단국대학교 멀티미디어공학 HCI 프로그래밍 II 중간고사 (2012년 가을학기) 2012년 10월 16일

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 암호를 기입하면 성적공고시 학번대신 암호를 사용할 것임.

1. 다음 프로그램의 실행 결과를 써라. 전역변수와 지역변수, 그리고 Integer(값형식) 메소드 매개 변수 전달 방식에 주의할 것. (10점)

```
class MethodTest
{
        static int i = 10;

        static void Multiply1(int i) {
            i *= 2;
            Console.WriteLine("Multiply1 : i=" + i.ToString());
        }
        static void Multiply2(ref int i) {
            i *= 2;
            Console.WriteLine("Multiply2 : i=" + i.ToString());
        }
        static int Multiply3(int a, int b) {
            int i = a * b;
            Console.WriteLine("Multiply3 : i=" + i.ToString());
            return i;
        }
        static void Multiply4(int a, int b, ref int c) {
            i *= 2;
            Console.WriteLine("Multiply4 : i=" + i.ToString());
            Multiply1(i);
            Console.WriteLine("Multiply4 : i=" + i.ToString());
            c = Multiply3(a, b);
            Console.WriteLine("Multiply4 : a={0} b={1} c={2}", a, b, c);
        }
         static void Main(string[ ] args) {
            Console.WriteLine("Main1: i=" + i.ToString());
            i *= 2;
            Console.WriteLine("Main2: i=" + i.ToString());
            Multiply2(ref i);
            Console.WriteLine("Main3: i=" + i.ToString());
            int a = 4, b = 5, c = 6;
            Multiply4(a, b, ref c);
            Console.WriteLine("Main4: i=" + i.ToString());
        }
}
```

학과 _____ 학번 _____ 이름 _____

출력결과:

```
Main1 : i=10
Main2 : i=20
Multiply2 : i=40
Main3 : i=40
Multiply4 : i=80
Multiply1 : i=160
Multiply4 : i=80
Multiply3 : i=20
Multiply4 : a=4 b=5 c=20
Main4 : i=80
```

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

```
class ArrayParameterPassingTest
{
        static void PrintArray(int[] arr) { // 원본 배열의 모든 원소를 한 줄에 출력
            for (int i = 0; i < arr.Length; i++)
                Console.Write("{0} ", arr[i]);
            Console.WriteLine();
        }
        static void FillArray1 (int[] arr) {
            for (int i = 0; i < arr.Length; i++)
                arr[i] = i + 1;
        }
        static void FillArray2 (out int[] arr) {
            arr = new int[5] { 1, 2, 3, 4, 5 };
        }
        static void ChangeArray1(ref int[] arr) {
            arr[0] = 10; // 원본 배열의 첫번째 값은 10로 변경
            Console.Write("ChangeArray1 inside1 :");
            PrintArray(arr);
            arr = new int[5] { -1, -2, -3, -4, -5 };
            Console.Write("ChangeArray1 inside2 :");
            PrintArray(arr);
        }
        static void ChangeArray2(int[] arr) {
            arr[0] = 10; // 원본 배열의 첫번째 값은 10로 변경
            Console.Write("ChangeArray2 inside1 :");
            PrintArray(arr);
            arr = new int[5] { -1, -2, -3, -4, -5 };
            Console.Write("ChangeArray2 inside2 :");
            PrintArray(arr);
        }
```

학과 _____ 학번 _____ 이름 _____

```
        static void Main(string[ ] args) {
            int[] myArray1 = new int[3];
            Console.Write("myArray1: ");
            PrintArray(myArray1);

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

            myArray1 = new int[3] { 1, 2, 3};
            Console.Write("myArray1: ");
            PrintArray(myArray1);

            FillArray2(myArray1);
            Console.Write("FillArray2: ");
            PrintArray(myArray1);

            int[] myArray2 = new int[3] { 1, 2, 3};
            ChangeArray1(ref myArray2);
            Console.Write("ChangeArray1: ");
            PrintArray(myArray2);

            myArray2 = new int[3] { 1, 2, 3};
            ChangeArray2(myArray2);
            Console.Write("ChangeArray2: ");
            PrintArray(myArray2);
        }
}
```

