

## 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 성적공고 시 학번대신 비밀번호를 사용할 것임.

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

```
public class A
{
    protected Graphics g;
    protected Pen pen;
    protected Rectangle area;
    protected Point[] vertices;
    protected A(Graphics g, Pen p, Rectangle area)
    {
        this.g = g;
        this.pen = p;
        this.area = area;
    }
    public void Draw()
    {
        if (pen != null)
            this.g.DrawPolygon(pen, vertices);
    }
}

public class B : A
{
    public B(Graphics g, Pen p, Rectangle area) : base(g, p, 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, area.Y + area.Height);
        vertices[3] = new Point(area.X, area.Y + area.Height);
    }
}
```

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

```
    }
}
public class C : A
{
    public C(Graphics g, Pen p, Rectangle area) : base(g, p, area)
    {
        vertices = new Point[3];
        vertices[0] = new Point(area.X, area.Y);
        vertices[1] = new Point(area.X + area.Width, area.Y + area.Height);
        vertices[2] = new Point(area.X - area.Width, area.Y + area.Height);
    }
}
public class D : A
{
    public D(Graphics g, Pen p, Rectangle area) : base(g, p, 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 partial class MainForm : Form
{
    enum DrawType { LINE, RECTANGLE, TRIANGLE }
    DrawType mode;
    Pen pen = null;
    Rectangle area = null;
    private void Form1_Paint(object sender, PaintEventArgs e)
    {
        Graphics g = e.Graphics;
        pen = new Pen(color, thickness); // 색깔과 두께가 정의된 펜
        area = new Rectangle(startPoint.X, startPoint.Y, cx, cy); // 마우스 시작점과 끝점으로 된 크기의 영역
        if (drag) {
            A s;
            switch (mode) {
                // 도형 그리기
            }
        }
    }
}
```

1) 위 예제의 밑줄 친 부분에 `protected A(Graphics g, Pen p, Rectangle area)` 를 사용하는 이유는 무엇인가? 만약 `protected` 대신 `private`을 사용한다면 무슨 문제가 발생할까? (5점)

외부에서 A 클래스의 객체를 직접 생성하지 못하게 하면서, A 클래스를 상속 받은 B, C, D 파생 클래스에서 상속받아 사용할 수 있도록 함. Private을 사용하면 B, C, D 클래스에서 호출할 수 없으므로 `inaccessible due to its protected level` 컴파일 에러 발생.

2) 위 예제의 밑줄 친 부분에 `protected Graphics g; protected Pen pen; protected Rectangle area; protected Point[] vertices;` 을 사용하는 이유는 무엇인가? (5점)

1)과 같이, 외부에서 g, pen, area, vertices 객체를 직접 생성하지 못하게 하면서, A 클래스를 상속 받은 B, C, D 파생 클래스에서 상속받아 사용할 수 있도록 함.

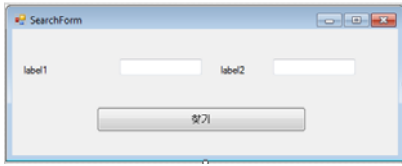
3) 위 예제의 밑줄 친 부분 `public B(Graphics g, Pen p, Rectangle area) : base(g, p, area)` 대신에 `public B(Graphics g, Pen p, Rectangle area)`로 호출했다면, 무슨 문제가 발생하는가? (5점)

`public B(Graphics g, Pen p, Rectangle area)`는 `public B(Graphics g, Pen p, Rectangle area) : base()`를 호출하므로, 즉 A 클래스에 기본생성자 A()가 없으므로, 컴파일 에러 발생

4) 위 예제의 switch 구문의 내부를 채워라. (10점)

```
A s;  
switch (mode)           // 도형 그리기  
{  
    case Drawtype.Line:  
        s = new D(g, pen, area);  
        s.Draw();  
        break;  
    case Drawtype.Rectangle:  
        s = new B(g, pen, area);  
        s.Draw();  
        break;  
    case Drawtype.Triangle:  
        s = new C(g, pen, area);  
        s.Draw();  
        break;  
}
```

