

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 암호를 기입하면 성적공고시 학번대신 암호를 사용할 것임.

1. 다음 Windows Forms 응용프로그램을 보고, 아래의 물음에 답하라. (25점)

```
// Program.cs
using System.Windows.Forms;
```

```
static class Program
{
    static void Main()
    {
        Application.Run(new Form1());
    }
}
```

```
// Form1.cs
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }
    private void button1_Click(object sender, EventArgs e)
    {
        label1.Text = "Hello";
    }
}
```

```
// Form1.Designer.cs
public partial class Form1
{
    // 중간생략..
    private void InitializeComponent()
    {
        this.button1 = new System.Windows.Forms.Button();
        this.label1 = new System.Windows.Forms.Label();
        this.SuspendLayout();
        this.button1.Location = new System.Drawing.Point(85, 113);
        this.button1.Name = "button1";
        this.button1.Size = new System.Drawing.Size(75, 23);
        this.button1.Text = "Click Here!";
        this.button1.Click += new System.EventHandler(this.button1_Click);
        this.label1.AutoSize = true;
        this.label1.Location = new System.Drawing.Point(83, 81);
        this.label1.Name = "label1";
        this.label1.Size = new System.Drawing.Size(0, 24);
        this.ClientSize = new System.Drawing.Size(292, 268);
```

```
        this.Controls.Add(this.label1);
        this.Controls.Add(this.button1);
        this.Name = "Form1";
        this.Text = "Form1";
        this.ResumeLayout(false);
        this.PerformLayout();
    }
    private System.Windows.Forms.Button button1;
    private System.Windows.Forms.Label label1;
}
```

1) Application.Run(new Form1())가 수행하는 작업이 무엇인지 간단히 설명하라.

- Application 클래스는 윈도우 응용프로그램 그 자체를 나타내는 클래스로, 응용프로그램의 시작과 종료, 메시지 처리, 여러 가지 정보조사 등의 작업을 하는 정적 메소드와 속성들을 제공함
- Application.Run 메소드를 호출하여 윈도우 응용프로그램을 실행함
- Run 메소드는 인수로 전달받은 폼을 메인 폼으로 하여 이 폼이 닫힐 때까지 메시지 루프를 돌리며 실행을 함
- 현재 스레드에서 표준 윈도우 응용프로그램 메시지 루프의 실행을 시작하고 지정된 폼을 표시
- 폼이 닫힐 때 메시지 루프에 종료 메시지를 보냄
- 메시지 루프는 메시지를 하나씩 꺼내어 처리하기 위한 반복문으로 응용프로그램이 종료될 때까지 반복

2) partial class가 무엇인지 간단히 설명하라.

- partial 키워드를 사용하여 클래스나 구조체 정의 또는 인터페이스를 두 개 이상의 소스 파일로 분할 가능
- 대규모 프로젝트를 진행하는 경우 클래스를 개별 파일로 분할하면 여러 프로그래머가 동시에 작업을 수행할 수 있음
- 위의 예제처럼 Form1 클래스를 Form1.cs과 Form1.Designer.cs로 분할하여, 자동으로 생성된 소스를 사용하여 작업하는 경우 소스파일을 다시 만들지 않고 클래스에 코드를 추가 가능

3) InitializeComponent()가 수행하는 작업이 무엇인지 간단히 설명하라.

- 윈도우 폼 디자이너에서 작업한 것에 따라 자동 생성해주는 코드
- 윈도우 폼을 응용프로그램에 연결해주는 코드가 포함
- 윈도우 폼과 이벤트 핸들러(Event Handler)를 연결하는 코드가 포함 (Application.Run 메소드가 이벤트 처리 스레드를 만드는 역할을 함)
- 폼 클래스의 생성자가 딱 한번 호출

4) this.button1.Click += new System.EventHandler(this.button1_Click); 가 수행하는 작업이 무엇인지 설명하라.

- 윈도우 폼의 button1 버튼에 Click 이벤트를 발생 시, button1_Click이라는 이벤트 핸들러를 호출할 수 있도록 등록
- 일반적으로 이벤트 핸들러의 메소드명은 컨트롤 이름(button1)에 밑줄 붙히고() 이벤트명(Click)을 명시, e.g. button1_Click

5) 이 응용프로그램의 실행결과를 그림으로 그려서 설명하라.

