

## 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. IEnumerable 인터페이스는 foreach 구문을 사용하여 컬렉션을 단순하게 반복할 수 있도록 지원한다. 다음 물음에 답하라. (20점)

class Tokens : IEnumerable

```
{
    private string[] elements;

    public Tokens (string source, char[] delimiters) {
        elements = source.Split(delimiters);
    }

    public IEnumerator GetEnumerator() {
        return new TokenEnumerator(this);
    }

    private class TokenEnumerator : IEnumerator {
        private int position = -1;
        private Tokens t;
        public TokenEnumerator (Tokens t) {
            this.t = t;
        }
        public object Current {
            get { return t.elements[position]; }
        }
        public void Reset() {
            position = -1;
        }
        public bool MoveNext() {
            if (position < t.elements.Length - 1) {
                position++;
                return true;
            }
            else {
                return false;
            }
        }
    }
}
```

학과 \_\_\_\_\_

학번 \_\_\_\_\_

이름 \_\_\_\_\_

class Program

```
{
    static void PrintArray (string[] arr) {
        // 내부 구현 필요...
    }

    static void Main(string[] args) {
        Tokens f = new Tokens ("Welcome to HCI2.", new char[] { ' ', ',' });

        foreach (string item in f)
            Console.WriteLine(item);

        PrintArray(f.Elements);
    }
}
```

1.1. Main 메소드 안의 밑줄 친 **foreach** 구문의 출력 결과를 적어라. (5점)

```
Welcome
to
HCI2
```

1.2. **for**와 **Console.WriteLine**를 사용하여 string 배열의 원소를 한 줄 씩 출력하는 **PrintArray(string[] arr)**를 구현하라. (5점)

```
static void PrintArray (string[] arr) {
    for (int i = 0; i < arr.Length; i++)
        Console.WriteLine(item);
}
```

1.3. Main 메소드 안의 **PrintArray(f.Elements)** 구문의 출력 결과가 1.1. 결과와 같도록 Tokens 클래스의 Elements라는 property를 구현하라. (5점)

```
public string[] Elements
{
    get { return elements; }
}
```

1.4. Main 메소드 안에 **IEnumerator**를 사용하여 출력 결과가 1.1 결과와 같도록 **while** 구문을 완성하라. (5점)

```
IEnumerator ie = f.GetEnumerator();
while ( ie.MoveNext() )
{
    Console.WriteLine(ie.Current);
}
```

2. 다음 프로그램의 실행 결과를 써라. Array (참조형식)의 메소드 매개 변수 전달 방식에 주의할 것. (20점)

```
class ArrayParameterPassingTest
{
    static void PrintArray(string[] arr) {
        foreach (var e in arr)
            Console.WriteLine(e);
    }
    static void FillArray1 (string[] arr) {
        for (int i = 0; i < arr.Length; i++)
            arr[i] = i.ToString();
    }
    static void FillArray2 (out string[] arr) {
        arr = new string[3] { "a", "b", "c" };
    }
    static void ChangeArray1(ref string[] arr) {
        arr[0] = "e";
        Console.WriteLine("ChangeArray1 inside1 :");
        PrintArray(arr);
        arr = new string[3] { "a", "b", "c" };
        Console.WriteLine("ChangeArray1 inside2 :");
        PrintArray(arr);
    }
    static void ChangeArray2(int[] arr) {
        arr[0] = "e";
        Console.WriteLine("ChangeArray2 inside1 :");
        PrintArray(arr);
        arr = new string[3] { "a", "b", "c" };
        Console.WriteLine("ChangeArray2 inside2 :");
        PrintArray(arr);
    }
    static void Main(string[] args) {
        string[] myArray1 = new string[3] { "a", "b", "c" };
        Console.WriteLine("myArray1: ");
        PrintArray(myArray1);

        FillArray1(myArray1);
        Console.WriteLine("FillArray1: ");
        PrintArray(myArray1);

        myArray1 = new string[5] { "1", "2", "3", "4", "5" };
        Console.WriteLine("myArray1: ");
        PrintArray(myArray1);

        FillArray2(out myArray1);
        Console.WriteLine("FillArray2: ");
        PrintArray(myArray1);

        string[] myArray2 = new string[3] { "a", "b", "c" };
        ChangeArray1(ref myArray2);
        Console.WriteLine("ChangeArray1: ");
        PrintArray(myArray2);

        myArray2 = new string[5] { "1", "2", "3", "4", "5" };
        ChangeArray2(myArray2);
    }
}
```

학과 \_\_\_\_\_

학번 \_\_\_\_\_

이름 \_\_\_\_\_

```

        Console.WriteLine("ChangeArray2: ");
        PrintArray(myArray2);
    }
}

```

**출력결과:**

```

myArray1 : a b c
FillArray1 : 0 1 2
myArray1 : 1 2 3 4 5
FillArray2 : a b c
ChangeArray1 inside1 : e b c
ChangeArray1 inside2 : a b c
ChangeArray1 : a b c
ChangeArray2 inside1 : e 2 3 4 5
ChangeArray2 inside2 : a b c
ChangeArray2 : e 2 3 4 5

