|  |
| --- |
| 윈도우즈 프로그래밍 |

-과제-

7주차

20144638

이한범

# 1번문제

학생의 정보를 저장하는 student 클래스 선언. 학번, 학생 이름, 생년월일, 성별을 필드로 가지는 student 클래스를 정의하여 생성 및 실행할 것.

1. 작성한 코드내용

# Program class (main)

|  |
| --- |
| namespace fifthWeek  {  class Program  {  static void Main(string[] args)  {  //1-4  student[] std = new student[4];  student temp = new student();  string info;  int n, num;  string name;  char zender;  int bir;  Console.Write("학생 수를 입력하시오 : ");  n = Int32.Parse(Console.ReadLine());  for(int i=0; i<n; i++)  {  Console.Write("학생의 정보를 입력하시오<학번 이름 성별 생년월일> : ");  info = Console.ReadLine();  std[i] = new student(Int32.Parse(info.Split()[0]), info.Split()[1], Char.Parse(info.Split()[2]), Int32.Parse(info.Split()[3]));  }  for (int i = 0; i < n; i++)  std[i].display();  Console.Write("수정할 학생의 학번을 입력하시오 : ");  num = Int32.Parse(Console.ReadLine());  Console.WriteLine("수정 전 정보 : ");  for (int i = 0; i < n; i++)  {  if (std[i].Stdnum == num)  {  std[i].display();  temp = std[i];  break;  }  }  Console.Write("학생의 이름을 입력하시오 : ");  name = Console.ReadLine();    Console.Write("학생의 성별을 입력하시오 : ");  zender = Char.Parse(Console.ReadLine());  Console.Write("학생의 생년 월일을 입력하시오 : ");  bir = Int32.Parse(Console.ReadLine());  temp.Zender = zender;  temp.Name = name;  temp.Birthday = bir;  Console.WriteLine("수정 후 정보 : ");  temp.display();  }  }  } |

▲ 메인 클래스. split() 메서드를 통해 공백으로 문자열을 잘라주어 필드들을 나누어 정의시켜줌.

수정시킬 학생의 정보를 탐색 후에 temp라는 student 객체에 저장시켜줌으로 써 for문 밖에서도 참조할 수 있도록 하였음.

# student class

|  |
| --- |
| namespace fifthWeek  {  class student  {  //1-1  private int stdnum;  private int birthday;  private string name;  private char zender;  //1-2  public int Stdnum{  get { return stdnum; }  set { stdnum = value; }  }  public int Birthday{  get { return birthday; }  set { birthday = value; }  }  public string Name{  get { return name; }  set { name = value; }  }  public char Zender{  get { return zender; }  set { zender = value; }  }  public student()  {  this.stdnum = 0;  this.birthday = 0;  this.name = "";  this.zender = ' ';  }  public student(int stdnum, string name, char zender, int birthday)  {  this.stdnum = stdnum;  this.birthday = birthday;  this.name = name;  this.zender = zender;  }  //1-3  public void display()  {  Console.WriteLine(this.stdnum + " " + this.name + " " + this.zender + " " + this.birthday);  }  }  } |

▲student 클래스 정의. 프로퍼티를 사용하여 각종 필드들의 접근자와 설정자를 정의해 주었으며, 학생의 정보를 출력시켜주는 display 메서드를 정의해 주었음.

2) 실행결과

|  |
| --- |
|  |

# 2번문제

Shape 클래스를 상속받아 Triangle 클래스와 Rectangle 클래스를 만들어라.

Triangle클래스는 Draw()함수를 재정의하여 크기에 맞춰 삼각형 모양을 출력하라.

Rectangle클래스는 Draw()함수를 재정의하여 가로, 세로에 맞춰 사각형 모양을 출력하라.작성한 코드내용

# second class (main)

|  |
| --- |
| namespace fifthWeek  {  class second  {  public static void Main(string[] argv) {  int length;  string size;  Console.Write("삼각형의 크기를 입력하시오 : ");  length = Int32.Parse(Console.ReadLine());  Console.Write("사각형의 가로, 세로 크기를 입력하시오 : ");  size = Console.ReadLine();  Triangle t = new Triangle(length);  Rectangle r = new Rectangle(Int32.Parse(size.Split(',')[0]), Int32.Parse(size.Split(',')[1]));  Console.WriteLine("삼각형");  t.Draw();  Console.WriteLine("사각형");  r.Draw();  }  }  } |

▲ 메인 클래스. 각각의 크기 값들에 대한 변수를 선언해 주었고, 삼각형 객체인 t와 사각형 객체인 r의 Draw() 메서드를 호출

# Shape class

|  |
| --- |
| namespace fifthWeek  {  //추상 클래스 정의  abstract class Shape  {  protected int Size { get; set; }  public Shape(int size){  Size = size;  }  //추상 메소드 Draw() 정의.  //추상메소드가 하나라도 있으면 추상클래스.  public abstract void Draw();  }  } |

▲추상 클래스 Shape 정의. 주석에 써있는 것과 같이 Draw()라는 추상메서드가 있으므로 Shape를 추상클래스로 만들어주어야한다.