- 윈도우 폼의 버튼을 Click하면, Hello! 라는 레이블이 출력



2. GDI+를 이용한 화면출력에서, Paint 이벤트, OnPaint(), Paint 이벤트 핸들러, CreateGraphics(), Invalidate()가 어떻게 작동되고 사용되는지 설명하라. (15점)

-Paint 이벤트는 폼이 화면에 처음 표시될 때, 폼이 최소화되었다가 최대화 버튼을 클릭할 때 폼의 크기를 조정할 때, 폼이 다른 윈도우로 가려졌다가 전면에 나타날 때, 프로그램에서 Invalidate() 메소드를 호출할 때 발생하는 이벤트

-OnPaint() 메소드는 Paint 이벤트가 발생할 때 수행하는 것으로 OnPaint() 메소드 안에서 Paint 이벤트 핸들러를 호출하므로, Paint 이벤트 핸들러보다 높은 우선순위

e.g., protected override void OnPaint(PaintEventArgs e)

```
{ Graphics g = e.Graphics; // 그래픽 객체 참조  
... }
```

-Paint 이벤트 핸들러,

e.g., private void pictureBox1_Paint(object sender, PaintEventArgs e)

```
{ Graphics g = e.Graphics; // 그래픽 객체 참조  
... }
```

-만약 다른 메소드에서 폼에 그림을 그려야 할 경우, CreateGraphics() 메소드를 사용하여 Graphics 객체를 받아서 그리기에 사용

e.g., private void Form1_MouseClick(object sender, MouseEventArgs e)

```
{ Graphics g = this.CreateGraphics(); // Form1의 그래픽 객체 참조  
... }
```

-프로그램에서 원하는 시점에 화면 출력을 하고자 할 경우, Invalidate() 메소드를 사용해서 Paint 이벤트를 발생시켜서 OnPaint와 Paint 이벤트 핸들러를 호출

e.g., private void Form1_KeyPress(object sender, KeyPressEventArgs e)

```
{ ...  
    Invalidate(); // Invalidate을 호출하여 Paint이벤트를 발생시킴  
}
```

3. 다음은 윈도우 폼 응용프로그램이다. 아래의 물음에 답하라. (20점)

```
public partial class Form1 : Form  
{  
    public List<Point> pList;  
  
    public Form1()  
    {  
        InitializeComponent();  
        pList = new List<Point>();  
    }  
  
    private void LoadFile(string path)  
    {  
        // 중간 생략..  
    }  
  
    private void WriteFile(string path)  
    {  
        // 중간 생략..  
    }  
  
    private void SortList(int index)  
    {  
        // 중간 생략.. (pList 정렬)  
    }  
  
    private void toolStripButton1_Click(object sender, EventArgs e)  
    {  
        // 중간생략..  
        if (!formExist)  
        {
```

```

        Form2 f = new Form2();
        f.Owner = this;
        f.Show();
    }
}

private void toolStripButton2_Click(object sender, EventArgs e)
{
    Form3 f = new Form3();
    if (f.ShowDialog() == DialogResult.OK)
    {
        Point p = new Point();
        p.X = int.Parse(f.textBox1.Text);
        p.Y = int.Parse(f.textBox2.Text);
        pList.Add(p);
        listView1.Items.Add(p.ToListViewItem());
    }
}

private void toolStripButton3_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        LoadFile(openFileDialog1.FileName);
    }
}

private void toolStripButton4_Click(object sender, EventArgs e)
{
    if (saveFileDialog1.ShowDialog() == DialogResult.OK)
    {
        WriteFile(saveFileDialog1.FileName);
    }
}

private void listView1_ColumnClick(object sender, ColumnClickEventArgs e)
{
    this.listView1.ListViewItemSorter = new ListViewItemComparer(e.Column);
    this.listView1.Sort();
    SortList(e.Column);
}

private void listView1_DoubleClick(object sender, EventArgs e)
{
    listView1.Items.Remove(listView1.FocusedItem);
    pList.RemoveAt(listView1.FocusedItem.Index);
}
}

```

1) listView1_ColumnClick과 listView1_DoubleClick는 언제 호출되는 것인지, 호출 시 수행하는 작업이 무엇인지 설명하라.

- listview1_ColumnClick은 리스트뷰에서 컬럼헤더를 클릭시 호출되는 핸들러로, listView1과 pList를 정렬(Sort)함
- listview1_DoubleClick은 리스트뷰에 아이템을 더블클릭시 호출되는 핸들러로, 선택한 아이템을 listView1과 pList에서 삭제(Remove) 함

2) 프로그램에서 공통 대화상자 (Common Dialog) 예를 찾아서 무엇인지 설명하고, 그 외에 다른 공통 대화상자를 서술하라.

-공통 대화상자는 닷넷에서 제공하는 일반적인 대화상자
-이 예제에서는 파일열기 (OpenFileDialog), 파일저장 (SaveFileDialog)가 있음
-그 외에 색상설정 (ColorDialog), 글꼴설정 (FontDialog), 페이지설정 (PageSetupDialog), 프린트설정 (PrintDialog)가 있음

3) 프로그램에서 모달형(Modal)과 비모달형(Modeless) 사용자 정의 대화상자가 무엇인지 예를 찾아서 설명하고, 작성 시 특징을 서술하라.

