%) 兩個錯誤

```

Test t = Test(1); ➔ Test t = new Test(1);
Console.WriteLine(t.accessI);

. . . . .

public class Test {
    int i;
    public Test(int i) {
        this.i = i;
    }
    protected int accessI { ➔ public int accessI {
        get { return i; }
    }
}

```

(b) (3%) 一個錯誤

```

interface Shape {
    public double GetArea(); ➔ double GetArea();
}

```

(c) (3%).兩組錯誤

```
public abstract class Bird { ➔ interface Bird {  
    public virtual void Fly(); ➔ void Fly();  
}  
  
public abstract class Beast { ➔ interface Beast {  
    public virtual void HasFur(); ➔ void HasFur();  
}  
  
public class Bat : Bird, Beast {  
    public Bat() {  
        Console.WriteLine("蝙蝠");  
    }  
    public void HasFur() {  
        Console.WriteLine("棕毛");  
    }  
    public void Fly() {  
        Console.WriteLine("飛行");  
    }  
}
```

(d) (3%) 一個錯誤

```
double a = 5.0;  
Square sq0 = new Square(a);  
Square sq1 = sq0; ➔ Square sq1 = new Square( sq0 );  
sq0.accessA = a + 1.0; // 不應改變 sq1 的內容  
. . . . .  
public class Square {  
    private double a;  
    public Square(double a) {  
        this.a = a;  
    }  
    public double accessA  
    {  
        get { return a; }  
        set { a = value; }  
    }  
}
```

```

    }
    public Square(Square sq){ // 複製建構式
        a = sq.a;
    }

}

```

(e) (3%) 一個錯誤

```

public class Test {
    int i;
    public Test(int i) {
        this.i = i;
    }
    public static bool operator >(Test op1,
        Test op2) {
        bool result = (op1.i > op2.i);
        return result;
    }
    → 增加
    public static bool operator <(Test op1,
        Test op2) {
        bool result = (op1.i < op2.i);
        return result;
    }
}

```

3. 試寫出下列程式的輸出 (12%)

```

using System;
namespace Final2007
{
    class Program
    {
        static void Main(string[] args)
        {
            Shape shape = new Shape();
            shape.Draw();
            System.Console.WriteLine("-----");
        }
    }
}

```

```
        Square sq = new Square();
        sq.Draw();
        System.Console.WriteLine("-----");

        Circle c = new Circle();
        c.Draw();
        System.Console.WriteLine("-----");

        Shape figure;

        figure = shape;
        figure.Draw();
        System.Console.WriteLine("-----");

        figure = sq;
        figure.Draw();
        System.Console.WriteLine("-----");

        figure = c;
        c.Draw();
        System.Console.WriteLine("-----");

        test(shape);
        test(sq);
        test(c);

        Shape[] shapes = new Shape[3];
        shapes[0] = new Shape();
        shapes[1] = new Square();
        shapes[2] = new Circle();
        foreach (Shape sh in shapes)
        {
            sh.Draw();
            System.Console.WriteLine("-----");
        }
    }
```

```
static void test(Shape sh)
{
    sh.Draw();
    System.Console.WriteLine("-----");
}
}

class Shape
{
    public virtual void Draw()
    {
        System.Console.WriteLine(
            "Drawing prepared");
    }
}

class Square : Shape
{
    public new virtual void Draw()
    {
        base.Draw();
        System.Console.WriteLine(
            "A square is drawn");
    }
}

class Circle : Shape
{
    public override void Draw()
    {
        base.Draw();
        System.Console.WriteLine(
            "A circle is drawn");
    }
}
```

答：

```
Drawing prepared
-----
Drawing prepared
A square is drawn
-----
Drawing prepared
A Circle is drawn
-----
Drawing prepared
-----
Drawing prepared
-----
Drawing prepared
A Circle is drawn
-----
Drawing prepared
-----
Drawing prepared
-----
Drawing prepared
A Circle is drawn
-----
Drawing prepared
-----
Drawing prepared
A Circle is drawn
-----
Drawing prepared
```

4. 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為(a)到(d)的部份。各部份配分分別註明，總共 15%。

