단국대학교 멀티미디어공학 HCI 프로그래밍 II 중간고사 (2016년 가을학기) 2016년 10월 24일

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. 다음은 Person, Student 클래스 예제 프로그램이다. 다음 질문에 답하시오. (75점)

```
class Person {
        private static int count = 0;
        public static int Count {   get { return count; }   }
        private string name;
        public string Name { get { return name; } set { name = value; } }
        private int age;
        public int Age { get { return age; } set { age = value; } }
        protected Person() : this("" , 0) {   }
        public Person(string name, int age) { Set(name, age); count++ }
        public virtual void Set(string name, int age) { this.name = name; this.age = age; }
        public virtual void Set(Person other) { Set(other.name, other.age); }
        public void Set(ref Person p) {
                p.name = "HCI"; p.age = 10;
        }
        public void Set(string name, int age, out Person p) {
                p = new Person(name, age);
        }
        public void Print() { Console.WriteLine("Person Name: {0} Age: {1}", name, age);   }
        public override string ToString() { return string.Format("Person Name: {0} Age: {1}", name, age);   }
}
class Student: Person {
        private static int count = 0;
        public static new int Count {   get { return count; }   }
        private int id;
        public int ID { get { return id; } set { id = value; } }
        public Student() { id = 5208; }
        public Student(string name, int age, int id) : base(name, age) { this.id = id; count++; }
        public void Set(string name, int age, int id) { base.Set(name, age); this.id = id; }
        public void Set(Student other) { this.Set(other.Name, other.Age, other.id); }
        public override void Set(string name, int age) { base.Set(name, age); }
        public override void Set(Person other) {
                if (other is Person) base.Set(other);
                else this.Set((Student)other);
        }
        public new void Print() { Console.WriteLine("Student Name: {0} Age: {1} ID: {2})", Name, Age, id);   }
        public override string ToString() { return string.Format("Student Name: {0} Age: {1} ID: {2})", Name, Age,
id);   }
}
class Program {
```

학과 _____ 학번 _____ 이름 _____

```
        static void Print(object o) { Console.WriteLine(o); }
        static void Print(Person[] arr) {   foreach(var e in arr) Console.WriteLine(e); }
        static void Main(string[ ] args) {
            Person h1 = new Person("Park", 20);
            Print(h1);                  // (1)
            Person h2 = new Student();
            object o = h2;
            ((Person)o).Print();        // (2)
            ((Student)o).Print();       // (3)
            Console.WriteLine(o);  // (4)
            Console.WriteLine("Person Count: " + Person.Count); // (5)
            Console.WriteLine("Student Count: " + Student.Count); // (6)
            Person[] pList = new Person[5];
            pList[0] = new Person("HCI1", 1);
            pList[0].Print();   // (7)
            pList[1] = pList[0];
            pList[1].Print();   // (8)
            pList[2] = new Student();
            pList[2].Age = 10;
            pList[2].Print();   // (9)
            pList[3] = new Student("HCI2", 2, 111);
            pList[3].Print();   // (10)
            pList[4] = pList[3];
            pList[4].Print();   // (11)
            Console.WriteLine("Person Count: " + Person.Count); // (12)
            Console.WriteLine("Student Count: " + Student.Count); // (13)
            pList[0].Set("HCI3", 3);
            pList[4].Set("HCI4", 4);
            Print(pList);   // (14-18)
            pList[0].Set(pList[4]);
            Print(pList);   // (19-23)
            ((Student)pList[2]).Set(new Student("HCI5", 5, 222));
            ((Student)pList[3]).Set(new Person("HCI6", 6));
            Print(pList);   // (24-28)
            ((Student)pList[2]).Set("HCI7", 7, 333);
            Print(pList);   // (29-33)
            pList[0].Set(ref pList[1]);
            Print(pList);   // (34-38)
            pList[0].Set("HCI8", 8, out pList[1]);
            Print(pList);   // (39-43)
            Console.WriteLine("Person Count: " + Person.Count); // (44)
            Console.WriteLine("Student Count: " + Student.Count); // (45)
        }
}
```

