

## 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 암호(4자리 숫자)를 기입하면 성적공고시 학번대신 암호를 사용할 것임.

1. 다음은 Person, Student 클래스 예제 프로그램이다. 다음 질문에 답하시오. (60점)

```

class Person : IComparable<Person> { // (1)
    public string Name { get; set; }
    public int Age { get; set; }
    protected Person() : this("", 0) {}
    public Person(string name, int age) { // (2)
        this.Name = name; this.Age = age;
    }
    public Person(Person other) { // (2)
        this.Name = other.Name; this.Age = other.Age;
    }
    public virtual void Set(Person other) { // (3)
        this.Name = other.Name; this.Age = other.Age;
    }
    public override string ToString() { return string.Format("Person Name: {0} Age: {1}", Name, Age); }
    public int CompareTo(Person other) { // (6)
        return this.Name.CompareTo(other.Name);
    }
}

class Student : Person { // (1)
    public int ID { get; set; }
    public Student() { ID = 5208; }
    public Student(string name, int age, int id) : base(name, age) { // (2)
        this.ID = id;
    }
    public Student(Student other) : base(other.Name, other.Age) { // (2)
        this.ID = other.ID;
    }
    public void Set(Student other) {
        this.Name = other.Name; this.Age = other.Age; this.ID = other.ID;
    }
    public override void Set(Person other) { // (3)
        if (other is Person) base.Set(other);
        else this.Set((Student)other);
    }
    public override string ToString() {
        return string.Format("Student Name: {0} Age: {1} ID: {2}", Name, Age, id);
    }
    public int CompareTo(Student other) {
        return this.Name.CompareTo(other.Name);
    }
}
    
```

```

    }
}
class Program {
    static void Print(object o) {
        Console.WriteLine(o);
    }
    static void Print(object[] arr) {
        Array.ForEach(arr, e => Console.WriteLine(e));
    }
    static void Main(string[] args) {
        PersonStudentTest1();
        PersonStudentTest2();
        PersonStudentTest3();
    }
    static void PersonStudentTest1() { // (3)
        Person[] pList = new Person[5];
        pList[0] = new Person("HCI0", 0);
        pList[1] = new Person("HCI1", 1);
        pList[2] = pList[0];
        pList[2].Age = 10;
        pList[3] = new Student("HCI2", 2, 222);
        pList[4] = new Student("HCI3", 3, 333);
        Print(pList); // 1-5
        pList[0].Set(new Student("HCI4", 4, 444)); // Person.Set(Student)
        pList[4].Set(new Person("HCI5", 5)); // Student.Set(Person)
        Print(pList); // 6-10
        pList[0].Set(pList[4]); // Person.Set(Student)
        pList[1].Set(new Person("HCI6", 6)); // Person.Set(Person)
        pList[2] = new Person("HCI7", 7);
        ((Student)pList[3]).Set(new Student("HCI8", 8, 888)); // Student.Set(Student)
        ((Student)pList[4]).Set(new Person("HCI9", 9)); // Student.Set(Person)
        Print(pList); // 11-15
    }
    static void PersonStudentTest2() {
        Person h1 = new Person();
        Print(h1); // 출력결과 Person Name: Age: 2016
        Student h2 = new Student("HCI");
        Print(h2); // 출력결과 Student Name: Age: 0 ID: 5208
    }
    static void PersonStudentTest3() {
        List<Person> pList = new List<Person>();
        pList.Add(new Person ("HCI4", 2));
        pList.Add(new Student ("HCI1", 3, 111));
        pList.Add(new Student ("HCI2", 4, 555));
        pList.Sort(); // (6)
        pList.ForEach(e => Console.WriteLine(e)); // (6)
        pList.Sort((p1, p2) => p1.Age.CompareTo(p2.Age)); // (6)
        pList.ForEach(e => Console.WriteLine(e)); // (6)
    }
}
}

```

- 1) Person 클래스는 IComparable 인터페이스를 상속받고, Student 클래스는 Person 클래스를 상속받았다. C#에서 인터페이스와 클래스 상속 관계의 차이점을 자세히 설명하라. (10점)

```
class Person: IComparable<Person> { // IComparable 인터페이스 상속 - 여러 개의 인터페이스 상속 가능. 단, 상속받은 인터페이스는 인터페이스의 추상메소드를 모두 구현해야 함
} 인터페이스는 다중 상속 가능.
class Student: Person { // Person 부모클래스를 상속받아서 부모클래스의 protected & public 를 그대로 사용하며 추가로 확장해서 사용가능 함 객체간의 계층적 관계를 맺음으로써 코드의 재사용 및 간결성이 가능함.
} 클래스는 단일 상속만 가능.
```

