

기말고사

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 비밀번호(숫자4자리)를 기입하면 성적공고시 학번대신 비밀 번호를 사용할 것임.

1. 다음 Windows Forms 응용프로그램을 보고, 아래의 물음에 답하라. (30점)

```
partial class Form1
{
    private void InitializeComponent()
    {
        this.label1 = new System.Windows.Forms.Label();
        this.label2 = new System.Windows.Forms.Label();
        this.label3 = new System.Windows.Forms.Label();
        this.label4 = new System.Windows.Forms.Label();
        this.textBox1 = new System.Windows.Forms.TextBox();
        this.textBox2 = new System.Windows.Forms.TextBox();
        this.textBox3 = new System.Windows.Forms.TextBox();
        this.textBox4 = new System.Windows.Forms.TextBox();
        this.listView1 = new System.Windows.Forms.ListView();
        this.columnHeader1 = new System.Windows.Forms.ColumnHeader();
        this.columnHeader2 = new System.Windows.Forms.ColumnHeader();
        this.columnHeader3 = new System.Windows.Forms.ColumnHeader();
        this.columnHeader4 = new System.Windows.Forms.ColumnHeader();
        this.button1 = new System.Windows.Forms.Button();
        this.button2 = new System.Windows.Forms.Button();
        this.button3 = new System.Windows.Forms.Button();
        this.SuspendLayout();

        this.label1.AutoSize = true;
        this.label1.Location = new System.Drawing.Point(29, 144);
        this.label1.Name = "label1";
        this.label1.Size = new System.Drawing.Size(29, 12);
        this.label1.Text = "이름";

        this.label2.AutoSize = true;
        this.label2.Location = new System.Drawing.Point(31, 170);
        this.label2.Name = "label2";
        this.label2.Size = new System.Drawing.Size(16, 12);
        this.label2.Text = "ID";

        this.label3.AutoSize = true;
        this.label3.Location = new System.Drawing.Point(213, 140);
        this.label3.Name = "label3";
        this.label3.Size = new System.Drawing.Size(53, 12);
        this.label3.Text = "전화번호";
    }
}
```

학과 _____ 학번 _____ 이름 _____ 비밀번호 _____

```
this.label4.AutoSize = true;
this.label4.Location = new System.Drawing.Point(214, 170);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(29, 12);
this.label4.Text = "주소";

this.textBox1.Location = new System.Drawing.Point(87, 134);
this.textBox1.Name = "textBox1";
this.textBox1.Size = new System.Drawing.Size(100, 21);

this.textBox2.Location = new System.Drawing.Point(87, 162);
this.textBox2.Name = "textBox2";
this.textBox2.Size = new System.Drawing.Size(100, 21);

this.textBox3.Location = new System.Drawing.Point(270, 134);
this.textBox3.Name = "textBox3";
this.textBox3.Size = new System.Drawing.Size(100, 21);

this.textBox4.Location = new System.Drawing.Point(270, 163);
this.textBox4.Name = "textBox4";
this.textBox4.Size = new System.Drawing.Size(100, 21);

this.listView1.Columns.AddRange(new System.Windows.Forms.ColumnHeader[] {
this.columnHeader1,
this.columnHeader2,
this.columnHeader3,
this.columnHeader4});
this.listView1.Location = new System.Drawing.Point(31, 13);
this.listView1.Name = "listView1";
this.listView1.Size = new System.Drawing.Size(342, 115);
this.listView1.UseCompatibleStateImageBehavior = false;
this.listView1.View = System.Windows.Forms.View.Details;
this.columnHeader1.Text = "이름";
this.columnHeader2.Text = "ID";
this.columnHeader3.Text = "전화번호";
this.columnHeader4.Text = "주소";

this.button1.Location = new System.Drawing.Point(122, 201);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(124, 23);
this.button1.Text = "리스트 Serialize";
this.button1.UseVisualStyleBackColor = true;
this.button1.Click += new System.EventHandler(this.button1_Click);

this.button2.Location = new System.Drawing.Point(247, 201);
this.button2.Name = "button2";
this.button2.Size = new System.Drawing.Size(126, 23);
this.button2.Text = "리스트 Deserialize";
this.button2.UseVisualStyleBackColor = true;
this.button2.Click += new System.EventHandler(this.button2_Click);

this.button3.Location = new System.Drawing.Point(26, 201);
this.button3.Name = "button3";
this.button3.Size = new System.Drawing.Size(91, 23);
this.button3.Text = "Person 추가";
this.button3.UseVisualStyleBackColor = true;
this.button3.Click += new System.EventHandler(this.button3_Click);
```

