단국대학교 멀티미디어공학 HCI 프로그래밍 II 중간고사 (2011년 가을학기) 2011년 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 암호를 기입하면 성적공고시 학번대신 암호를 사용할 것임.

## 1. 다음 프로그램의 실행 결과를 써라. 전역변수와 지역변수에 주의할 것. (10점)

```
class MethodTest
{
        static int i = 5;
        static void Multiply1(int i) {
            i *= 2;
            Console.WriteLine("Multiply1 : i=" + i.ToString());
        }
        static int Multiply2(int a, int b) {
            int i = 2;
            i *= 2;
            Console.WriteLine("Multiply2 : i=" + i.ToString());
            Multiply1(i);
            int c = a * b;
            Console.WriteLine("Multiply2 : a={0} b={1} c={2}", a, b, c);
            return c;
        }
        static void Multiply3(int a, int b, ref int c) {
            i *= 2;
            Console.WriteLine("Multiply3 : i=" + i.ToString());
            c = Multiply2(a, b);
            Console.WriteLine("Multiply3 : c=" + c.ToString());
        }
         static void Main()
         {
             Console.WriteLine("Main1: i=" + i.ToString());
             i *= 2;
             Console.WriteLine("Main2: i=" + i.ToString());
             Multiply1(i);
```

```
            Console.WriteLine("Main3: i=" + i.ToString());
            int a = 3, b = 4, c = 6;
            Multiply3(a, b, ref c);
            Console.WriteLine("Main4: i=" + i.ToString());
        }
}
```

```
출력결과:
        Main1 : i=5
        Main2 : i=10
        Multiply1 : i=20
        Main3 : i=10
        Multiply3 : i=20
        Multiply2 : i=4
        Multiply1 : i=8
        Multiply2 : a=3 b=4 c=12
        Multiply3 : c=12
        Main4 : i=20
```

## 2. StandardWeightCalculator 프로그램의 다음 메소드들의 내부 구현을 하라. (10점)

```
enum Gender { Female, Male }
enum BMI { Underweight, Normal, Overweight, Obesity }
enum Activity { Low, Medium, High }

// GetGenderFromString 메소드는 str값을 남녀에 따라 Gender 열거형으로 반환
Gender GetGenderFromString(string str)
{
            if (str == "남" || str == "M")
                return Gender.Male;
            return Gender.Female;
}

// GetNormalWeight 메소드는 남녀별로 신장(cm)을 이용하여 표준체중(kg)를 계산
// 남자표준체중 = 신장(cm) x 신장(cm) x 22 x 0.0001f
// 여자표준체중 = 신장(cm) x 신장(cm) x 21 x 0.0001f
float GetNormalWeight(float height, Gender gender)
{
            if (gender == Gender.Male)
                return height * height * 22;
            return height * height * 21;
}
```

학과 _____ 학번 _____ 이름 _____

```
// GetBMI 메소드는 기준에 근거하여 BMI 열거형으로 반환
// BMI = 체중(kg)/신장^2(m^2)
// BMI계산값이 20미만 저체중, 20~24 정상, 25~29 과체중, 30이상 비만
BMI GetBMI(float height, float weight)
{
        float bmi = weight / (height * 0.01f)/ (height * 0.01f);
        Console.WriteLine("bmi=" + bmi);
        if (bmi < 20.0)
            return BMI.Underweight;
        else if (bmi >= 20.0 && bmi < 24.0)
            return BMI.Normal;
        else if (bmi >= 24.0 && bmi < 30.0)
            return BMI.Overweight;
        return BMI.Obesity;
}

// GetActivityFromString 메소드는 str값이 Low/Medium/High에따라 Activity 열거형으로 반환
Activity GetActivityFromString(string str)
{
        if (str == "Low")
            return Activity.Low;
        else if (str == "Medium")
            return Activity.Medium;
        return Activity.High;
}
```