2.1 Upcasting과 downcasting은 무엇인가? Main 메소드 안에서 각각 그 예를 적어도 3개이상 찾아서 설명하라. (10점)

upcasting은 하위 클래스 객체가 상위 클래스로 자동 형변환되는 것으로,
Print(h1)에서 h1이 Person->object, Person h2 = new Student()에서 h2가 Student->Person,
object o = h2에서 h2가 Person->object, pList[2] = new Student()에서 pList[2]가 Student->Person
downcasting은 상위 클래스 객체가 하위 클래스 객체로 명시적으로 형변환되는 것으로,
((Person)o).Print()에서 o가 object->Person,
Print((Student)o)에서 먼저 o가 object->Student, 그후 다시 o가 Student->object로 업캐스트
((Student)pList[2]).Set(new Student("HCI5", 5, 222))에서 pList[2]가 Person->Student
((Student)pList[3]).Set(new Person("HCI6", 6))에서 pList[3]가 Person->Student

학과 _____ 학번 _____ 이름 _____

2.2 Program의 실행결과를 자세히 나타내라. 실행결과에 호출 번호도 표시할 것. Set 메소드 호출 시 parameter passing에 유의할 것. (45점)

```
Person Name: Park Age: 20          // (1)
Person Name: Age: 0                // (2)
Student Name: Age: 0 Id: 5208      // (3)
Student Name: Age: 0 Id: 5208      // (4)
Person Count=2                     // (5)
Student Count=0                    // (6)
Person Name: HCI1 Age: 1           // (7)
Person Name: HCI1 Age: 1           // (8)
Person Name:   Age: 10             // (9)
Person Name: HCI2 Age: 2           // (10)
Person Name: HCI2 Age: 2           // (11)
Person Count=5                     // (12)
Student Count=1                    // (13)
Person Name: HCI3 Age: 3           // (14-18)
Person Name: HCI3 Age: 3
Student Name: Age: 10 Id: 5208
Student Name: HCI4 Age: 4 Id: 111
Student Name: HCI4 Age: 4 Id: 111
Person Name: HCI4 Age: 4           // (19-23)
Person Name: HCI4 Age: 4
Student Name: Age: 10 Id: 5208
Student Name: HCI4 Age: 4 Id: 111
Student Name: HCI4 Age: 4 Id: 111
Person Name: HCI4 Age: 4           // (24-28)
Person Name: HCI4 Age: 4
Student Name: HCI5 Age: 5 Id: 222
Student Name: HCI6 Age: 6 Id: 111
Student Name: HCI6 Age: 6 Id: 111
Person Name: HCI4 Age: 4           // (29-33)
Person Name: HCI4 Age: 4
Student Name: HCI7 Age: 7 Id: 333
Student Name: HCI6 Age: 6 Id: 111
Student Name: HCI6 Age: 6 Id: 111
Person Name: HCI Age: 10           // (34-38)
Person Name: HCI Age: 10
Student Name: HCI7 Age: 7 Id: 333
Student Name: HCI6 Age: 6 Id: 111
Student Name: HCI6 Age: 6 Id: 111
Person Name: HCI Age: 10           // (39-43)
Person Name: HCI8 Age: 5
Student Name: HCI7 Age: 7 Id: 333
Student Name: HCI6 Age: 6 Id: 111
Student Name: HCI6 Age: 6 Id: 111
Person Count=8                     // (44)
Student Count=2                    // (45)
```

2.3 Method overloading은 무엇인가? 위의 예제에서 그 예를 찾아서 설명하라. (5점)

Method overloading은 동일한 함수명에 다양한 파라메터와 리턴을 정의하는 것으로
Person클래스의 void Set(string, int), void Set(Person), void Set(ref Person), void Set(string, int, out Person)와
Student 클래스의 void Set(string, int, int), void Set(Student)가 있다.

학과 _____ 학번 _____ 이름 _____

2.4 위의 예제에서 Method overriding 예를 찾아서 설명하라. (5점)

> Method overriding은 상속받은 파생클래스에서 동일한 함수명에 동일한 매개변수 정의하여 함수를 재정의하는 것으로 상속되어진 함수의 기능을 변경해서 재사용하고 싶은 때 사용하는 것으로 Person과 Student 클래스에 동일한 함수를 재정의해서 사용한 void Set(string, int), void Set(Person), string ToString()가 있다.