학과 _____ 학번 _____ 이름 _____ 비밀번호 _____

```

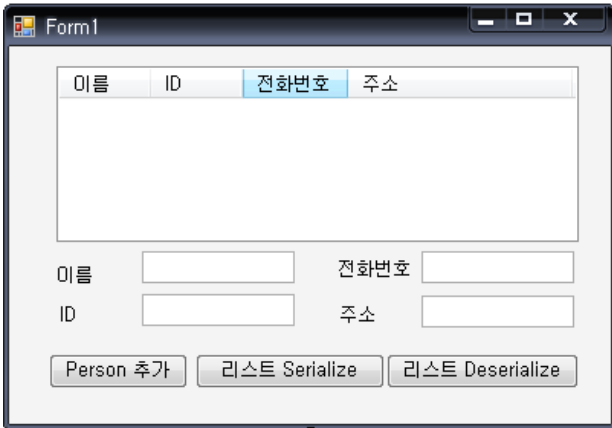
        this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 12F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(394, 247);
        this.Controls.Add(this.button3);
        this.Controls.Add(this.button2);
        this.Controls.Add(this.button1);
        this.Controls.Add(this.listView1);
        this.Controls.Add(this.textBox4);
        this.Controls.Add(this.textBox3);
        this.Controls.Add(this.textBox2);
        this.Controls.Add(this.textBox1);
        this.Controls.Add(this.label4);
        this.Controls.Add(this.label3);
        this.Controls.Add(this.label2);
        this.Controls.Add(this.label1);
        this.Name = "Form1";
        this.Text = "Form1";
        this.ResumeLayout(false);
        this.PerformLayout();
    }
}

public partial class Form1 : Form
{
    List<Person> pList = new List<Person>();
    public Form1()
    {
        InitializeComponent();
    }
    private void button1_Click(object sender, EventArgs e)
    {
        FileStream fs = new FileStream(@"C:\PersonList.xml", FileMode.Create, FileAccess.Write);
        XmlSerializer xs = new XmlSerializer(typeof(List<Person>));
        xs.Serialize(fs, pList);
        fs.Close();
    }
    private void button2_Click(object sender, EventArgs e)
    {
        FileStream fs = new FileStream(@"C:\PersonList.xml", FileMode.Open, FileAccess.Read);
        XmlSerializer xs = new XmlSerializer(typeof(List<Person>));
        pList = (List<Person>)xs.Deserialize(fs);
        fs.Close();

        listView1.Items.Clear();
        foreach (Person p in pList)
        {
            listView1.Items.Add(p.ToListViewItem());
        }
    }
    private void button3_Click(object sender, EventArgs e)
    {
        // int.TryParse 를 이용하여 textBox2.Text값이 Int32로 변환이 가능하다면
        // Person 개체를 생성하여 pList 와 listView1 에 추가한다.
    }
}

```

1) InitializeComponent()에서 생성한 컨트롤을 그림으로 그려서 나타내라. (10점)



2) button1 컨트롤의 Click 이벤트가 발생시 그 실행결과를 서술하라. (5점)

- button1 버튼에 Click 이벤트를 발생 시, button1_Click이라는 이벤트 핸들러가 호출
- C:\WPersonList.xml 파일을 생성하여 pList (Person 리스트)를 XMLSerializer를 이용하여 Serialize 한다.

3) button2 컨트롤의 Click 이벤트가 발생시 그 실행결과를 서술하라. (5점)

- button2 버튼에 Click 이벤트를 발생 시, button2_Click이라는 이벤트 핸들러가 호출
- C:\WPersonList.xml 파일을 읽어서 XMLSerializer를 이용하여 Deserialize 하여 pList (Person 리스트)를 얻는다. 그리고, listView1에 pList에 있는 Person 정보를 리스트 아이템으로 추가하여 출력한다.