출력결과:

```
myArray1 : 0 0 0
FillArray1 : 1 2 3
myArray1 : 1 2 3
FillArray2 : 1 2 3 4 5
ChangeArray1 inside1 : 10 2 3
ChangeArray1 inside2 : -1 -2 -3 -4 -5
ChangeArray1 : -1 -2 -3 -4 -5
ChangeArray2 inside1 : 10 2 3
ChangeArray2 inside2 : -1 -2 -3 -4 -5
ChangeArray2 : 10 2 3
```

학과 _____ 학번 _____ 이름 _____

3. 다음 대리자(Delegate)를 사용한 예제 프로그램의 실행 결과를 설명과 함께 적어라. (10점)

```
public delegate bool NumTest(int value);

class NumValue {
        public int A { get;   set; }
        public static int B { get;    set; }
        public readonly int C=1;
        public NumValue(int a, int b, int c) { A = a; B = b; C = c; }
        public bool TestA(int value) { Console.WriteLine("TestA"); return value < A; }
        public static bool TestB(int value) { Console.WriteLine("TestB"); return B == value; }
        public bool TestC(int value) { Console.WriteLine("TestC"); return value > C; }
        public static bool TestD(int value) { Console.WriteLine("TestD"); return value > 5 ? false : true; }
}

class NumValueTest {
        static void Main(string[ ] args) {
            NumValue n = new NumValue(5, 4, 3);
            NumTest Test = new NumTest(n.TestA);
            if (Test(3))
                Console.WriteLine("Test1 true");
            else
                Console.WriteLine("Test1 false");
            Test += new NumTest(NumValue.TestB);
            if (Test(4))
                Console.WriteLine("Test2 true");
            else
                Console.WriteLine("Test2 false");
            Test = Test - NumTest(NumValue.TestB) + new NumTest(NumValue.TestD);
            if (Test(1))
                Console.WriteLine("Test3 true");
            else
                Console.WriteLine("Test3 false");
            Test = new NumTest(n.TestC);
            if (Test(2))
                Console.WriteLine("Test4 true");
            else
                Console.WriteLine("Test4 false");
        }
}
```

출력결과:

```
TestA
Test1 true // TestA 에서 3 < 5 true 이므로
TestA
TestB
Test2 true // TestA 에 4 < 5 true 이고 TestB 에 4==4 true 이므로
TestA
TestD
Test3 true // TestA 에 1 < 5 true 이고 TestD 에 1>5 아니라서 true 이므로
TestC
Test4 false // TestC 에 2 > 3 아니라서 false
```

학과 _____ 학번 _____ 이름 _____

4. 다음은 instance와 static과 readonly에 대한 예제를 보여준다. 먼저 n.Print()와 같은 출력 결과가 나오도록 WriteLine 메소드 안에 밑줄 친 빈칸을 채워라. 그리고 아래 밑줄 친 부분에서 compile error가 발생하는 부분이 어디인지 지적하고 올바르게 수정하라. (10점)

```
class NumValue {
        // 중간 생략....
        public void Print() { // Print 메소드 구현
            Console.WriteLine("A={0} B={1} C={2}", A, B, C);
        }
}
class NumValueTest2 {
        static void Main(string[ ] args) {
            NumValue n = new NumValue(5, 4, 3);
            n.Print();      // 출력결과는 A=5 B=4 C=3
            Console.WriteLine("A={0} B={1} C={2}", n.A, NumValue.B, n.C); // n.Print와 같도록

            if (n.TestA(6) && n.TestC(6))
                Console.WriteLine("TestA(6) & TestC(6) true");
            if (n.TestB(7) && n.TestD(7))
                Console.WriteLine("TestB(7) & TestD(7) true");
            NumValue.A = 3;
            n.B = 3;
            n.C = 3;
        }
}
```

오류:

if (n.TestB(7) && n.TestD(7)) => if (NumValue.TestB(7) && NumValue.TestD(7))

NumValue.A = 3; => n.A = 3; // A는 instance property이기 때문에

n.B = 3; => NumValue.B = 3; // B는 static property이기 때문에

n.C = 3; => readonly field는 생성자에서 지정한 숫자에서 바꿀 수가 없다.