# Rectangle class

|  |
| --- |
| namespace fifthWeek  {  //사각형  class Rectangle : Shape  {  private int horizontal;  private int vertical;  public Rectangle(int vertical, int horizontal) : base(horizontal)  {  this.horizontal = horizontal;  this.vertical = vertical;  }  public override void Draw()  {  for (int i = 0; i < horizontal; i++)  {  for (int star = 0; star < vertical; star++)  Console.Write('\*');  Console.WriteLine('\n');  }  }  }  } |

▲Shape를 상속받는 사각형 클래스인 Rectangle 클래스 정의. 가로, 세로값을 필드로 가지며 생성자를 통해 값을 설정해 줄 수 있다. Draw()는 재정의함으로 써 가로, 세로 길이에 따른 도형을 \*을 찍음으로써 콘솔창에 보여준다.

# Triangle class

|  |
| --- |
| namespace fifthWeek  {  //삼각형  class Triangle : Shape  {  //크기 필드 정의  private int length;  public Triangle(int length) : base(length) {  this.length = length;  }  //Draw() override  public override void Draw()  {  for (int i =0; i < length; i++)  {  for (int x=0; x<length-1-i;x++)  Console.Write(' ');  for (int star = 0; star<(2\*i)+1; star++)  Console.Write('\*');  Console.WriteLine('\n');  }  }  }  } |

▲ Shape를 상속받는 삼각형 클래스인 Triangle 클래스 정의. 높이값을 필드로 가지며 생성자를 통해 값을 설정해 줄 수 있다. Draw()는 재정의함으로 써 length에 따른 도형을 \*을 찍음으로써 콘솔창에 보여준다.

2) 실행결과

|  |
| --- |
|  |

# 3번문제

Shape클래스를 상속받아 Rectangle, Circle, Triangle 클래스를 만들어라.

1) 작성한 코드내용

# Program class (main)

|  |
| --- |
| namespace fifthWeek  {  class Program  {  public static void Main(string[] argv) {  Rectangel r = new Rectangel();  Rectangel r2 = new Rectangel(3, 5);  Circle c = new Circle();  Circle c2 = new Circle(5);  Triangle t = new Triangle();  Triangle t2 = new Triangle(3, 4, 5);  Console.WriteLine("사각형 넓이 : {0}, 둘레 : {1}", r.area(), r.perimeter());  Console.WriteLine("사각형 넓이 : {0}, 둘레 : {1}", r2.area(), r2.perimeter());  Console.WriteLine("원 넓이 : {0}, 둘레 : {1}", c.area(), c.perimeter());  Console.WriteLine("원 넓이 : {0}, 둘레 : {1}", c2.area(), c2.perimeter());  Console.WriteLine("삼각형 넓이 : {0}, 둘레 : {1}", t.area(), t.perimeter());  Console.WriteLine("삼각형 넓이 : {0}, 둘레 : {1}", t2.area(), t2.perimeter());  }  }  } |

▲ 메인 클래스. 각 도형들에 대한 객체를 선언. 하나는 인수 없이, 하나는 인수를 넣고서 객체를 만들고 각기 area(), perimeter() 메서드를 호출하여 값을 콘솔창에 출력.

# Shape class

|  |
| --- |
| namespace fifthWeek  {  //도형을 그리고자 할 때 상속받는 추상클래스.  //area, perimeter는 상속시 재정의하여야 함.  abstract class Shape  {  protected double width, length, radius, a, b, c;  public abstract double area();  public abstract double perimeter();  }  } |

▲area(), perimeter()라는 추상메서드를 가지는 추상클래스 Shape 클래스 정의. 각 도형들은 이 클래스를 상속받아 area와 perimeter를 구현한다.

# Rectangle class

|  |
| --- |
| namespace fifthWeek  {  class Rectangel : Shape  {  public Rectangel()  {  this.width = 1;  this.length = 1;  }  //사각형 생성자  public Rectangel(double w, double l)  {  this.width = w;  this.length = l;  }  //area, perimeter 재정의  public override double area()  {  return this.width \* length;  }  public override double perimeter()  {  return (this.width + this.length) \* 2;  }  }  } |

▲Shape를 상속받아 area(), perimeter()를 구현하는 사각형 클래스 정의.

# Triangle class

|  |
| --- |
| namespace fifthWeek  {  class Triangle : Shape  {  private double s;  public Triangle()  {  this.a = 1;  this.b = 1;  this.c = 1;  }  public Triangle(double a, double b, double c)  {  this.a = a;  this.b = b;  this.c = c;  }  //area, perimeter 재정의  public override double area()  {  s = (this.a + this.b + this.c) / 2;  return Math.Sqrt(s \* (s - a) \* (s - b) \* (s - c));  }  public override double perimeter()  {  return a + b + c;  }  }  } |

▲ Shape를 상속받아 area(), perimeter()를 구현하는 원 클래스 정의.

# Circle class

|  |
| --- |
| namespace fifthWeek  {  class Circle : Shape  {  private const double PI = 3.141592;  public Circle()  {  this.radius = 1;  }  public Circle(double r)  {  this.radius = r;  }  //area, perimeter 재정의  public override double area()  {  return Math.Pow(this.radius, 2) \* PI;  }  public override double perimeter()  {  return this.radius \* 2 \* PI;  }  }  } |

▲ Shape를 상속받아 area(), perimeter()를 구현하는 삼각형 클래스 정의.

2) 실행결과

|  |
| --- |
|  |