2.5 위의 예제에서 Student 클래스의 메소드에 new 키워드를 사용한 용도는 무엇인가? 이로 인한 장점과 문제점을 설명하라. (10점)

> 상속관계에서 부모클래스와 자식클래스에서 동일한 이름을 사용하는 경우,
> new 키워드를 사용하여 명시적으로 명확하게 구분하기 위하여 메소드를 새정의(new) 한다.
> Person과 Student의 Count경우 이름은 같아도 각각 자기 자신의 count 값을 전달해야 하기 때문에 new가 필요함.
>
> 그런데 Person과 Student의 Print()의 경우는 new를 사용하여 새정의했기 때문에,
> 객체의 타입에 따라 메소드가 호출되므로
> Main안에서 pList[2].Print() // (9)와 pList[3].Print() // (10)와 pList[4].Print() // (11)는
> 객체의 타입인 Person의 Print가 호출되는 문제가 발생한다.
> 따라서 void Print()도 virtual/override를 사용하여 late-binding이 가능하도록 해야 한다.

2. Point와 Bound 클래스 예제 프로그램이다. 다음 질문에 답하시오. (25점)

```
public class Point {
    public double X { get; set; }
    public double Y { get; set; }
    public static readonly Point Zero; // (1) (0, 0)인 상수
    public Point() : this(0.0, 0.0) { }
    public Point(double x, double y) { Set(x, y); }
    public Point(Point other) { Set(other); }
    public Point(double[] xy) { Set(xy); }
    public void Print() { Console.WriteLine("(" + X + "," + Y + ")"); }
    public String ToString() { return "(" + X + "," + Y + ")"; }
    public Point Clone() { /* 내부구현 필요 */ }  // (2) 자기 자신을 복제하는 함수
    public void Set(Point other) { /* 내부구현 필요 */ } // (3) 다른 점으로 Set
    public void Set(double x, double y) { /* 내부구현 필요 */ } // (4) x, y 값으로 Set
    public void Set(double[ ] xy) { /* 내부구현 필요 */ } // (5) double[ ]로 Set
}
public class Bound {
    private Point bottomLeft;
    private Point bottomRight;
    private Point topLeft;
    private Point topRight;
    private double width;
    private double height;
    public Bound() { /* 내부구현 필요 */ } // (1) 기본 생성자 (2점)
    public Bound(Point origin, double width, double height) {
        this.bottomLeft = origin;
        this.bottomRight = new Point(origin.X + width, origin.Y);
        this.topLeft = new Point(origin.X, origin.Y + height);
        this.topRight = new Point(origin.X + width, origin.Y + height);
        this.width = width;
        this.height = height;
    }
```

학과 _____ 학번 _____ 이름 _____

```
    public Bound(int x, int y, double width, double height) { /* 내부구현 필요 */ } // (2) (4점)
    public Bound(Point bottomLeft, Point topRight) { /* 내부구현 필요 */ } (3) (4점)
    public string ToString() {
        return "Bound [" + bottomLeft + ","+ bottomRight + "," + topRight + "," + topLeft + "]" + width + "x" +
height;
    }
}
```

2.1 Point 클래스의 밑줄 친 메소드를 구현하라. (각각 3점씩 전체 15점)

```
public static readonly Point Zero = new Point(0, 0); // (1) (0, 0)인 상수
public Point Clone() { // (2) 자기 자신을 복제하는 함수
    Point p = new Point(this.X, this.Y); return p;
}
public void Set(Point other) { // (3) 다른 점으로 Set
    this.X = other.X; this.Y = other.Y;
}
public void Set(double x, double y) { // (4) x, y 값으로 Set
    this.X = x; this.Y = y;
}
public void Set(double[] xy) { // (5) double[]로 Set
    if (xy.Length == 2) {
        this.X = xy[0]; this.Y = xy[1];
    }
}
```

2.2 Bound 클래스의 생성자를 구현하라. (10점)

```
public Bound() : this(new Point(0.0, 0.0), 1.0, 1.0) { // (1)
}

public Bound(int x, int y, double width, double height) { // (2) (4점)
    this.bottomLeft = new Point(x, y);
    this.bottomRight = new Point(x + width, y);
    this.topLeft = new Point(x, y + height);
    this.topRight = new Point(x + width, y + height);
    this.width = width;
    this.height = height;
}

public Bound(Point bottomLeft, Point topRight) { // (3) (4점)
    this.bottomLeft = bottomLeft;
    this.bottomRight = new Point(topRight.X, bottomLeft.Y);
    this.topLeft = new Point(bottomLeft.X, topRight.Y);
    this.topRight = topRight;
    this.width = topRight.X – bottomLeft.X;
    this.height = topRight.Y – bottomLeft.Y;
}
```

-끝-