5. Point 클래스를 사용한 값형식과 참조형식을 비교(==연산자)한다. 이 프로그램이 실행되는 출력결과를 써라. (10점)

```
class Point : IEquatable<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 void SetPosition(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)); }
        public static bool operator==(Point p1, Point p2) { return p1.Equals(p2); }
        public static bool operator!=(Point p1, Point p2) { return !p1.Equals(p2); }
        public override int GetHashCode() { return X ^ Y; }
        public bool Equals(Point other) {
            if (this.X == other.X && this.Y == other.Y)
                return true;
            return false;
        }
}
class EqualTest {
```

학과 _____ 학번 _____ 이름 _____

```
static void Main(string[] args) {
    int val1 = 5;
    int val2 = 10;
    if (val1 == val2)
        Console.WriteLine("val1 == val2");
    else
        Console.WriteLine("val1 != val2");
    object val3 = val1;
    object val4 = val2;
    if (val3 == val4)
        Console.WriteLine("val3 == val4");
    else
        Console.WriteLine("val3 != val4");
    int val5 = (int)val3;
    int val6 = (int)val3;
    if (val5 == val6)
        Console.WriteLine("val5 == val6");
    else
        Console.WriteLine("val5 != val6");
    string val7 = val5.ToString();
    string val8 = val6.ToString();
    if (val7 == val8)
        Console.WriteLine("val7 == val8");
    else
        Console.WriteLine("val7 != val8");
    Point pos1 = new Point(val1, val2);
    Point pos2 = new Point(val1, val2);
    if (pos1 == pos2)
        Console.WriteLine("pos1 == pos2");
    else
        Console.WriteLine("pos1 != pos2");
    if (pos1.X == pos2.X)
        Console.WriteLine("pos1.X == pos2.X");
    else
        Console.WriteLine("pos1.X != pos2.X");
    if (pos1.Y == pos2.Y)
        Console.WriteLine("pos1.Y == pos2.Y");
    else
        Console.WriteLine("pos1.Y != pos2.Y");
    object pos3 = pos1;
    object pos4 = pos2;
    if (pos3 == pos4)
        Console.WriteLine("pos3 == pos4");
    else
        Console.WriteLine("pos3 != pos4");
    Point pos5 = (Point) pos3;
    Point pos6 = (Point) pos4;
    if (pos5 == pos6)
        Console.WriteLine("pos5 == pos6");
    else
        Console.WriteLine("pos5 != pos6");
    string pos7 = pos5.ToString();
    string pos8 = pos6.ToString();
    if (pos7 == pos8)
        Console.WriteLine("pos7 == pos8");
    else
        Console.WriteLine("pos7 != pos8");
```

학과 _____ 학번 _____ 이름 _____

```
    }
}
```

출력결과:

```
val1 != val2
val3 != val4
val5 == val6
val7 == val8
pos1 == pos2
pos1.X == pos2.X
pos1.Y == pos2.Y
pos3 != pos4
pos5 == pos6
pos7 == pos8
```

6. 다음 프로그램에서 Bounds 클래스의 생성자 3개를 정의하라. Bounds 클래스는 X와 Y 시작 위치와 Width와 Height 너비와 높이 속성을 갖는다. (10점)

```
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;
        }
```

```
}
```