程式描述：用類別 Matrix 來計算矩陣乘法  $c_{ij} = \sum_{k=1}^q a_{ik} * b_{kj}$  ,  $1 \leq i \leq m$  ,  
 $1 \leq j \leq n$  ,  $1 \leq k \leq q$

```

using System;

namespace MatrixMultiplication
{
    class Program
    {
        static void Main(string[] args)
        {
            Matrix a = new Matrix(2, 3);
            a[1, 1] = 10;
            a[2, 1] = 11;
            a[1, 2] = 11;
            a[2, 2] = 12;
            a[1, 3] = 12;
            a[2, 3] = 13;
            a.print();

            Matrix b = new Matrix(3, 2);
            b[1, 1] = 1;
            b[2, 1] = 2;
            b[3, 1] = 3;
            b[1, 2] = 2;
            b[2, 2] = 3;
            b[3, 2] = 4;
            b.print();

            Matrix c = a * b;
            c.print();
        }
    }

    public class Matrix
    {
        int nRow;
        int nCol;
        int[,] array;

        public Matrix(int nRow, int nCol)
        {
            /* (a) 加入敘述完成建構式，不必處理例外狀況 (3%) */
        }
    }
}

```

```

}

public int this[int i, int j]
{
    get
    {
        /* (b) 加入敘述完成屬性回傳，不必處理例外狀況 (3%) */
    }
    set
    {
        /* (c) 加入敘述完成屬性設定，不必處理例外狀況 (3%) */
    }
}

public void print()
{
    Console.WriteLine("-----");
    int i;
    int j;
    for(i=1; i<=nRow; ++i)
    {
        for(j=1; j<=nCol; ++j)
        {
            Console.Write( array[i-1, j-1] + "\t" );
        }
        Console.WriteLine();
    }
}

/* (d) 撰寫運算元*函式，計算兩矩陣乘積，傳回結果矩陣，  

   不必處理例外狀況 (6%) */

}
}
}

```

答：

(a)

```

this.nRow = nRow;
this.nCol = nCol;
array = new int[nRow, nCol];

```

```

(b)
    return array[i - 1, j - 1];
(c)
    array[i - 1, j - 1] = value;

(d)
public static Matrix operator *(Matrix op1, Matrix op2)
{
    Matrix result = new Matrix(op1.nRow, op2.nCol);
    int i;
    int j;
    int k;
    for (i = 1; i <= op1.nRow; ++i)
    {
        for (j = 1; j <= op2.nCol; ++j)
        {
            result[i, j] = 0;
            for (k = 1; k <= op1.nCol; ++k)
            {
                result[i, j] += op1[i, k] * op2[k, j];
            }
        }
    }
    return result;
}

```

5. 繼承 `System.Collections` 內的 `DictionaryBase` 寫電話號碼簿類別 `PhoneBook`。在測試主程式中建立如下的電話號碼簿 `pb`，然後以”李四”為 key，以索引子`[]`印出對應的電話號碼。在 `PhoneBook` 類別程式中只要完成索引子`[]`、及函式 `Add`、`Contains`、`Remove` 即可，不須處理例外狀況。

key	value
趙大	(02)77777777
錢二	(02)56565656
孫三	(039)333444
李四	(04)666789
周五	(02)78978922
吳六	(02)74774755

(25%)

答：

```
using System;
using System.Collections;

namespace UsingPhoneBook
{
    class Program
    {
        static void Main(string[] args)
        {
            PhoneBook pb = new PhoneBook();

            pb.Add("趙大", "(02)77777777");
            pb.Add("錢二", "(02)56565656");
            pb.Add("孫三", "(039)333444");
            pb.Add("李四", "(04)666789");
            pb.Add("周五", "(02)789789");
            pb.Add("吳六", "(02)74774755");
            string key = "李四";
            if( pb.Contains( key ) )
            {
                Console.WriteLine(pb[key]);
            }
        }
    }

    class PhoneBook : DictionaryBase
    {
        public String this[String key]
        {
            get
            {
                return ((String)Dictionary[key]);
            }
            set
            {
                Dictionary[key] = value;
            }
        }
    }
}
```

```
public void Add(string key, string value)
{
    Dictionary.Add(key, value);
}

public bool Contains(string key)
{
    return (Dictionary.Contains(key));
}

public void Remove(string key)
{
    Dictionary.Remove(key);
}
}
```