단국대학교 멀티미디어공학 HCI 프로그래밍 II 중간고사 (2015년 가을학기) 2015년 10월 20일

학과 _____ 학번 _____ 이름 _____

중간고사

담당교수: 단국대학교 멀티미디어공학전공 박경신

- 답은 반드시 답안지에 기술할 것. 공간이 부족할 경우 반드시 답안지 몇 쪽의 뒤에 있다고 명기한 후 기술할 것. 그 외의 경우의 답안지 뒤쪽이나 연습지에 기술한 내용은 답안으로 인정 안 함. 답에는 반드시 네모를 쳐서 확실히 표시할 것.
- 답안지에 학과, 학번, 이름 외에 본인의 암호(4자리 숫자)를 기입하면 성적공고시 학번대신 암호를 사용할 것임.

1. HW1 (도형의 영역 계산기) 프로그램 일부이다. 다음 물음에 답하라. (35점)

```
enum FigureType { Triangle, Square, Rectangle }
class AreaCalculator
{
        double bottom=1.0;
        double side=2.0;
        double width=3.0, height=4.0;

        // 사용자 입력에 따른 FigureType 반환
        public FigureType? GetFigureTypeFromString(string input) {
            // 내부 구현 필요
        }
        // 도형 영역(Area) 계산은 삼각형(bottom, height), 정사각형(side), 직사각형(width, height) 사용
        public double CalculateArea(FigureType type) {
            // 내부 구현 필요
        }
    }
}

class Program
{
        static FigureType[] figures = (FigureType[]) FigureType.GetValues(typeof(FigureType));

        static void Main(string[ ] args) {
            AreaCalculator calc = new AreaCalculator ();
            FigureType? fig = calc.GetFigureTypeFromString(Console.ReadLine());
            Console.WriteLine(fig);
            foreach (var f in figures)
                Console.WriteLine("{0} : {1}", f, calc.CalculateArea(f));
        }
}
```

1.1. Main 메소드에 밑줄 친 부분의 실행 결과를 자세히 설명하라. (5점)

FigureType의 이름과 그 도형의 영역을 계산하여 출력하는 것으로, 출력 결과는 다음과 같다.
Triangle : 2.0
Square : 4.0
Rectangle : 12.0

학과 _____ 학번 _____ 이름 _____

1.2. 1.1과 동일한 결과가 출력될 수 있도록 밀줄 친 부분을 while/do while 로 구현하라. (5점)

```
int i = 0; // do..while 또는 while
do {
    Console.WriteLine("{0} : {1}", figures[i], calc.CalculateArea(figures[i]));
    i++;
} while (i < figures.Length);
```

1.3. 사용자 string 입력에서 도형의 종류 enum 값을 반환하는 public FigureType? GetFigureTypeFromString(string input) 를 switch 구문을 사용하여 구현하라. (10점)

```
public FigureType? GetFigureTypeFromString(string input) {
    switch (input) {
        case "삼각형":
        case "Triangle":
            return FigureType.Triangle;
        case "정사각형":
        case "Square":
            return FigureType.Square;
        case "직사각형":
        case "Rectangle":
            return FigureType.Rectangle;
        default:
            return null;
    }
}
```

1.4. 1.3과 동일하게 public FigureType? GetFigureTypeFromString(string input) 를 if/elseif 구문을 사용하여 구현하라. (10점)

```
public FigureType? GetFigureTypeFromString(string input) {
    if (input == "삼각형" || input == "Triangle")
            return FigureType.Triangle;
    else if (input == "정사각형" || input == "Square")
            return FigureType.Square;
    else if (input == "직사각형" || input == "Rectangle")
            return FigureType.Rectangle;
    else
            return null;
}
```

학과 _____ 학번 _____ 이름 _____

1.5. 도형에 따라 영역(Area)을 계산하는 public double CalculateArea(FigureType type) 를 구현하라. (5점)

```
public double CalculateArea(FigureType type) { // if/elseif 또는 switch 구문
    if (type == FigureType.Triangle) return bottom * height * 0.5;
    else if (type == FigureType.Square) return side * side;
    else if (type == FigureType.Rectangle) return width * height;
    else return 0.0;
}
```