학과 _____ 학번 _____ 이름 _____

7. 다음은 추가로 A, B, C, D, C 클래스를 구현하였다. 다음 프로그램의 출력 결과를 적고 다형성 (Polymorphism)이 무엇인지 예를 들어 자세히 설명하라. (10점)

```
enum Color { Red, Green, Blue }
abstract class A {
        protected Color fillColor;
        protected Bounds area;
        protected Point[ ] vertices;
        public A(Color fillColor, Bounds area) {
            this.fillColor = fillColor;
            this.area = area;
        }
        public abstract void Draw();
}
class B : A {
        public B(Color fillColor, Bounds area) : base(fillColor, area) {
            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}색과 {1} {2}점으로 선을 그린다.",
                fillColor, vertices[0].ToString(), vertices[1].ToString());
        }
}
class C : A {
        public C(Color fillColor, Bounds area) : base(fillColor, area) {
            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}색과 {1} {2} {3}점으로 삼각형을 그린다.",
                fillColor, vertices[0],vertices[1], vertices[2]);
        }
}
class D : A {
        public D(Color fillColor, Bounds area) : base(fillColor, area) {
            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/2, area.Y + area.Height);
            vertices[3] = new Point(area.X + area.Width/2, area.Y + area.Height);
        }
        public override void Draw() {
            Console.WriteLine("{0}색과 {1} {2} {3} {4}점으로 사각형을 그린다.",
                fillColor, vertices[0],vertices[1], vertices[2], vertices[3]);
        }
}
class E : D {
        public E(Color fillColor, Bounds area) : base(fillColor, area) {
            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 sealed override void Draw() {
            Console.WriteLine("{0}색과 {1} {2} {3} {4}점으로 정사각형을 그린다.",
                fillColor, vertices[0],vertices[1], vertices[2], vertices[3]);
        }
}

class Program {
        static void Main(string[] args) {
            Point p1 = new Point3D(30, 40);
            Point p2 = new Point3D(80, 100);
            A anInstance = new C(Color.Red, new Bounds(30, 40, 50, 60));
            anInstance.Draw();
            anInstance = new D(Color.Green, new Bounds(30, 40, 50, 60));
            anInstance.Draw();
            anInstance = new E(Color.Blue, new Bounds(p1, p2));
            anInstance.Draw();
            anInstance = new B(Color.Blue, new Bounds(p1, p2));
            anInstance.Draw();

            D s1 = new E(Color.Red, new Bounds(new Point(10, 20), new Point(50, 60)));
            s1.Draw();
            E s2 = (E)s1;
            s2.Draw();
            B s3 = (B)anInstance;
            s3.Draw();
        }
}
```

- **다형성** – 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 Draw() 메소드는 실행시간에 정의된 객체의 타입에 맞는 Draw()로 호출되어 진다.
- **Upcasting의 예** – 생성된 모든 anInstance는 각각 C, D, E, B 클래스 객체로써 upcasting이다. 또한 s1도 E 클래스 객체를 갖는 upcasting이다.

- **Downcasting의 예** – s2는 s1이 실제 E 클래스의 객체이므로 downcasting된다. 또한 s3도 마지막 anInstance가 B 클래스의 객체이므로 B 타입으로 downcasting된다.

- **실행결과**

  Red색과 (30, 40) (80, 40) (55, 100) 점으로 삼각형을 그린다.
  Green색과 (30, 40) (80, 40) (80, 100) (30, 100) 점으로 사각형을 그린다.
  Blue색과 (30, 40) (80, 40) (80, 90) (30, 90) 점으로 정사각형을 그린다.
  Blue색과 (30, 40) (80, 100) 점으로 선을 그린다.
  Red색과 (10, 20) (50, 20) (50, 60) (10, 60) 점으로 정사각형을 그린다.
  Red색과 (10, 20) (50, 20) (50, 60) (10, 60) 점으로 정사각형을 그린다.
  Blue색과 (30, 40) (80, 100) 점으로 선을 그린다.