학과 _____ 학번 _____ 이름 _____

**3. 값 형식과 참조 형식에 대한 Pass-by-value와 Pass-by-reference 메소드 매개 변수 전달 방식을 본인이 직접 예를 들어 간단히 설명하라. (10점)**

- **Pass value type by value (값 형식을 값에 의한 전달 방식)**
  - 값 형식(value type)은 값을 copy하여 매개변수에 전달
  - static void Square1(int x) {

```
        x*= x; Console.WriteLine("The value inside: {0}", x); // 함수내부에선 변경된 값이 출력 }
    static void Main() {
        int i =5; Square1(i);
        Console.WriteLine("i={0}", i); // 그러나, 함수에서 변경된 값이 외부에 반영되지 않음
    }
```

- **Pass reference type by value (참조 형식을 값에 의한 전달 방식)**
  - 참조형식(reference type)은 참조를 copy하여 매개변수에 전달
  - static void ChangeArray1(int[] arr) {

```
        arr[0] = 888; // 메인의 myArray를 같이 참조하고 있으므로 myArray[0]=888로 변경
        arr = new int[5] {-3, -1, -2, -3, -4}; // 함수내부에서 arr는 새로 지정
        Console.WriteLine("The value inside: arr[0]={0}", arr[0]); // 새로 지정된 배열로 출력 }
    static void Main() {
        int[] myArray ={1, 4, 5};    ChangeArray1(myArray);
        Console.WriteLine("myArray[0]={0}", myArray[0]); // myArry[0]=888로 출력
    }
```

- **Pass value type by reference (값 형식을 참조에 의한 전달 방식)**
  - ref 키워드를 사용하여, 값 형식(value type)을 참조에 의한 전달방식을 사용
  - static void Square2(ref int x) {

```
        x*= x; Console.WriteLine("The value inside: {0}", x); // 메인 i를 참조하므로 값이 변경 }
    static void Main() {
        int i =5; Square2(i);
        Console.WriteLine("i={0}", i); // 함수에서 변경된 값이 외부로 반영됨
    }
```

- **Pass reference type by reference (참조 형식을 참조에 의한 전달 방식)**
  - ref 키워드를 사용하여, 참조 형식(reference type)을 참조에 의한 전달방식을 사용
  - static void ChangeArray2(ref int[] arr) {

```
        arr[0] = 888; // 메인의 myArray를 같이 참조하고 있으므로 myArray[0]=888로 변경
        arr = new int[5] {-3, -1, -2, -3, -4}; // 그러나 원본 배열이 다시 변경
        Console.WriteLine("The value inside: arr[0]={0}", arr[0]); // 새로 지정된 배열로 출력 }
    static void Main() {
        int[] myArray ={1, 4, 5};    ChangeArray2(myArray);
        Console.WriteLine("myArray[0]={0}", myArray[0]); // myArry[0]=-3로 출력
    }
```

학과 _____ 학번 _____ 이름 _____

**4. 다음 A클래스에서 포함하는 Size 클래스를 정의하라. Size 클래스는 Width와 Height 속성과 생성자와 Print 그리고 ToString 메소드를 갖는다. (10점)**

```
enum Color { Red, Green, Blue }
abstract class A
{
        public Color FillColor {
            get;
            set;
        }
        public Point Position {
            get;
            set;
        }
        public Size Size {
            get;
            set;
        }
        public A(Color fillColor, Point position, Size size) {
            this.FillColor = fillColor;
            this.Position = position;
            this.Size = size;
        }
        public abstract void Draw();
}
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 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)); }
}
class Size
{
        public int Width {
                get;
                set;
        }
        public int Height {
                get;
                set;
        }
        public Size() : this(0, 0) { }
        public Size(int width, int height) { this.Width = width; this.Height = height; }
        public virtual void Print() { Console.WriteLine("Size ({0}x{1})", Width, Height); }
        public override string ToString() { return (String.Format("({0},{1})", Width, Height)); }

}
```