4) button3_Click 이벤트 핸들러의 내부 코드를 작성하라. (10점)

```
private void button3_Click(object sender, EventArgs e)
{
    int value;
    if (int.TryParse(textBox2.Text, out value))
    {
        Person p = new Person(textBox1.Text, value, textBox3.Text, textBox4.Text);
        pList.Add(p);
        listView1.Items.Add(p.ToListViewItem());
    }
}
```

2. 다음은 GraphicsPath 클래스를 사용한 Scribble 그림 그리기 프로그램이다. 아래의 물음에 답하라. (30점)

```

public partial class Form1 : Form
{
    Graphics g;
    Pen p;
    GraphicsPath curve;
    Point prevPos;

    public Form1()
    {
        InitializeComponent();
        g = this.CreateGraphics();
        p = new Pen(Color.Black);
        curve = new GraphicsPath();
    }
    private void Form1_MouseDown(object sender, MouseEventArgs e)
    {
        if (e.Button == MouseButton.Left)
        {
            prevPos.X = e.X;
            prevPos.Y = e.Y;
            curve.StartFigure();
        }
    }
    private void Form1_MouseMove(object sender, MouseEventArgs e)
    {
        if (Capture && e.Button == MouseButton.Left)
        {
            Point currPos = new Point(e.X, e.Y);
            curve.AddLine(prevPos, currPos);
            g.DrawLine(p, prevPos, currPos);
            prevPos = currPos;
        }
    }
    private void Form1_Paint(object sender, PaintEventArgs e)
    {
        Graphics g = e.Graphics;
        g.DrawPath(p, curve);
    }
    private void Form1_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (e.KeyChar == (char)27)
        {
            this.Close();
        }
    }
    private void Form1_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Space)
        {
            g.Clear(Color.White);
        }
        else if (e.KeyCode == Keys.D1)
        {
            p.Dispose();
        }
    }
}

```

학과 _____ 학번 _____ 이름 _____ 비밀번호 _____

```

        p = new Pen(Color.Red);
    }
    else if (e.KeyCode == Keys.D2)
    {
        p.Dispose();
        p = new Pen(Color.Green);
    }
    else if (e.KeyCode == Keys.D3)
    {
        p.Dispose();
        p = new Pen(Color.Blue);
    }
}
}
}

```

1) 위의 예제에서 밑줄 친 부분 (즉, CreateGraphics()와 e.Graphics)이 전체 코드에서 어떻게 작동되고 사용되는지 설명하라. (10점)

-Graphics 클래스는 직접 객체를 생성할 수 없는 클래스이기 때문에, Paint가 아닌 이벤트에서 그림 그리기를 할 경우 정적 메서드인 CreateGraphics()를 이용해서 그래픽 객체를 생성해서 사용해야 한다.
 -Form1() 생성자 안에서 g = this.CreateGraphics()는 폼의 객체로부터 그래픽 생성 메서드인 CreateGraphics()를 이용하여 전역적 그래픽 객체 g를 생성하여서 Form1_MouseMove나 Form1_KeyDown에서 그림을 그리고자 할 때 사용하였다.
 -Form1_Paint 이벤트 핸들러 안에서 PaintEventArgs e로부터 받은 Graphics g = e.Graphics는 폼의 그래픽 객체를 참조한 것을 사용하여 그림을 그린다.

2) Form1_KeyPress 이벤트 핸들러와 Form1_KeyDown 이벤트 핸들러가 언제 호출되는 것인지, 호출 시 실행결과를 설명하라. (10점)

- Form1_KeyPress 이벤트 핸들러
 ASCII 코드에 해당하는 키(Key)가 눌려졌을 때 호출된다.
 ESC-키가 눌려졌을 때 폼을 닫는다 (즉, 프로그램 종료).
 - Form1_KeyDown 이벤트 핸들러
 수정자 키 (예: Shift, Alt, Ctrl)를 포함한 F1~F12, 방향키 등등 키가 눌려졌을 때 호출된다.
 SpaceBar-키가 눌려졌을 때 전체 폼 화면을 흰색으로 칠한다(지운다).
 숫자1-키가 눌려졌을 때 그림 그리는 펜의 색을 빨간색으로 바꾼다.
 숫자2-키가 눌려졌을 때 그림 그리는 펜의 색을 녹색으로 바꾼다.
 숫자3-키가 눌려졌을 때 그림 그리는 펜의 색을 파란색으로 바꾼다.