- 2) Person과 Student 클래스의 생성자를 구현하라. (10점)

```
public Person(string name, int age) { // (2)
    this.Name = name; this.Age = age;
}
public Person(Person other) { // (2)
    this.Name = other.Name; this.Age = other.Age;
}
public Student(string name, int age, int id) : base(name, age) { // (2)
    this.ID = ID;
}
public Student(Student other) : base(other.Name, other.Age) { // (2)
    this.ID = other.ID;
}
```

- 3) Person과 Student 클래스에서 밑줄 친 부분인 void Set(Person)에 대해 동작원리를 자세히 설명하라. (10점)

Method overriding은 상속받은 파생클래스에서 동일한 함수명에 동일한 매개변수 정의하여 함수를 재정의하는 것으로 상속되어진 함수의 기능을 변경해서 재사용하고 싶은 때 사용하는 것.

Person과 Student 클래스에 void Set(Person)는 동일한 함수를 재정의해서 사용한 method overriding Person 클래스의 void Set(Person)은 가상함수로 만들어 overriding이 가능하도록 했으며, Student 클래스의 void Set(Person)는 override해서 해당 객체가 Person이면 base(즉 Person)의 Set을 호출하여 지정했고, 해당 객체가 Student이면 자신의 Set을 호출하여 지정했다.

- 4) Program의 PersonStudentTest1() 실행결과를 자세히 나타내라. 실행결과에 호출 번호도 표시할 것. (15점)

```
Person Name: HCI0 Age: 10 // (1-5)
Person Name: HCI1 Age: 1
Person Name: HCI0 Age: 10
Student Name: HCI2 Age: 2 Id: 222
Student Name: HCI3 Age: 3 Id: 333
Person Name: HCI4 Age: 4 // (6-10)
Person Name: HCI1 Age: 1
Person Name: HCI4 Age: 4
Student Name: HCI2 Age: 2 Id: 222
Student Name: HCI5 Age: 5 Id: 333
Person Name: HCI4 Age: 4 // (11-15)
Person Name: HCI4 Age: 4
Person Name: HCI4 Age: 4
Student Name: HCI6 Age: 6 Id: 111
Student Name: HCI6 Age: 6 Id: 111
```

- 5) PersonStudentTest2()에서 밑줄 친 부분에 에러가 발생한다 그 이유를 설명하고, 출력결과가 같도록 에러를 수정하라. (10점)

Person() 기본 생성자는 protected로 되어 있어서 외부 클래스에서 access가 불가하므로 에러 발생  
 Person h1 = new Person("", 2016);로 수정 해야 함.  
 또는 Person() { this.Name = ""; this.Age = 2016; }  
 Student(string name) 생성자가 없기 때문에 에러가 발생  
 Student h2 = new Student();  
 또는 Student h2 = new Student ("", 0, 5208);

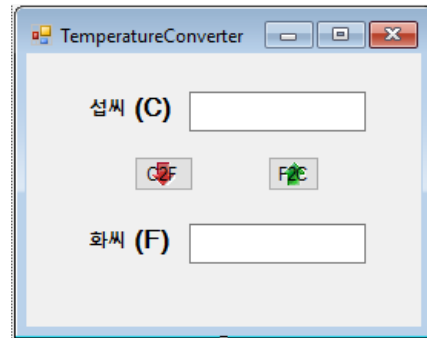
- 6) PersonStudentTest3()에서 출력 결과를 적고, Sort 방법에 대한 차이점을 설명하라. (10점)

출력결과:  
 Student Name: HCI1 Age: 30 ID: 333  
 Student Name: HCI2 Age: 10 ID: 555  
 Person Name: HCI4 Age: 20  
 Student Name: HCI2 Age: 10 ID: 555  
 Person Name: HCI4 Age: 20  
 Student Name: HCI1 Age: 30 ID: 333

pList.Sort();는 기본 Sort 동작, Person과 Student 클래스의 CompareTo가 호출되는 방식  
 pList.Sort((p1, p2) => p1.Age.CompareTo(p2.Age));는 람다식을 사용하여 Person 간에 Age를 비교한 방식