학과 _____ 학번 _____ 이름 _____

**5. 다음 Point3D 클래스와 Point4D 클래스의 메소드를 모두 구현하라. base와 this 키워드를 사용하여 코드를 간결하게 작성한다. (10점)**

```
class Point3D : Point
{
        public int Z {
                get;
                set;
        }
        public Point3D() : this(0, 0, 0) { }
        public Point3D(int x, int y, int z) : base(x, y) { this.Z = z; }
        public void SetPosition(int x, int y, int z) { base.SetPosition(x, y); this.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)); }
}

class Point4D : Point3D
{
        public int W {
                get;
                set;
        }

        public Point4D() : this(0, 0, 0, 0) { }
        public Point4D(int x, int y, int z, int w) : base(x, y, z) { this.W = w; }
        public void SetPosition(int x, int y, int z, int w) { base.SetPosition(x, y, z); this.W = w; }
        public override void Print() { Console.WriteLine("Point4D ({0},{1},{2},{3})", X, Y, Z, W); }
        public override string ToString() { return (String.Format("Point4D ({0}, {1}, {2},{3})", X, Y, Z, W)); }
}
```

학과 _____ 학번 _____ 이름 _____

**6. 다음은 B, C, D, E 클래스를 추가적으로 보여주고 있다. 다형성 (Polymorphism)이 무엇인지 예를 들어 자세히 설명하라. 그리고 Main 메소드의 출력 결과를 적어라. (10점)**

```
class B : A
{
        public B(Color fillColor, Point position, Size size)
            : base(fillColor, position, size) { }
        public override void Draw() {
            Console.WriteLine("B는 ({0} {1})에 {2}x{3}크기와 {4}색",
                                Position.X, Position.Y, Size.Width, Size.Height, FillColor);
        }
}

class C : A
{
        public C(Color fillColor, Point position, Size size)
            : base(fillColor, position, size) { }
        public override void Draw() {
            Console.WriteLine("C는 ({0} {1})에 {2}x{3}크기와 {4}색",
                                Position.X, Position.Y, Size.Width, Size.Height, FillColor);
        }
}

class D : A
{
        public D(Color fillColor, Point position, Size size)
            : base(fillColor, position, size) { }
        public override void Draw() {
            Console.WriteLine("D는 ({0} {1})에 {2}x{3}크기와 {4}색",
                                Position.X, Position.Y, Size.Width, Size.Height, FillColor);
        }
}

class E : D
{
        public E(Color fillColor, Point position, Size size)
            : base(fillColor, position, size)   {
            if (size.Width < size.Height)
                Size.Height = size.Width;
            else
                Size.Width = size.Height;
        }
        public sealed override void Draw() {
            Console.WriteLine("E는 ({0} {1})에 {2}x{3}크기와 {4}색",
                                Position.X, Position.Y, Size.Width, Size.Height, FillColor);
        }
}

class Program
{
        static void Main(string[] args)
        {
            A[] aList = new A[5] { new C(Color.Red, new Point(0, 0), new Size(10, 30)),
                                   new D(Color.Green, new Point(20, 30), new Size(10, 30)),
                                   new B(Color.Blue, new Point(30, 40), new Size(20, 30)),
                                   new E(Color.Blue, new Point(50, 60), new Size(40, 50)),
                                   new D(Color.Blue, new Point(10, 20), new Size(20, 30))
```

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

        Point p1 = new Point(1, 2);
        Point3D p2 = new Point3D(10, 11, 12);
        Point4D p3 = new Point4D(20, 21, 22, 23);
        Console.WriteLine("p1: {0}", p1);
        Console.WriteLine("p2: {0}", p2);
        Console.WriteLine("p3: {0}", p3);
        p1 = p2;
        p2 = p3;
        p1.Print();
        p2.Print();
        p3.Print();

        p2.SetPosition(4, 5, 6);
        Console.WriteLine("p2: {0}", p2);
        p3 = (Point4D)p2;
        Console.WriteLine("p3: {0}", p3);
        Point4D p4 = (Point4D)p3;
        Console.WriteLine("p4: {0}", p4);
    }
}
```

- **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.
- **예를 들어** - p1 = p2; // upcasting; p1은 Point3D객체 (10,11,12)
  p2 = p3; // upcasting; p2는 Point4D객체 (20,21,22,23)
  p2.SetPosition(4, 5, 6); // p2가 실제 Point4D객체 (20,21,22,23)이므로 (4,5,6,23)로 변경
  p3 = (Point4D)p2; // downcasting; p2가 실제로 Point4D객체이므로 p3도 Point4D 객체
  Point4D p4 = (Point4D)p3; // downcasting; p3가 실제로 Point4D객체이므로 p4도 Point4D 객체
- **실행결과**