-사용자 정의 대화상자는 모달형 (Modal Dialog)과 비모달형 (Modeless Dialog)로 나뉨

-Form2는 Modeless 대화상자로 Show() 메소드를 사용함
Form2 대화상자가 표시된 직후 Show 메소드 다음 코드가 실행됨
Form2 대화상자를 소유하는 폼을 Owner 속성으로 설정하여 접근할 수 있도록 함
Form1의 pList는 public으로 설정해야 Form2에서 접근이 가능해짐

-Form3는 Modal 대화상자로 ShowDialog() 메소드를 사용함
Form3 대화상자가 닫힐 때까지 ShowDialog 메소드의 다음 코드가 실행되지 않음
Form3 대화상자는 닫힐 때 반환값으로, DialogResult.OK, DialogResult.Cancel 등을 설정해야 함
Form3의 textBox1과 textBox2는 public으로 설정해야 Form1에서 접근이 가능해짐

4) pList를 직렬화(Serialization)하기 위한 방법을 서술하라.

-BinaryFormatter 방식
Point 클래스는 [Serializable] 특성을 명시하고
BinaryFormatter bf = new BinaryFormatter(); bf.Serialize(fs, pList);

-XmlSerializer 방식
Point 클래스는 [XmlType] [XmlAttribute] 등의 특성을 명시하고
XmlSerializer xs = new XmlSerializer(typeof(List<Point>)); xs.Serialize(fs, pList);

4. 다음 Point/Point3D 프로그램을 참고하여, 아래의 물음에 답하라. (30점)

```
class Point
{
    protected static int count = 0;
    protected int x, y;
    public Point() : this(0, 0) {}
    public Point(int x, int y) { this.x = x; this.y = y; count++; }
    ~Point() { count--; }
    public int X
    {
        get { return this.x; }
        set { this.x = value; }
    }
    public int Y
    {
        get { return this.y; }
        set { this.y = value; }
    }
}
```

```

    }
    public static int Count
    {
        get { return count; }
    }

    public void SetPosition(int x, int y) { this.x = x; this.y = y; }
    public void Move(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
{
    protected int z;
    public Point3D() : this(0, 0, 0) { }
    public Point3D(int x, int y, int z) : base(x, y) { this.z = z; }
    ~Point3D() { }
    public int Z
    {
        get { return this.z; }
        set { this.z = value; }
    }

    public void SetPosition(int x, int y, int z) { base.SetPosition(x, y); this.z = z; }
    public void Move(int x, int y, int z) { base.Move(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 PointTest
{
    static void Swap(ref Point3D p1, ref Point3D p2) // 두 Point3D를 서로 swap하는 메소드
    {
        Point3D p;
        p = p1;
        p1 = p2;
        p2 = p;
    }

    static void Main(string[] args)
    {
        Point3D p1 = new Point3D();
        Point3D p2 = new Point3D();
        p1.SetPosition(10, 20);
        p2.SetPosition(30, 40, 50);

        object r1 = p1;
        object r2 = p2;

        Swap(ref p1, ref p2); // p1과 p2를 swap
        Console.WriteLine("Swap : {0}, {1}", p1, p2);

        if (r1 == (object)p2)
            Console.WriteLine("r1 == p2");
        else
            Console.WriteLine("r1 != p2");

        Point p3 = new Point(20, 30);
        Point p4 = new Point(20, 30);
    }
}

```

```
if (p3 == p4) // p3==p4가 TRUE
    Console.WriteLine("p3 == p4");

if ((object)p3 == (object)p4)
    Console.WriteLine("object p3 == object p4");
else
    Console.WriteLine("object p3 != object p4");

Point p5 = p1;
p5.Print();
}
}
```

1) Point 클래스와 Point3D 클래스의 X, Y, Z, Count 속성(Property)을 구현하라 (코드에 직접 작성할 것).

2) PointTest에 Point3D 클래스의 swap 메소드를 구현하라 (코드에 직접 작성할 것). 그리고 두 개의 Point3D 객체를 swap하고 난 후의 결과를 설명하라.

-Swap(ref p1, ref p2); // 참조형 객체를 참조에 의한 전달을 사용하여 함수밖에서도 변경된 값을 가짐
-Console.WriteLine("Swap : {0}, {1}", p1, p2); // 따라서, Swap : (30, 40, 50), (10, 20, 0)

3) PointTest에서 if (r1==(object)p2)의 결과가 어떻게 나오는 지 간단히 설명하라.