## 2. 다음 응용프로그램을 보고, 아래의 물음에 답하라. (40점)



```
public partial class MainForm : Form
```

```
{
    List<MeteorologicalData> mList = null;
    public delegate void LogHandler(string status);
    public event LogHandler Log;
    protected void OnLog(string status)
    {
        if (Log != null) {
            Log(status);
        }
    }
    private void Logger(string info)
    {
        System.Diagnostics.Trace.WriteLine(info);
    }

    public MainForm()
    {
        InitializeComponent();
        mList = new List<MeteorologicalData>();
        Log += new LogHandler(Logger);
    }

    private void toolStripButton4_Click(object sender, EventArgs e)
    {
        SearchForm f = new SearchForm();
        f.Owner = this;
        f.Text = "날씨정보 찾기 ";
        f.label1.Text = "기온구간 between: ";
        f.label2.Text = " and ";
        f.ShowDialog();
        if (f.DialogResult == DialogResult.OK) {
            double value, min=-999.0, max=999.0;
            if (double.TryParse(f.textBox1.Text, out value))
                min = value;
            if (double.TryParse(f.textBox2.Text, out value))
                max = value;
            mList = GetListByTemperatureRange_AnonymousMethod(min, max);
            listView1.Items.Clear();
            foreach (MeteorologicalData m in mList) {
                listView1.Items.Add(m.ToListViewItem());
            }
            listView1.Refresh();
            string message = f.Text + f.label1.Text + min.ToString() + f.label2.Text + max.ToString();
            OnLog(message);
        }
    }
}
```

(1)

(2)

(3)

(4)

(5)

```
public List<MeteorologicalData> GetListByTemperatureRange_AnonymousMethod(double min, double max)
{
    List<MeteorologicalData> aList = mList.FindAll(delegate(MeteorologicalData m)
                                                { return m.Temperature >= min && m.Temperature <= max; });

    return aList;
}
```

1) GetListByTemperatureRange\_AnonymousMethod 대신에 사용할 수 있는 GetListByTemperatureRange\_Lambda 메소드의 내부를 구현하라. (10점)

```
public List<MeteorologicalData> GetListByTemperatureRange_Lambda(double min, double max)
{
    List<MeteorologicalData> aList = mList.FindAll(
        (m) => { return (m.Temperature >= min && m.Temperature <= max); });

    return aList;
}
```

2) GetListByTemperatureRange\_AnonymousMethod 대신에 사용할 수 있는 GetListByTemperatureRange\_LINQ 메소드의 내부를 구현하라. (10점)

```
public List<MeteorologicalData> GetListByTemperatureRange_LINQ(double min, double max)
{
    List<MeteorologicalData> aList = (from m in mList
                                     where (m.Temperature >= min && m.Temperature <= max)
                                     orderby m select m).ToList();

    return aList;
}
```

3) MainForm의 Log 이벤트와 Logger 이벤트 핸들러의 동작 원리를 설명하기 위하여, 위의 코드에서 어느 부분에 해당하는지 찾아서 적어라. (10점)

(1) 이벤트 핸들러 대리자 선언:

```
public delegate void LogHandler(string status);
```

(2) 이벤트 선언:

```
public event LogHandler Log;
```

(3) 이벤트를 호출하는 On이벤트 메소드 정의:

```
protected void OnLog(string status)
{
    if (Log != null) {
        Log(status);
    }
}
```

(4) 이벤트 핸들러 메소드 정의:

```
private void Logger(string info)
{
    System.Diagnostics.Trace.WriteLine(info);
}
```

(5) 이벤트 핸들러를 이벤트에 등록:

```
Log += new LogHandler(Logger);
```

(6) 이벤트 발생:

```
OnLog(message);
```

4) MainForm의 toolStripButton4\_Click 메소드가 실행된 결과를 설명하라. (10점)

(1) SearchForm의 Text, label1, label2를 각각 “날씨정보 찾기”과 “기온구간 between ”과 “ and ”로 지정하고, 모달형으로 나타나게 한다.

(2) SearchForm가 DialogResult.OK를 넘겨주면, SearchForm의 textBox1, textBox2 에 입력 값을 double 값인 min, max로 변환한다.

(3) GetListByTemperatureRange\_AnonymousMethod(min, max)를 사용하여, mList에 해당 구간(min ~ max) 내에 있는 모든 것을 찾아서 새롭게 mList를 업데이트한다.

(4) MainForm의 listView1에 새로운 mList 정보로 업데이트한다.

(5) 메인폼에서 Log 이벤트를 발생시켜서, Logger 이벤트 핸들러를 호출하여, 출력창(Output Window)에 “날씨정보 찾기 기온구간 between 20 and 30” 같이 출력해준다.