```
C는 (0 0)에 10x30크기와 Red색
D는 (20 30)에 10x30크기와 Green색
B는 (30 40)에 20x30크기와 Blue색
E는 (50 60)에 40x40크기와 Blue색
D는 (10 20)에 20x30크기와 Blue색
p1: (1, 2)
p2: (10, 11, 12)
p3: (20, 21, 22, 23)
Point3D (10, 11, 12)
Point4D (20, 21, 22, 23)
Point4D (20, 21, 22, 23)
p2: (4, 5, 6, 23)
p3: (4, 5, 6, 23)
p4: (4, 5, 6, 23)
```

학과 _____ 학번 _____ 이름 _____

**7. 위의 4-6번 예제에서 밑줄 친 virtual/override와 abstract/override그리고 sealed의 차이점이 무엇인지 예를 들어 간단히 서술하라. (10점)**

**virtual/override :**
Point, Size 클래스의 public virtual void Print() { ……..}는 이 클래스를 상속받는 파생클래스에서 이 메소드를 재정의할 수 있다는 의미로 virtual로 지정한다.
Point를 상속받은 Point3D, 그리고 Point3D를 상속받은 Point4D의 public override void Print() { …….. }에서 추가된 부분에 대한 구현을 재정의 한다.
Point, Size, Point3D, Point4D 클래스의 public override void ToString() { ..... }는 모든 클래스가 System.Object에서 상속받아서 각 클래스에 맞게 ToString()을 재정의해서 사용한다.

**abstract/override :**
추상클래스 A에 public abstract void Draw();는 abstract으로 지정된 메소드로 내부 구현을 할 수 없다.
A를 상속받은 B, C, D, E 클래스의 public override void Draw() { …….. }에서 내부 구현을 해야 한다.

**sealed :**
E 클래스의 public sealed override void Draw() { …….. }에서는 sealed하여 E 클래스를 상속하여 사용하는 클래스에서 이 메소드를 더 이상 재정의해서 사용할 수 없음을 지정한다.

**8. 위의 Point 클래스를 참조하여 값형식과 참조형식을 비교한다. 이 프로그램이 실행되는 출력결과를 써라. (10점)**

```
static void Main()
{
        int val1 = 5;
        int val2 = 10;
        if (val1 == val2)
            Console.WriteLine("val1 == val2");
        else
            Console.WriteLine("val1 != val2");

        object val3 = val1;
        object val4 = val1;
        if (val3 == val4)
            Console.WriteLine("val3 == val4");
        else
            Console.WriteLine("val3 != val4");

        int val5 = (int)val3;
        int val6 = (int)val4;
        if (val5 == val6)
            Console.WriteLine("val5 == val6");
        else
            Console.WriteLine("val5 != val6");

        Point pos1 = new Point(val1, val2);
        Point pos2 = new Point(val1, val2);
        if (pos1 == pos2)
            Console.WriteLine("pos1 == pos2");
        else
            Console.WriteLine("pos1 != pos2");
```

학과 _____ 학번 _____ 이름 _____

```
            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");

            Point pos3 = pos1;
            Point pos4 = pos1;
            if (pos3 == pos4)
                Console.WriteLine("pos3 == pos4");
            else
                Console.WriteLine("pos3 != pos4");

            object pos5 = pos3;
            object pos6 = pos4;
            if (pos5 == pos6)
                Console.WriteLine("pos5 == pos6");
            else
                Console.WriteLine("pos5 != pos6");

