|  |
| --- |
| 윈도우즈 프로그래밍 |

-과제-

12주차

윈도우 폼 : 키보드, 마우스, 고급 컨트롤 수업예제

20144638

이한범

# 예제 1번

메뉴 항목 이벤트 및 상황 메뉴 설계를 작성.

메뉴에는 파일, 편집, 도움말이 있으며 몇몇 메뉴는 단축키를 지정한다. 파일메뉴 항목들을 클릭하게 되면 리스트 박스에 메뉴 정보가 추가된다. 오른쪽 클릭을 누르게 될 경우 팝업창을 띄운다.

1. 윈도우 폼 구성도

|  |
| --- |
|  |

1. 작성한 코드내용

# 각종 메뉴 클릭 시 발생되는 이벤트

|  |
| --- |
| namespace week  {  public partial class Form1 : Form  {  public Form1()  {  InitializeComponent();  }  private void newFile\_Click(object sender, EventArgs e)  {  listBox1.Items.Add(newFile.Text);  }  private void openFile\_Click(object sender, EventArgs e)  {  listBox1.Items.Add(openFile.Text);  }  private void closeFile\_Click(object sender, EventArgs e)  {  listBox1.Items.Add(closeFile.Text);  }  private void saveFile\_Click(object sender, EventArgs e)  {  listBox1.Items.Add(saveFile.Text);  }  private void saveFile\_another\_Click(object sender, EventArgs e)  {  listBox1.Items.Add(saveFile\_another.Text);  }  private void print\_Click(object sender, EventArgs e)  {  listBox1.Items.Add(print.Text);  }  private void preview\_Click(object sender, EventArgs e)  {  listBox1.Items.Add(preview.Text);  }  private void exit\_Click(object sender, EventArgs e)  {  listBox1.Items.Add(exit.Text);  }  private void clear\_b\_Click(object sender, EventArgs e)  {  listBox1.Items.Clear();  }  private void exit\_b\_Click(object sender, EventArgs e)  {  Application.Exit();  }  }  } |

▲ 각각의 메뉴 요소들마다 클릭시에 listBox1에 해당 메뉴 이름의 값을 가지는 리스트들을 추가시켜준다.

▲ 만일 clear버튼을 누르게 될 경우, listBox1 내용을 초기화하고, exit 버튼을 누르게 될 경우 해당 폼을 종료한다.

3) 실행결과

|  |
| --- |
|  |

# 예제 2번

마우스 이동 이벤트에 대한 예제 작성.

마우스가 폼 내부로 들어갈 시, MouseEnter 이벤트가 발생. 이 때, 어느 좌표인지 메시지 박스를 통해 출력한다.

1. 윈도우 폼 구성도

|  |
| --- |
|  |

1. 작성한 코드내용

# 마우스 클릭 이벤트

|  |
| --- |
| public partial class Form1 : Form  {  public Form1()  {  InitializeComponent();  }  private void Form1\_MouseEnter(object sender, EventArgs e)  {  Point mousePoint = PointToClient(MousePosition);  string msg = "Mouse Position : " +mousePoint.X + ", "+mousePoint.Y;  MessageBox.Show(msg);  }  } |

▲ 마우스가 폼 위로 들어가게 될 경우 마우스의 x,y좌표를 메시지박스를 통해 띄우게 된다.

3) 실행결과

|  |
| --- |
|  |

# 예제 3번

입력 포커스를 적용하는 예제를 작성.

버튼을 눌렀을 때, 입력 포커스를 지정해주는 예시를 작성한다. 버튼에 클릭이벤트가 발생하면 다른 버튼의 색깔을 어둡게 바꾼다.

1. 윈도우 폼 구성도

|  |
| --- |
|  |

1. 작성한 코드내용

# 입력 포커스를 보이기 위한 버튼이벤트

|  |
| --- |
| public partial class Form1 : Form  {  public Form1()  {  InitializeComponent();  }  private void button1\_Click(object sender, EventArgs e)  {  this.button1.BackColor = SystemColors.Control;  this.button2.Focus();  if (this.button2.Focused)  this.button2.BackColor = SystemColors.ControlDark;  }  private void button2\_Click(object sender, EventArgs e)  {  this.button2.BackColor = SystemColors.Control;  this.button1.Focus();  if (this.button1.Focused)  this.button1.BackColor = SystemColors.ControlDark;  }  } |

▲ 만일 button1이 클릭될 경우, button2를 포커싱하게 되고, button2가 포커싱 된 상태라면 button2의 배경색을 어두운 색으로 바꾼다. 반대도 동일.

3) 실행결과

|  |
| --- |
|  |

# 예제 4번

키보드 이벤트를 처리하는 예시 작성.

사용자로부터 키보드가 눌리는 KeyUp 이벤트가 발생할 경우, 폼 위의 버튼이 눌려진 키보드의 값으로 바뀌는 코드를 작성한다. 단, Up, Down, Left, Right 가 눌렸을 경우, 각 방향으로 버튼이 이동한다.

1. 윈도우 폼 구성도

|  |
| --- |
|  |

1. 작성한 코드내용

# 각종 메뉴 클릭 시 발생되는 이벤트

|  |
| --- |
| public partial class Form1 : Form  {  public int xPt, yPt;  public static readonly int MOVE = 10;  public Form1()  {  InitializeComponent();  }  private void button1\_KeyUp(object sender, KeyEventArgs e)  {  this.xPt = this.button1.Location.X;  this.yPt = this.button1.Location.Y;  switch (e.KeyCode)  {  case Keys.Left:  xPt -= MOVE; break;  case Keys.Right:  xPt += MOVE; break;  case Keys.Up:  yPt -= MOVE; break;  case Keys.Down:  yPt += MOVE; break;  }  this.button1.Text = e.KeyCode.ToString();  this.button1.Location = new Point(xPt, yPt);  }  } |

▲ button1의 keyUP 이벤트는 사용자가 키보드를 입력하게 될경우 발생되는데, button1의 텍스트는 사용자가 입력한 키 값이 되며, 만일 화살표키가 입력될 경우, 각 방향으로 이동하게 된다.

3) 실행결과

|  |
| --- |
|  |