### 3. 다음 응용프로그램을 보고, 아래의 물음에 답하라. (35점)

```
partial class AridityIndexForm
{
    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.textBox1 = new System.Windows.Forms.TextBox();
        this.textBox2 = new System.Windows.Forms.TextBox();
        this.textBox3 = new System.Windows.Forms.TextBox();
        this.button1 = new System.Windows.Forms.Button();
        this.SuspendLayout();

        this.label1.Location = new System.Drawing.Point(30, 35);
        this.label1.Text = "연평균온도(C)";

        this.label2.Location = new System.Drawing.Point(30, 82);
        this.label2.Text = "연강수량(mm)";

        this.label3.Location = new System.Drawing.Point(30, 132);
        this.label3.Text = "건조지수(Ia)";

        this.textBox1.Location = new System.Drawing.Point(152, 32);
        this.textBox1.Size = new System.Drawing.Size(151, 21);
        this.textBox1.TextChanged += new System.EventHandler(this.textBox1_TextChanged);
        this.textBox1.KeyPress += new System.Windows.Forms.KeyPressEventHandler(this.textBox_KeyPress);

        this.textBox2.Location = new System.Drawing.Point(159, 79);
        this.textBox2.Size = new System.Drawing.Size(151, 21);
        this.textBox2.TextChanged += new System.EventHandler(this.textBox2_TextChanged);
        this.textBox3.KeyPress += new System.Windows.Forms.KeyPressEventHandler(this.textBox_KeyPress);

        this.textBox3.Location = new System.Drawing.Point(159, 129);
        this.textBox3.Size = new System.Drawing.Size(151, 21);

        this.button1.Location = new System.Drawing.Point(105, 176);
        this.button1.Size = new System.Drawing.Size(151, 21);
        this.button1.Text = "확인";
        this.button1.Click += new System.EventHandler(this.button1_Click); // button1 클릭 이벤트 핸들러

        this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 12F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(348, 237);
        this.Controls.Add(this.button1);
        this.Controls.Add(this.textBox3);
        this.Controls.Add(this.textBox2);
        this.Controls.Add(this.textBox1);
        this.Controls.Add(this.label3);
        this.Controls.Add(this.label2);
        this.Controls.Add(this.label1);
        this.Name = "AridityIndexForm";
        this.Text = "건조지수";
        this.ResumeLayout(false);
        this.PerformLayout();
    }
}
```

```

    }

    public partial class AridityIndexForm : Form
    {
        double T, P;
        private enum AridityIndexLevel { Perhumid, Humid, SubHumid, SemiArid, Arid, ExtremelyArid }
        private AridityIndexLevel AridityIndex(double T, double P)
        {

            // 건조지수 I = P/(T+10) [연강수량(P)과 연평균기온(T)]
            // ~60: Perhumid, 60~30: Humid, 30~20: SubHumid, 20~15: SemiArid, 15~5: Arid, 5~0:
            ExtremelyArid

        }

        public Form1()
        {
            InitializeComponent();

        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {
            // textBox1 값에서 T를 받고, AridityIndex 메소드를 이용하여 계산 후 textBox3에 출력
        }

        private void textBox2_TextChanged(object sender, EventArgs e)
        {
            // textBox2 값에서 P를 받고, AridityIndex 메소드를 이용하여 계산 후 textBox3에 출력
        }

        private void textBox_KeyPress(object sender, KeyPressEventArgs e)
        {
            if (!System.Text.RegularExpressions.Regex.IsMatch(e.KeyChar.ToString(), @"^[0-9]*(?:\.[0-9]*)?$"))
                e.Handled = true;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }

```

1) InitalizeComponent()에서 생성한 컨트롤을 그림으로 그려서 나타내라. (5점)



2) button1을 클릭(Click) 이벤트가 발생하면 폼을 닫는 이벤트 핸들러를 작성하라. 위의 코드에서 밑줄 친 빈 칸을 채우시오. (5점)

3) TextBox 컨트롤의 KeyPress 이벤트 발생시 그 실행결과를 서술하라. (5점)

- textBox1 과 textBox2 컨트롤에 키보드 키 입력 이벤트가 발생할 시, textBox.KeyPress이라는 이벤트 핸들러가 호출  
- 입력된 키 캐릭터(char)값이 0~9 숫자와 소수점(.)인지 확인해보고 (즉, double 숫자인지 보고) 아니면 입력하지 못하게 한다.

4) textBox1\_TextChanged 메소드의 내부 코드를 작성하라. (10점)

```
// textBox1 값에서 T를 받고, AridityIndex 메소드를 이용하여 계산 후 textBox3에 출력
private void textBox1_TextChanged(object sender, EventArgs e)
{
    bool valid = double.TryParse(textBox1.Text, out T);
    if (valid) {
        textBox3.Text = AridityIndex(T, P).ToString();
    }
}
```

5) 건조지수(I)를 계산하는 AridityIndexLevel AridityIndex(double T, double P) 메소드의 내부 코드를 작성하라. (10점)

```
private AridityIndexLevel AridityIndex(double T, double P)
{
    double I = P / (T + 10);
    if (I > 60.0) return AridityIndexLevel.Perhumid;
    else if (I <= 60.0 && I > 30.0) return AridityIndexLevel.Humid;
    else if (I <= 30.0 && I > 20.0) return AridityIndexLevel.SubHumid;
    else if (I <= 20.0 && I > 15.0) return AridityIndexLevel.SemiArid;
    else if (I <= 15.0 && I > 5.0) return AridityIndexLevel.Arid;
    else return AridityIndexLevel.ExtremelyArid;
}
```

-끝-