```

2. 다음은 struct, class, inheritance 예제 프로그램이다. 다음 질문에 답하시오. (45점)

```

struct Person2 {
    protected string name;
    protected int age;
    public Person2(): this("HCI", 20) {}
    public Person2(string name, int age) { this.name = name; this.age = age; }
    public virtual void Print() { Console.WriteLine("Person2 Name={0} Age={1}", name, age); }
    public override string ToString() { return string.Format("Person2 Name={0} Age={1}", name, age); }
}

class Person {
    private static int count;
    public static int Count { get { return count; } }
    protected string name;
    protected int age;
    protected Person(): this("HCI", 20) { count++; }
    public Person(string name, int age) { this.name = name; this.age = age; }
    public virtual 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 int id;
    public int Age { get { return age; } set { age = value; } }
    public Student() { id = 2013; }
    public Student(string name, int age, int id) : base(name, age) { this.id = id; }
    public override 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 PersonTest {
    static void Print(object o) {
        Console.WriteLine(o);
    }

    static void Main(string[] args) {
        Person2 h1 = new Person2();
        Print(h1);
        Person2 h2 = new Person2("Park", 20);
        object o2 = h2;
        ((Person2)o2).Print();

        Person p = new Person();

        Person p1 = new Person("HCI", 20);
        Print(p1);
        Student s1 = new Student();
        s1.Age = 10;
        Print(s1);

        Person s2 = new Student("HCI2", 25, 111);
        s2.Print();
        Person s3 = new Student();
        Console.WriteLine(s3);

        Console.WriteLine("Person Count=" + Person.Count);
    }
}
```

3.1 Person2 구조체의 밑줄 친 4부분은 에러가 발생한다. 각각의 이유를 간단히 설명하고 에러가 발생하지 않게 코드를 수정하라. (10점)

Struct는 상속에 사용할 수 없으므로 protected 키워드를 사용할 수 없다.  
 public string name;  
 public int age; 로 수정  
 Struct는 기본 생성자를 쓸 수 없다. Person2() 는 전체를 제거  
 Struct는 상속에 사용할 수 없으므로 virtual 키워드를 사용할 수 없다. public void Print()로 수정

3.2 Main 메소드 안의 밑줄 친 부분 Person p = new Person()은 에러가 발생한다. 그 이유를 설명하라. (5점)

Person 클래스의 기본생성자 Person()은 protected를 사용하여 외부의 접근을 방지하였고  
 상속관계인 Student 클래스에서만 호출 가능하도록 함

3.3 다음 프로그램의 출력 결과를 적어라. (10점)

Person2 Name= Age=0  
 Person2 Name=Park Age=20  
 Person Name=HCI Age=20  
 Student Name=HCI Age=10 ID=2013  
 Student Name=HCI2 Age=25 ID=111  
 Student Name=HCI Age=20 ID=2013  
 Person Count=2

3.4 Polymorphism(다형성)이 무엇인지 설명하라. 위의 코드에서 그 예를 찾아 적어라. (5점)

polymorphism이란 상속을 통해 클래스를 한 개 이상의 형식으로 사용할 수 있는 것을 다형성이라고 한다. 상위클래스에서 선언된 메소드를 하위클래스에서 재정의 (override)하고 실행시간에 새로 정의된 override된 멤버의 내용으로 처리(late binding)한다. 위 예제에서 s2.Print();는 Student의 Print()가 호출되며; 또한 Console.WriteLine(s3);은 Student의 ToString()이 호출; Print(p1);은 Person의 ToString()이 호출; Print(s1);은 Student의 ToString()이 호출;

3.5 만약 위의 코드에서 Person과 Student 클래스에서 Print() 메소드 오버라이딩을 하지 않았다면 출력결과가 어떻게 달라지는지 설명하라. (5점)

위의 예제에서 s2.Print(); 부분에서, s2 객체가 실제 Student 형인데도 불구하고 Person 데이터형으로 인지하고 Person의 Print()가 호출됨

3.6 Boxing과 unboxing이 무엇인지 설명하라. 위의 코드에서 그 예를 찾아 적어라. (5점)

Boxing은 value type을 reference type으로 바꾸는 것  
Unboxing은 reference type을 value type으로 바꾸는 것  
위의 예제에서 Print(h1)은 struct => object 로 boxing이 발생  
마찬가지로 object o2 = h2;도 boxing이 발생  
반면 ((Person2)o2).Print();는 object 형인 o2를 Person2 struct 형으로 unboxing한 후 Print() 호출

3.7 Static 멤버 필드와 메소드가 무엇인지 설명하라. 위의 코드에서 그 예를 찾아 적어라. (5점)

Static 으로 지정된 멤버 필드와 메소드는 클래스의 전역데이터로 클래스 당 하나만 할당됨  
클래스 외부에서는 클래스명.정적필드명 또는 클래스명.정적메소드명 형태로 사용  
위의 예제에서 Person 클래스의 private static int count 와 public static int Count