            Point pos7 = (Point) pos6;
            if (pos1 == pos7)
                Console.WriteLine("pos1 == pos7");
            else
                Console.WriteLine("pos1 != pos7");
            if (pos2 == pos7)
                Console.WriteLine("pos2 == pos7");
            else
                Console.WriteLine("pos2 != pos7");

            Console.WriteLine();
        }
    }
```

**출력결과:**

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

학과 _____ 학번 _____ 이름 _____

**9. IEquatable와 IComparable 인터페이스를 간단히 설명하라. 다음 Point 클래스에 IEquatable<T>와 IComparable<T> 인터페이스를 상속받아 구현(implement)하라. (10점)**

```
interface IEquatable<T>
{
    bool Equals(T obj);
}
interface IComparable<T>
{
    int CompareTo(T obj);
}
class Point : IEquatable<Point>, IComparable<Point>
{

// 중간 생략 ...
        // Point 클래스의 X와 Y가 모두 같으면 true 아니면 false
        public bool Equals(Point other)
        {
            if (this.X == other.X && this.Y == other.Y)
                return true;
            return false;
        }
        // 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.

학과 _____ 학번 _____ 이름 _____

**10. 다음 Main 메소드에 밑줄 친 네 줄의 코드를 TestA, TestB, Print, PrintThree 메소드를 사용하여 수정하라. (5점)**

```
class NumValue
{
        public int A { get; set; }
        public static int B{ get; set; }
        public const int THREE=3;
        public NumValue(int a, int b) { A = a; B = b; }
        public bool TestA(int value) { return A > value; }
        public static bool TestB(int value) { return B == value; }
        public void Print() { Console.WriteLine("a={0} b={1}", A, B); }
        public static void PrintThree() { Console.WriteLine("Three!"); }
}
class NumValueTest
{
        static void Main() {
            NumValue n = new NumValue(5, 3);
            if (n.A > NumValue.THREE)
                Console.WriteLine("a={0} b={1}", n.A, NumValue.B);
            if (NumValue.B == NumValue.THREE)
                Console.WriteLine("Three!");
        }
}
```

```
        if (n.TestA(NumValue.THREE))
            n.Print();
        if (NumValue.TestB(NumValue.THREE))
            NumValue.PrintThree();
```

학과 _____ 학번 _____ 이름 _____

**11. 다음은 원하는 온도변환 방법을 지정하여 사용할 수 있는 프로그램이다. 빈 칸을 채워라. (5점)**

```
public delegate double Conversion(double from);

class DelegateTest
{
        static ___double___ F2C(___double f___)
        {
            return (f - 32.0) * (5.0 / 9.0);
        }
        static ___double___ C2F(___double c___)
        {
            return ((9.0 / 5.0) * c + 32.0);
        }
        // factory method for returning delegate
        static public Conversion GetConversionMethod(string conversionType)
        {
            Conversion conversion;
            switch (conversionType)
            {
                case "C2F":
                    conversion = ___new Conversion(C2F);
                    break;
                case "F2C":
                    conversion = ___new Conversion(F2C);
                    break;
                default:
                    throw new ArgumentException("알수없는 conversion type: "+conversionType);
            }
            return conversion;
        }
        // user input to temperature and type of conversion
        static public void GetUserInput(out double temperature, out string conversionType)
        {
            Console.WriteLine("Temperature Converter");
            Console.WriteLine("C2F: Celsius => Fahrenheit");
            Console.WriteLine("F2C: Fahrenheit => Celsius");
            Console.Write("원하는 온도변환을 입력하시오 (C2F or F2C):  ");
            conversionType = Console.ReadLine();
            Console.Write("변환하고자하는 온도를 입력하시오:  ");
            temperature = Double.Parse(Console.ReadLine());
        }
        static public void Main()
        {
            double temperature;
            string conversionType;
            GetUserInput(out temperature, out conversionType);
            Conversion doConversion = GetConversionMethod(conversionType);
            double result = doConversion(temperature);
            Console.WriteLine("\n conversionType={0} Temperature={1} Result={2}\n",
                                    conversionType, temperature, result);
        }
}
```

-끝-