3) 위 예제에서 C-key가 눌려졌을 때 Pen의 색을 자유롭게 바꿀 수 있도록, Form1_KeyDown 이벤트 핸들러 안에 공통대화상자의 하나인 ColorDialog 클래스를 이용하여 작성하라. (10점)

```

private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    // 중간 생략..
    else if (e.KeyCode == Keys.C)
    {
        ColorDialog f = new ColorDialog();
        if (f.ShowDialog() == DialogResult.OK)
        {
            p.Dispose();
            p = new Pen(f.Color);
        }
    }
}

```

3. 다음 Point/Point3D/Point4D 프로그램을 참고하여, 아래의 물음에 답하라. (20점)

```

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

class Point3D : Point
{
    public Point3D() : this(0, 0, 0) {}
    public Point3D(int x, int y, int z) : base(x, y) { this.Z = z; }
    public int Z { get; set; }
    public void SetPosition(int x, int y, int z) { base.SetPosition(x, y); this.Z = z; }
    public override void Print() { Console.WriteLine("X={0} Y={1} Z={2}", X, Y, Z); }
    public override string ToString() { return (String.Format("{0}, {1}, {2}", X, Y, Z)); }
}

class Point4D : Point3D
{
    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 int W { get; set; }
    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("X={0} Y={1} Z={2} W={3}", X, Y, Z, W); }
    public override string ToString() { return (String.Format("{0}, {1}, {2}, {3}", X, Y, Z, W)); }
}

class PointTest
{
    static void Main(string[] args)
    {
        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();
        p1.SetPosition(1, 2);
        Console.WriteLine("p1: {0}", p1);
        p2.SetPosition(3, 4, 5);
        Console.WriteLine("p2: {0}", p2);
        p3 = (Point4D)p2;
        p3.SetPosition(6, 7, 8, 9);
        Console.WriteLine("p3: {0}", p3);
        Point3D p4 = (Point3D)p1;
        p4.SetPosition(10, 11, 12);
    }
}
    
```

```
        Console.WriteLine("p4: {0}", p4);
    }
}
```

1) 다형성 (Polymorphism)의 핵심인 upcasting과 downcasting이 무엇인지 간단히 설명하라. 그리고 위 예제 코드에서 upcasting과 downcasting의 예를 찾아서 적어라. (10점)

-Upcast는 `p1 = p2; p2 = p3;` 와 같이 하위 클래스의 객체를 상위 클래스의 객체로 암묵적 타입변환이 가능한 것을 말한다. 따라서, 업캐스팅하고 난후 `p1.Print();`는 실제로는 `Point3D` 클래스의 `Print`를 호출, `p2.Print();`도 마찬가지로 `Point4D` 클래스의 `Print`를 호출한다.

-Downcast는 `p3.SetPosition(6, 7, 8, 9);`를 하기 위하여 `p3 = (Point4D)p2;`로 상위 클래스의 객체를 명시적으로 타입캐스트 해서 하위 클래스의 객체 타입으로 바꿔주는 것을 말한다. 같은 예로 `Point3D p4 = (Point3D)p1;`의 경우도 `p1`은 원래 `Point3D` 객체이나 타입은 `Point` 클래스이므로 다운캐스팅하고 `p4.SetPosition(10, 11, 12)`를 사용할 수 있도록 했다.

2) 위 예제의 프로그램 실행 결과를 써라. 다형성 (Polymorphism)에 주의할 것. (10점)

```
p1: (1, 2)
p2: (10, 11, 12)
p3: (20, 21, 22, 23)
X=10 Y=11 Z=12
X=20 Y=21 Z=22 W=23
X=20 Y=21 Z=22 W=23
p1: (1, 2, 12)
p2: (3, 4, 5, 23)
p3: (6, 7, 8, 9)
p4: (10, 11, 12)
```