4. 다음은 추상클래스에 관한 프로그램이다. 다음 질문에 답하시오. (15점)

```
class Point {
    public int X { get; set; }
    public int Y { get; set; }
    public Point() : this(0, 0) {}
    public Point(int x, int y) { this.X = x; this.Y = y; }
    public virtual void Print() { Console.WriteLine("Point ({0},{1})", X, Y); }
    public override string ToString() { return (String.Format("{0},{1})", X, Y)); }
}

class Bounds {
    public int X { get; set; }
    public int Y { get; set; }
    public int Width { get; set; }
    public int Height { get; set; }
    public Bounds() : this(0, 0, 1, 1) {}
    public Bounds(int x, int y, int width, int height)
    {
        this.X = x; this.Y = y; this.Width = width; this.Height = height;
    }
}
```

```

    public Bounds (Point bottomLeft, Point topRight)
    {
        this.X = bottomLeft.X; this.Y = bottomLeft.Y;
        this.Width = topRight.X - bottomLeft.X; this.Height = topRight.Y - bottomLeft.Y;
    }
}

enum Color { Red, Green, Blue, Yellow }
abstract class A {
    protected Color fillColor;
    protected Bounds area;
    protected Point[] vertices;
    protected A(Color fillColor) {
        this.fillColor = fillColor;
        this.area = new Bounds(30, 40, 50, 60);
    }
    public abstract void Draw();
}

class B : A {
    public B(Color fillColor) : base(fillColor) {
        vertices = new Point[2];
        vertices[0] = new Point(area.X, area.Y);
        vertices[1] = new Point(area.X + area.Width, area.Y + area.Height);
    }
    public override void Draw() {
        Console.WriteLine("{0}, {1}, {2}", fillColor, vertices[0], vertices[1]);
    }
}

class C : A {
    public C(Color fillColor) : base(fillColor) {
        vertices = new Point[3];
        vertices[0] = new Point(area.X, area.Y);
        vertices[1] = new Point(area.X + area.Width, area.Y);
        vertices[2] = new Point(area.X + area.Width/2, area.Y + area.Height);
    }
    public override void Draw() {
        Console.WriteLine("{0}, {1}, {2}, {3}", fillColor, vertices[0], vertices[1], vertices[2]);
    }
}

class D : A {
    public D(Color fillColor) : base(fillColor) {
        vertices = new Point[4];
        vertices[0] = new Point(area.X, area.Y);
        vertices[1] = new Point(area.X + area.Width, area.Y);
        vertices[2] = new Point(area.X + area.Width, area.Y + area.Height);
        vertices[3] = new Point(area.X, area.Y + area.Height);
    }
    public override void Draw() {
        Console.WriteLine("{0}, {1}, {2}, {3}, {4}",
            fillColor, vertices[0], vertices[1], vertices[2], vertices[3]);
    }
}

class E : D {
    public E(Color fillColor) : base(fillColor) {
        if (area.Width < area.Height) {
            area.Height = area.Width;
            vertices[2].Y = vertices[3].Y = area.Y + area.Height;
        }
    }
}

```

```

    }
    else {
        area.Width = area.Height;
        vertices[1].X = vertices[2].X = area.X + area.Width;
    }
}
public override void Draw() {
    Console.WriteLine("{0}, {1}, {2}, {3}, {4}",
        fillColor, vertices[0], vertices[1], vertices[2], vertices[3]);
}
}
class AFactory {
    static public A GetInstance(int id) {
        switch (id) {
            case 1: return new B (Color.Red);
            case 2: return new C (Color.Green);
            case 3: return new D (Color.Blue);
            case 4: return new E (Color.Yellow);
            default: return null;
        }
    }
}
class Program {
    static void Main(string[] args) {
        A anInstance = AFactory.GetInstance(1);
        anInstance.Draw();
        anInstance = AFactory.GetInstance(3);
        anInstance.Draw();
        anInstance = AFactory.GetInstance(2);
        anInstance.Draw();
        anInstance = AFactory.GetInstance(4);
        anInstance.Draw();
        anInstance = AFactory.GetInstance(3);
        anInstance.Draw();
    }
}

```

4.1 Main 메소드에 실행결과와 같은 결과가 나타나도록, A[] aList 를 정의하고 foreach 구문을 사용하여 출력하는 코드를 작성하라. (5점)

```

A[] aList = new A[5] { new B(Color.Red),
                       new D(Color.Blue),
                       new C(Color.Green),
                       new E(Color.Yellow),
                       new D(Color.Blue) };

foreach (var elem in aList) {
    elem.Draw();
}

```

4.2 만약 B 클래스의 생성자에서 base(fillColor)를 직접 호출하지 않았다면, 즉 public B(Color fillColor) 만 호출한다면 무슨 문제가 발생하는가? (5점)

public B(Color fillColor)는 public B(Color fillColor) : base()를 호출하므로  
A 클래스에 기본생성자가 없으므로, 컴파일 에러 발생



학과 \_\_\_\_\_ 학번 \_\_\_\_\_ 이름 \_\_\_\_\_

4.3 다음 프로그램에서 is-a 관계와 has-a 관계를 찾아서 설명하라. (5점)

A 클래스 has-a Bounds 와 Point 클래스  
B, C, D, E 클래스 is-a A 클래스  
E 클래스 is-a D 클래스

-끝-