2. 다음은 Point 클래스 예제 프로그램이다. 다음 질문에 답하시오. (50점)

$$Magnitude = \sqrt{x^2 + y^2}$$

$$Distance = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

```
class Point {
        public static readonly Point Zero = new Point(0, 0);
        public static readonly Point One = new Point(1, 1);
        public int X { get; set; }
        public int Y { get; set; }
        public Point() : this(0, 0) { }
        public Point(int x, int y) { X = x; Y = y; }
        public virtual void Print() { Console.WriteLine("Point ({0} {1})", X, Y); }
        public override string ToString() { return string.Format("Point ({0} {1})", X, Y); }
        public double Magnitude { // 원점으로부터 거리
                get { return Math.Sqrt(X*X + Y*Y); }
        }
        public static double Distance(Point p, Point q) { // 두 점 간의 거리
                return Math.Sqrt((p.X – q.X)*(p.X – q.X) + (p.Y – q.Y)*(p.Y – q.Y));
        }
        public virtual double Distance(Point other) { // 두 점 간의 거리
                return Distance(this, other);
        }
        public virtual void Set(Point p) {
                X = p.X; Y = p.Y;
                Console.WriteLine("Set(Point): " + ToString());
        }
        public void Set() {
                X = 10; Y = 10;
                Console.WriteLine("Set(): " + ToString());
        }
        public Point Set(int i) {
                Point p = new Point(i, i);
                Console.WriteLine("Set(i): " + p.ToString());
                return p;
        }
        public void Set(int i, out Point p) {
                p = new Point(i, i);
                Console.WriteLine("Set(i, out Point): " + p.ToString());
        }
        public static void Set(ref Point p) {
                p.X = 10; p.Y = 10;
                Console.WriteLine("Set(ref Point): " + p.ToString());
        }
}
```

학과 _____ 학번 _____ 이름 _____

```
class Point3D: Point {
	public new static readonly Point3D Zero = new Point3D(0, 0, 0);
	public new static readonly Point3D One = new Point3D(1, 1, 1);
	public int Z {  get;  set;  }
	public Point3D() : this(0, 0, 0) { }
	public Point3D(int x, int y, int z) : base(x, y) {  Z = z;  }
	public override void Print() { Console.WriteLine("Point3D ({0} {1} {2})", X, Y, Z);  }
	public override string ToString() { return string.Format("Point3D ({0} {1} {2})", X, Y, Z);  }
	public new double Magnitude { // 원점으로부터 거리
		get {
			return Math.Sqrt(X*X + Y*Y + Z*Z);
		}
	}
	public static double Distance(Point3D p, Point3D q) {
		return Math.Sqrt((p.X – q.X)*(p.X – q.X) + (p.Y – q.Y)*(p.Y – q.Y) + (p.Z – q.Z)*(p.Z – q.Z));
	}
	public double Distance(Point3D other) { // 두 점 간의 거리
		return Distance(this, other);
	}
	public override double Distance(Point other) { // 두 점 간의 거리
		if (other is Point3D) return Distance((Point3D)other);
		else return base.Distance(other);
	}
	public void Set(Point3D p) {
		X = p.X; Y = p.Y; Z = p.Z;
		Print();
	}
	public override void Set(Point p) {
		if (p is Point3D) Set((Point3D)p);
		else base.Set(p);
	}
}
```

2.1 Point 클래스의 Magnitude와 Distance(Point other)를 구현하라. (5점)

```
public virtual double Magnitude { // 원점으로부터 거리
	get {
		return Math.Sqrt(X*X + Y*Y);
	}
}
public double Distance(Point other) { // 두 점 간의 거리
	return Distance(this, other);
	//또는 return Math.Sqrt((X – other.X)*(X – other.X) + (Y – other.Y)*(Y – other.Y));
}
```

2.2 Point3D 클래스의 Distance(Point3D other)와 Distance(Point other)를 구현하라. (5점)

```
public double Distance(Point3D other) { // 두 점 간의 거리
	return Distance(this, other);
}
public override double Distance(Point other) { // 두 점 간의 거리
	if (other is Point3D) return Distance((Point3D)other);
	else return base.Distance(other);
}
```

학과 _____ 학번 _____ 이름 _____

```
class PointTest {
        static void Main(string[ ] args) {
            Point p = new Point(5, 5);
            Console.WriteLine("p: " + p);
            p.Set();                                // (1)
            Console.WriteLine("p: " + p);
            Point q = p.Set(5);                     // (2)
            Console.WriteLine("p: " + p);
            Console.WriteLine("q: " + q);
            p.Set(q);                               // (3)
            Console.WriteLine("p: " + p);
            Console.WriteLine("q: " + q);
            Point r;
            p.Set(1, out r);                        // (4)
            Console.WriteLine("p: " + p);
            Console.WriteLine("r: " + r);
            Point.Set(ref r);                       // (5)
            r.Set(1);                               // (6)
            Console.WriteLine("p: " + p);
            Console.WriteLine("r: " + r);
        }
}
```