4. 다음은 `QuakeData`를 리스트로 관리하는 `QuakeDataManager` 클래스를 보여주고 있다. 아래 물음에 답하라. (20점)

```
class QuakeDataManager
{
    private List<QuakeData> qList = null;
    // 중간생략...
    public List<QuakeData> Find(QuakeDataRange r)
    {
        if (qList != null)
        {
            return qList.FindAll(delegate(QuakeData q)
            {
                if (q.Lat >= r.LatRange.LowerBound && q.Lat <= r.LatRange.UpperBound
                    && q.Lon >= r.LonRange.LowerBound && q.Lon <= r.LonRange.UpperBound
                    && q.Magnitude >= r.MagnitudeRange.LowerBound
                    && q.Magnitude <= r.MagnitudeRange.UpperBound
                    && q.Depth >= r.DepthRange.LowerBound && q.Depth <= r.DepthRange.UpperBound)
                    return true;
                return false;
            });
        }
        return null;
    }
}
```


학과 _____ 학번 _____ 이름 _____ 비밀번호 _____

```
public void Print()
{
    if (qList != null)
    {
        foreach(QuakeData q in qList)
        {
            System.Diagnostics.Trace.WriteLine(q);
        }
    }
}

public void SortList(int index)
{
    if (qList != null)
    {
        if (index == 0)
            qList.Sort();
        else if (index == 1)
            qList.Sort(new QuakeMonthComparer());
        else if (index == 2)
            qList.Sort(new QuakeDayComparer());
        else if (index == 3)
            qList.Sort(new QuakeTimeComparer());
        else if (index == 4)
            qList.Sort(new QuakeLatComparer());
        else if (index == 5)
            qList.Sort(new QuakeLonComparer());
        else if (index == 6)
            qList.Sort(new QuakeMagnitudeComparer());
        else if (index == 7)
            qList.Sort(new QuakeDepthComparer());
    }
}
}
```

1) Print 메소드의 기능을 설명하라. (5점)

-QuakeDataManager 클래스의 Print 메소드는 qList 안에 있는 모든 데이터 q를 QuakeData 클래스의 ToString 메소드를 사용하여 디버거 출력창 (Output window)에 출력해준다.

2) Find 메소드의 기능을 설명하라. (5점)

-QuakeDataManager 클래스의 Find 메소드는 qList 안에 있는 모든 데이터 q를 QuakeDataRange r의 조건에 일치하는 것을 (즉, Lat/Lon/Magnitude/Depth의 range가 주어진 조건 안에 들어서 true를 반환할 경우만) 모아서 새로운 List<QuakeData>로 반환해준다.

3) SortList 메소드가 동작하기 위해서 QuakeData 클래스에 IComparable<QuakeData> 인터페이스, QuakeMonthComparer, QuakeDayComparer, ... 등은 IComparer<QuakeData> 인터페이스가 구현되어야 한다. 직접 코드를 구현하여 설명하라. (10점)

```
class QuakeData : IComparable<QuakeData>
{
// 중간생략
    public int CompareTo(QuakeData other)
    {
        return this.Year.CompareTo(other.Year);
    }
}

class QuakeMonthComparer : IComparer<QuakeData>
{
    public int Compare(QuakeData p, QuakeData q)
    {
        return p.Month.CompareTo(q.Month);
    }
}

class QuakeDayComparer : IComparer <QuakeData>
{
    public int Compare(QuakeData p, QuakeData q)
    {
        return p.Day.CompareTo(q.Day);
    }
}

class QuakeTimeComparer : IComparer <QuakeData>
{
    public int Compare(QuakeData p, QuakeData q)
    {
        return p.Time.CompareTo(q.Time);
    }
}

class QuakeLatComparer : IComparer <QuakeData>
{
    public int Compare(QuakeData p, QuakeData q)
    {
        return p.Lat.CompareTo(q.Lat);
    }
}

class QuakeLonComparer : IComparer <QuakeData>
{
    public int Compare(QuakeData p, QuakeData q)
    {
        return p.Lon.CompareTo(q.Lon);
    }
}

class QuakeMagnitudeComparer : IComparer <QuakeData>
{
    public int Compare(QuakeData p, QuakeData q)
    {
        return p.Magnitude.CompareTo(q.Magnitude);
    }
}

class QuakeDepthComparer : IComparer <QuakeData>
{
    public int Compare(QuakeData p, QuakeData q)
    {
        return p.Depth.CompareTo(q.Depth);
    }
}
```

-끝-