2. 다음은 TemperatureConvertForm 예제 프로그램이다. 다음 질문에 답하시오. (40점)

```
public partial class Form1 : Form {
    private double FromFahrenheitToCelsius(double F) {
        return (F - 32.0) * (5.0 / 9.0);
    }
    private double FromCelsiusToFahrenheit(double C) {
        return ((9.0 / 5.0) * C + 32.0);
    }
    private double ToDouble(string str) { // (1)
        double value;
        bool result = double.TryParse(str, out value);
        return value;
    }
    public Form1() {
        InitializeComponent(); // (2)
    }
    private void button1_Click(object sender, EventArgs e) { // (3)
        double tempC = ToDouble(textBox1.Text); // (1)
        double tempF = FromCelsiusToFahrenheit(tempC);
    }
}
```



```

        textBox2.Text = tempF.ToString();
    }
    private void button2_Click(object sender, EventArgs e) { // (3)
        double tempF = ToDouble(textBox2.Text); // (1)
        double tempC = FromFahrenheitToCelsius(tempF);
        textBox1.Text = tempC.ToString();
    }
    private void textBox1_KeyDown(object sender, KeyEventArgs e) { // (4)
        if (e.KeyCode == Keys.Enter) {
            button1_Click(sender, e);
        }
    }
    private void textBox2_KeyDown(object sender, KeyEventArgs e) { // (4)
        if (e.KeyCode == Keys.Enter) {
            button2_Click(sender, e);
        }
    }
}
partial class Form1 {
    private void InitializeComponent() { // (1)
        this.textBox1 = new System.Windows.Forms.TextBox();
        this.textBox2 = new System.Windows.Forms.TextBox();
        this.label1 = new System.Windows.Forms.Label();
        this.label2 = new System.Windows.Forms.Label();
        this.button1 = new System.Windows.Forms.Button();
        this.button2 = new System.Windows.Forms.Button();
        this.imageList1 = new System.Windows.Forms.ImageList(this.components);
        this.SuspendLayout();

        // 중간 생략..
        this.button1.Click += new System.EventHandler(this.button1_Click); // (3)
        this.button2.Click += new System.EventHandler(this.button2_Click); // (3)
        this.textBox1.KeyDown += new System.Windows.Forms.KeyEventHandler(this.textBox1_KeyDown); //
(4)
        this.textBox2.KeyDown += new System.Windows.Forms.KeyEventHandler(this.textBox2_KeyDown); //
(4)
        // 중간 생략..
    }
}

```

2.1 Form1 클래스의 `double ToDouble(string str)` 메소드는 무엇인지 기능을 자세히 설명하라. 이 메소드가 프로그램에서 어떻게 사용되었는지 자세히 설명하라. (10점)

`double.TryParse`를 이용해서 `string` 객체를 `double` 값으로 변환해주는 메소드이다.  
만약 `double`로 변환이 되지 않는 경우라면 0을 넘겨주는 메소드이다.  
`textBox1` & `textBox2`에서 입력받은 `Text string` 값을 `double`로 변환하는데 사용했다.

2.2 Form1 클래스의 `InitializeComponent()` 메소드는 무엇인지 기능을 자세히 설명하라. (10점)

Form1 클래스의 `InitializeComponent()` 함수는 그림에서 보이듯이 Form에 넣은 각종 레이블, 버튼, 텍스트박스, 이미지 리스트 등 컨트롤 클래스 객체를 생성하고 배치한 결과를 코드로 작성된 메소드이다. 즉, 각종 컨트롤 객체의 속성(예: `Location`, `Size`, `Name`, `Text` 등등)을 지정하고 컨트롤 객체의 이벤트(예: 버튼의 클릭 이벤트)에 따른 이벤트 핸들러(예: `button1_Click`)를 등록한다.

2.3 이벤트와 이벤트 핸들러가 무엇인지 자세히 설명하라. Form1 클래스 `InitializeComponent()` 메소드 내에 `this.button1.Click += new System.EventHandler(this.button1_Click);`는 무엇인가? (10점)

이벤트는 발생한 사건을 알리기 위해 보내는 메시지  
이벤트 핸들러는 이벤트에 응답하는 메소드를 가리키는 대리자(delegate)  
버튼1 또는 버튼2이 클릭되었을 때 `Click` 이벤트가 발생하여 그에 따른 이벤트 핸들러인 `button1_Click`과 `button2_Click`이 호출

2.4 Form1 클래스 `private void textBox1_KeyDown(object sender, KeyEventArgs e)`의 `KeyEventArgs`가 무엇인지, 해당 메소드의 내부 동작 원리를 설명하라. (10점)

`KeyEventArgs`는 이벤트 발생시 넘겨 줄 추가정보로써, 특정 키 코드 (`Key.Code`) 값이 무엇인지 코드에서는 엔터키가 눌렸는지 확인하여 맞으면 `button1_Click`을 호출함으로써 화씨에서 섭씨로 온도 변환을 수행한다.

-끝-