

## 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. HW3에서 사용했던 X, Y, Width, Height 를 멤버로 갖는 Bound 클래스를 작성하라. 기본 생성자, 인자를 받는 생성자, ToString()를 갖는다. (10점)

```
public class Bound
{
    public int X
    {
        get;
        set;
    }
    public int Y
    {
        get;
        set;
    }
    public int Width
    {
        get;
        set;
    }
    public int Height
    {
        get;
        set;
    }

    public Bound() : this (0, 0, 10, 10) {}

    public Bound(int x, int y, int width, int height)
    {
        this.X = x;
        this.Y = y;
        this.Width = width;
        this.Height = height;
    }

    public override ToString()
    {
        return string.Format("{0},{1},{2},{3}", X, Y, Width, Height);
    }
}
```

## 2. Bound 클래스에 IComparable<Bound>를 구현하라. (10점)

```
public class Bound : IComparable<Bound>
{
// 중간 생략

    public int CompareTo(Bound other)
    {
        return X.CompareTo(other.X);
    }
}
```

## 3. 두 개의 Bound의 Width로 비교하는 BoundWidthComparer 클래스를 구현하라. (10점)

```
public class BoundWidthComparer : IComparer<Bound>
{
    public int Compare(Bound b1, Bound b2)
    {
        return b1.Width.CompareTo(b2.Width);
    }
}
```

## 4. 다음 코드를 보고 (1),(2),(3),(4) 동작 원리를 설명하라. (10점)

```
private List<Bound> boundQuery;
public Form4(List<Quadrilateral> quadList)
{
    // quadList에서 각 Quadrilateral의 Bound만 query해서 List로 만들어줌
    boundQuery = (from q in quadList select q.Bounds).ToList();
}
private void SortList(int column)
{
    if (column == 0) boundQuery.Sort(); // (1)
    else if (column == 1) boundQuery.Sort((p, q) => p.Y.CompareTo(q.Y)); // (2)
    else if (column == 2) boundQuery.Sort(new BoundWidthComparer()); // (3)
    else if (column == 3) boundQuery.Sort(delegate(Bound b1, Bound b2) // (4)
        {
            return b1.Height.CompareTo(b2.Height);
        });
}
```

(1) Bound 클래스의 IComparable<Bound>가 호출되어 X로 Sort

(2) Lambda 식을 사용하여 boundQuery에 두 개 Bound의 Y로 Sort

(3) BoundWidthComparer 클래스의 IComparer<Bound>가 호출되어 Width로 Sort

(4) 무명 대리자를 사용하여 두 개 Bound의 Height로 Sort

5. 다음 코드의 (1),(2),(3),(4),(5) 동작 원리를 설명하라. (10점)

```
private Bound Method1(List<Point> vertices)
{
    var xQuery = from Point p in vertices select p.X; // (1)
    int xmin = xQuery.Min(); // (2)
    int xmax = xQuery.Max(); // (2)
    var yQuery = from Point p in vertices select p.Y; // (3)
    int ymin = yQuery.Min(); // (4)
    int ymax = yQuery.Max(); // (4)
    return new Bound(xmin, ymin, xmax - xmin, ymax - ymin); // (5)
}
```

- (1) vertices 에서 각 점 p의 X만 선택한 xQuery 리스트
- (2) xQuery에서 X의 Min값과 Max 값을 찾음
- (3) vertices 에서 각 점 p의 Y만 선택한 yQuery 리스트
- (4) yQuery에서 Y의 Min값과 Max 값을 찾음
- (5) vertices 에서 X,Y의 Min,Max를 가지고 Bound 클래스 객체를 생성해서 반환

-끝-