-Swap(ref p1, ref p2)에 의해 p1과 p2 포인터가 바뀌었으므로, r1과 p2는 동일한 객체를 가리킴
-따라서, r1 == (object)p2는 TRUE

4) PointTest에서 if (p3==p4)가 TRUE가 될 수 있도록 Point 클래스에 IEquatable<Point>와 Equality operator (==와 !=)를 구현하라.

```
class Point : IEquatable<Point>
{
    // 중간 생략
    public bool Equals(Point other)
    {
        if (this.x == other.x && this.y == other.y)
            return true;
        return false;
    }
    public override int GetHashCode() { return x ^ y; }
    public override bool Equals(object obj)
    {
        if (!(obj is Point))
            return false;
        return Equals((Point)obj);
    }
    public static bool operator==(Point p1, Point p2) { return p1.Equals(p2); }
    public static bool operator!=(Point p1, Point p2) { return !p1.Equals(p2); }
}
```

5) PointTest에서 if ((object)p3==(object)p4)의 결과가 어떻게 나오는 지 간단히 설명하라.

-클래스는 참조형이므로 참조가 같지 않으며 == 는 FALSE 반환

-p3와 p4에 x, y값은 같으나, p3와 p4가 동일한 참조를 하지 않으므로 결과는 FALSE로 나옴

6) p5.Print()의 결과가 어떻게 나오는 지 간단히 설명하라.

-p5의 실질적인 객체는 p1 (즉, Point3D 클래스)이므로, late binding에 의하여 Point3D의 Print()가 호출됨

-따라서, 결과는 X=30 Y=40 Z=50을 출력함

5. 다음은 Point를 리스트로 관리하는 PointList 클래스를 보여주고 있다. 아래 물음에 답하라. (20점)

```
class PointList
{
    private List<Point> pList = new List<Point>();

    public void Add(Point p)
    {
        pList.Add(p);
    }

    public void Remove(Point p)
    {
        pList.Remove(p);
    }

    public void Clear()
    {
        pList.Clear();
    }

    public Point Find(Point p) // p와 같은 것이 있는지 찾아서 있으면 객체를 넘겨줌
    {
        return pList.Find(delegate(Point o)
        {
            return o == p;
        });
    }

    public int FindIndex(Point p) // p와 같은 것이 있는지 찾아서 index를 넘겨줌
    {
        return pList.FindIndex(delegate(Point o)
        {
            return o == p;
        });
    }

    public int GetCount()
    {
        return pList.Count;
    }

    private void Print(Point p)
    {
        Console.WriteLine(p);
    }
}
```



```

public void Print()
{
    pList.ForEach(Print);
}

public void Sort()
{
    pList.Sort(); // Point나 Point3D의 x값 순서대로 정렬
}
}

```

1) Find와 FindIndex 메소드를 구현하라 (코드에 직접 작성할 것).

2) Sort가 동작하기 위해서 Point 클래스와 Point3D 클래스에 추가적인 코드가 필요하다. 무엇인지 설명하라.

```

class Point : IEquatable<Point>, IComparable<Point>
{
    // 중간생략
    public int CompareTo(Point other)
    {
        return this.x.CompareTo(other.x);
    }
}

class Point3D : IEquatable<Point3D>, IComparable<Point3D>
{
    // 중간생략
    public int CompareTo(Point3D other)
    {
        return this.x.CompareTo(other.x);
    }
}

```

3) PointList를 테스트하는 PointTest의 Main 메소드에 다음 실행 결과를 출력하라.

```

class PointTest
{
    static void Main(string[] args)
    {
        // 위의 PointTest에 Main 메소드를 그대로 사용하고, PointList 테스트를 추가
        PointList aList = new PointList();
        aList.Add(p1);
        aList.Add(p2);
        aList.Add(p3);
        aList.Add(p4);
        aList.Add(p5);
        aList.Add(new Point3D(110, 210, 310));
        aList.Add(new Point3D(50, 60, 70));
        aList.Print();
        Console.WriteLine();
        aList.Sort();
        aList.Print();
        Console.WriteLine();
        aList.Remove(p3);
        aList.Print();
        Console.WriteLine();
        int f = aList.FindIndex(new Point3D(50, 60, 70));
        Console.WriteLine(f);
    }
}

```

학과 _____

학번 _____

이름 _____

출력결과:

(30, 40, 50)
(10, 20, 0)
(20, 30)
(20, 30)
(30, 40, 50)
(110, 210, 310)
(50, 60, 70)

(10, 20, 0)
(20, 30)
(20, 30)
(30, 40, 50)
(30, 40, 50)
(50, 60, 70)
(110, 210, 310)

(10, 20, 0)
(20, 30)
(30, 40, 50)
(30, 40, 50)
(50, 60, 70)
(110, 210, 310)

4

-끝-