2.3 PointTest의 실행결과를 자세히 나타내라. 실행결과에 Set 메소드 호출 번호도 표시할 것. Set 메소드 호출 시 parameter passing에 유의할 것. (10점)

```
p: (5, 5)
Set(): (10, 10)                 // (1)
p: (10, 10)
Set(i): (5, 5)                  // (2)
p: (10, 10)
q: (5, 5)
Set(Point): (5, 5)              // (3)
p: (5, 5)
q: (5, 5)
Set(i, out Point): (1, 1)       // (4)
p: (5, 5)
r: (1, 1)
Set(ref Point): (10, 10)        // (5)
Set(i): (1, 1)                  // (6)
p: (5, 5)
r: (10, 10)
```

2.4 위의 예제에서 method overloading과 overriding을 예로 들어 설명하라. (5점)

Method overloading은 동일한 함수명에 다양한 파라메터와 리턴을 정의하는 것으로
Point클래스의 void Set(Point), void Set(), Point Set(i), void Set(i, out Point), void Set(ref Point)와
Point3D 클래스의 void Set(Point3D)가 있다.
double Distance(Point), double Distance(Point3D), double Distance(Point, Point)가 있다.

Method overriding은 상속받은 파생클래스에서 동일한 함수명에 동일한 매개변수 정의하여 함수를 재정의하는 것으로 상속되어진 함수의 기능을 변경해서 재사용하고 싶은 때 사용하는 것으로
Point와 Point3D 클래스에 동일한 함수를 재정의해서 사용한 void Set(Point p),
double Distance(Point other), string ToString(), void Print()가 있다.

학과 _____ 학번 _____ 이름 _____

```
class PointTest2 {
        static void Main(string[ ] args) {
            Point[] points = new Point[4];
            points[0] = Point.One;
            points[1] = new Point(4, 5);
            points[2] = Point3D.Zero;
            points[3] = new Point3D(3, 4, 5);
            foreach (var i in points) i.Print(); // (1)
            foreach (var i in points) Console.WriteLine(i.Magnitude); // (2)

            double[] lengths = new double [5];
            lengths[0] = points[0].Distance(points[1]);
            lengths[1] = Point.Distance(points[0], points[1]);
            lengths[2] = Point3D.Distance(points[2], points[3]);
            lengths[3] = Point3D.Distance((Point3D)points[2], (Point3D)points[3]);
            lengths[4] = points[2].Distance(points[3]);
            for (int i = 0; i < lengths.Length; i++) Console.WriteLine("len[{0}] = {1}", i, lengths[i]); // (3)
        }
}
```

2.5 PointTest2의 실행결과를 자세히 나타내라. 실행결과에 sqrt를 사용해서 숫자나 계산으로 적어줄 것. Polymorphism에 유의할 것. (10점)

```
(1, 1)                                  // (1) (3점)
(4, 5)
(0, 0, 0)
(3, 4, 5)
1.4142              // sqrt(2=1+1)      // (2) (3점)
6.4031              // sqrt(41=16+25)
0                   // sqrt(0=0+0)
5                   // sqrt(25=9+16)
len[0] = 5          // sqrt(25=9+16)    // (3) (4점)
len[1] = 5          // sqrt(25=9+16)
len[2] = 5          // sqrt(25=9+16)
len[3] = 7.071      // sqrt(50=9+16+25)
len[4] = 7.071      // sqrt(50=9+16+25)
```

2.6 Point3D 클래스의 Magnitude는 new 키워드를 사용하였다. new 키워드의 사용 용도는 무엇인가? Magnitude 경우 이로 인한 문제점과 이를 해결하는 방법을 자세히 설명하라. (10점)

상속관계에서 부모클래스와 자식클래스에서 동일한 이름을 사용하는 경우,
new 키워드를 사용하여 명시적으로 명확하게 구분하기 위하여 메소드를 새정의(new) 한다.
Point.One과 Point3D.One경우 이름은 같아도 각각 다른 상수를 제공해야하기 때문에 new가 필요
그런데 PointTest2에서 보이듯이 Magnitude 경우, new를 사용했기때문에 객체의 타입에 따라
메소드가 호출되므로 points[2].Magnitude와 points[3].Maginitude는 Point.Magnitude가 호출되는
문제가 발생한다.
따라서 Magnitude 도 void Print()나 void Set(Point)와 같이 virtual/override를 사용하여
late-binding이 가능하도록 해야 한다.