학과 _____ 학번 _____ 이름 _____

8. 다음 TodayMoneyExchangeConverter 클래스의 GetTodayCurrencyExchangeRateBy 메소드 안에 if와 swtich문을 사용하여 나머지 부분을 완성하라. (10점)

```
// 통화명과 환율 기준방법으로 오늘의 환율 시세 얻기
double GetTodayCurrencyExchangeRateBy(string name, ExchangeMethod exchange) {
        WebClient client = new WebClient();
        client.Headers.Add("Referer", "http://blog.naver.com/usback");
        Stream xmlDataStream =
            client.OpenRead("http://www.naver.com/include/timesquare/widget/exchange.xml");
        XmlReader reader = XmlReader.Create(xmlDataStream);
        reader.MoveToContent();
        // 통화이름(name)과 기준방법(exchange)으로
        while (reader.ReadToFollowing("currency")) {
            reader.ReadToFollowing("hname");
            string hname = reader.ReadElementContentAsString("hname", reader.NamespaceURI);
            reader.ReadToFollowing("standard");
            double standard = reader.ReadElementContentAsDouble("standard", reader.NamespaceURI);
            reader.ReadToFollowing("buy");
            double buy = reader.ReadElementContentAsDouble("buy", reader.NamespaceURI);
            reader.ReadToFollowing("sell");
            double sell = reader.ReadElementContentAsDouble("sell", reader.NamespaceURI);
            reader.ReadToFollowing("sign");
            string sign = reader.ReadElementContentAsString("sign", reader.NamespaceURI);
            if (hname == name)
            {
                switch (exchange)
                {
                    case ExchangeMethod.BASIS:
                        return standard;
                    case ExchangeMethod.BUY:
                        return buy;
                    case ExchangeMethod.SELL:
                        return sell;
                }
            }
        }
        return 0.0;
    }
```

학과 _____ 학번 _____ 이름 _____

9. IEquatable, IComparable, IEnumerable과 IEnumerator 인터페이스를 간단히 설명하라. 다음 Point 클래스에 IComparable<T> 인터페이스를 상속받아 구현(implement)하라. (10점)

```
class Point : IEquatable<Point>, IComparable<Point>
{

// 중간 생략 ...

        // Point 클래스의 X가 같으면 Y가 누가 더 큰지를 비교
        public int CompareTo(Point other)
        {
            if (this.X == other.X)
                return this.Y.ToCompare(other.Y);
            return this.X.ToCompare(other.X);
        }
}
```

IEquatable 인터페이스:

IEquatable 인터페이스는 현 객체가 동일한 형태의 다른 객체와 같은지 여부를 확인하는 ***Equals*** 메소드를 제공하는 인터페이스로, 현 객체가 obj 인수의 값과 같으면 true, 다르면 false를 반환하도록 구현한다. ==와 != 연산자 오버로딩이 함께 되어있을 때 두 객체의 내용이 같은지 비교하는데 사용된다.

IComparable 인터페이스:

IComparable 인터페이스는 현 객체를 동일한 형식의 다른 객체와 비교하는 (인터페이스 정렬 순서를 확인하는) ***CompareTo*** 메소드를 제공하는 인터페이스로, 현 객체가 obj 인수보다 작으면 0보다 작은 수를 같으면 0을 그면 0보다 큰 수를 반환하도록 구현한다. 컬렉션을 사용할 시 Sort (정렬)할 때 사용된다.

IEnumerable & IEnumerator 인터페이스:

IEnumerable 인터페이스는 컬렉션을 반복하는 IEnumerator 객체를 반환하는 ***GetEnumerator*** 메소드를 제공하는 인터페이스로, foreach를 사용하여 컬렉션을 반복하는 것을 지원하기 위해 구현하여 사용한다.
IEnumerator 인터페이스는 컬렉션에서 요소들을 참조할 수 있는 3개의 메소드 및 속성을 갖는다.
object Current – 컬렉션의 현재 요소를 가져오는 속성
bool MoveNext() – 컬렉션의 다음 요소로 이동하는 메소드
void Reset() – 컬렉션의 첫 번째 요소 앞의 초기 위치로 설정하는 메소드

-끝-