2.7 Point와 Point3D 클래스의 static readonly 변수의 사용 용도를 예를 들어 자세히 설명하라. (5점)

PointTest2에서 보이듯이, points[0] = Point.One를 사용하여 Point (1,1) 객체 생성
마찬가지로 points[2] = Point3D.One을 사용하여 Point3D (1,1,1) 객체 생성
static readonly를 사용하여 상수를 지정해놓고, <클래스명>.<readonly 상수명>을 이용하여 호출

학과 _____ 학번 _____ 이름 _____

3. Figure 추상 클래스와 IFigure 인터페이스를 사용하여 둘레를 계산하는 프로그램이다. 다음 실행 결과와 동일하게 나타나도록 (빈칸 안에) 프로그램을 작성하라. (15점)

```
Red Triangle 10 + 10 + 10 = 30
Blue Rectangle (10 + 20) x 2 = 60
Green Square 20 x 4 = 80
Magenta Trapezoid 7 + 10 + 7 + 6 = 30
Cyan Parallelogram (5 + 7) x 2 = 24
Yellow Rhombus 30 x 4 = 120
```

```
enum FigureType { Triangle, Rectangle, Square, Parallelogram, Rhombus, Trapezoid }

interface IFigure {
        FigureType Type {
            get;
        }
        ConsoleColor Color {
            get;
        }
}

abstract class Figure : IFigure { // (5점)
        protected FigureType type;
        protected ConsoleColor color;
        protected double [ ] sides;
        public FigureType Type {
            get { return type; }
        }
        public ConsoleColor Color {
            get { return color; }
        }
        public abstract double Perimeter {
            get;
        }
        public abstract string PerimeterFormula {
            get;
        }
         public override string ToString() {
            return string.Format("{0} {1} {2} {3}", color, type, PerimeterFormula, Perimeter);
        }
}

class Triangle : Figure { // (5점)
        public Triangle() {
            type = FigureType.Triangle;
            color = ConsoleColor.Red;
            sides = new double[3] { 10, 10, 10 }; // 삼각형의 세 변은 10, 10, 10
        }
        public override double Perimeter { // 삼각형 둘레는 세 변의 길이 합
            get { return sides[0] + sides[1] + sides[2]; }
        }
        public override string PerimeterFormula { // 삼각형 둘레 공식 출력
            get { return sides[0] + " + " + sides[1] + " + " + sides[2] + " = "; }
        }
}
```

학과 _____ 학번 _____ 이름 _____

```
class Rectangle : Figure {
    public Rectangle () {
        type = FigureType.Rectangle;
        color = ConsoleColor.Blue;
        sides = new double [2] { 10, 20 };
    }
    public override double Perimeter { // 직사각형 둘레는 (가로 + 세로) x 2
        get { return (sides[0] + sides[1]) * 2; }
    }
    public override string PerimeterFormula { // 직사각형 둘레 공식 출력
        get { return "(" + sides[0] + " + " + sides[1] + ") x 2 ="; }
    }
}
class Square : Rectangle {
    public Square() {
        type = FigureType.Square;
        color = ConsoleColor.Green;
        sides = new double [1] { 20 };
    }
    public override double Perimeter { // 정사각형 둘레는 한 변 x 4
        get { return sides[0] * 4; }
    }
    public override string PerimeterFormula { // 정사각형 둘레 공식 출력
        get { return sides[0] + " x 4 ="; }
    }
}
class Trapezoid : Rectangle {
    public Trapezoid () {
        type = FigureType.Trapezoid;
        color = ConsoleColor.Magenta;
        sides = new double [4] { 7, 10, 7, 6 };
    }
    public override double Perimeter { // 사다리꼴 둘레는 네 변의 합
        get { return sides[0] + sides[1] + sides[2] + sides[3]; }
    }
    public override string PerimeterFormula { // 사다리꼴 둘레 공식 출력
        get { return sides[0] + " + " + sides[1] + " + " + sides[2] + " + " + sides[3] + " = "; }
    }
}
class Parallelogram : Rectangle { // Rectangle의 Perimeter 사용 가능
    public Parallelogram() {
        type = FigureType.Parallelogram;
        color = ConsoleColor.Cyan;
        sides = new double [2] { 5, 7 };
    }
}
class Rhombus : Square { // Square의 Perimeter 사용 가능
    public Rhombus () {
        type = FigureType.Rhombus;
        color = ConsoleColor.Yellow;
        sides = new double [1] { 30 };
    }
}
```

단국대학교 멀티미디어공학 HCI 프로그래밍 II 중간고사 (2015년 가을학기) 2015년 10월 20일

학과 _____ 학번 _____ 이름 _____

```
class Program {
        static void Main(string[] args) { // (5점)
            Figure [ ] aList= new Figure [6] {
                                    new Triangle(),
                                    new Rectangle(),
                                    new Square(),
                                    new Trapezoid(),
                                    new Parallelogram(),
                                    new Rhombus() };
            foreach (var elem in aList) {
                Console.WriteLine(elem);
            }
        }
}
```

-끝-