Abstract Class & Interface

Calculate Area of some figures

Class squre

{

Int side, area;

Public GetDimensions()

{

Enter Value of Side;

}

Public void CalclateArea()

{

Area = side \* side;

}

Publiv void DisplayArea()

{

Area Is

}

}

Class rectangle

{

Int length , breadth, area;

Public GetDimensions()

{

Enter Value of Side;

}

Public void CalclateArea()

{

Area = length \* breadth;

}

Publiv void DisplayArea()

{

Area Is

}

}

using System;

class square

{

int side, area;

public void GetDimensions()

{

Console.WriteLine("Enter Value of Side");

side = Convert.ToByte(Console.ReadLine());

}

public void CalclateArea()

{

area = side \* side;

}

public void DisplayArea()

{

Console.WriteLine("Area Is " + area);

}

}

class rectangle

{

int length , breadth, area;

public void GetDimensions()

{

Console.WriteLine("Enter Value of Length");

length = Convert.ToByte(Console.ReadLine());

Console.WriteLine("Enter Value of Breadth");

breadth = Convert.ToByte(Console.ReadLine());

}

public void CalclateArea()

{

area = length \* breadth;

}

public void DisplayArea()

{

Console.WriteLine("Area Is " + area);

}

}

class triangle

{

int basee, height, area;

public void GetDimensions()

{

Console.WriteLine("Enter Value of Length");

basee = Convert.ToByte(Console.ReadLine());

Console.WriteLine("Enter Value of Breadth");

height = Convert.ToByte(Console.ReadLine());

}

public void CalclateArea()

{

area = (int).5 \* basee \* height;

}

public void DisplayArea()

{

Console.WriteLine("Area Is " + area);

}

}

Abstract Class : Its is class in which some of the methods are defined and some of the methods are not defined?

Why do we create it : Only to be inherited.

We cannot create its object.

All the abstract methods of this class have to be implanted in the child classes which inherit it.

using System;

abstract class figure

{

protected int area;

public abstract void GetDimensions();

public abstract void CalclateArea();

public void DisplayArea()

{

Console.WriteLine("Area Is " + area);

}

}

class square : figure

{

int side;

public override void GetDimensions()

{

Console.WriteLine("Enter Value of Side");

side = Convert.ToByte(Console.ReadLine());

}

public override void CalclateArea()

{

area = side \* side;

}

}

class rectangle : figure

{

int length , breadth;

public override void GetDimensions()

{

Console.WriteLine("Enter Value of Length");

length = Convert.ToByte(Console.ReadLine());

Console.WriteLine("Enter Value of Breadth");

breadth = Convert.ToByte(Console.ReadLine());

}

public override void CalclateArea()

{

area = length \* breadth;

}

}

class triangle : figure

{

int basee, height;

public override void GetDimensions()

{

Console.WriteLine("Enter Value of Length");

basee = Convert.ToByte(Console.ReadLine());

Console.WriteLine("Enter Value of Breadth");

height = Convert.ToByte(Console.ReadLine());

}

public override void CalclateArea()

{

area = (int).5 \* basee \* height;

}

}

class circle : figure

{

double radius;

public override void CalclateArea()

{

area = (int)3.14 \* (int)Math.Pow(radius, 2);

}

public override void GetDimensions()

{

Console.WriteLine("Enter Radius");

radius = Convert.ToDouble(Console.ReadLine());

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp4

{

class Program

{

static void Main(string[] args)

{

square s = new square();

s.GetDimensions();

s.CalclateArea();

s.DisplayArea();

rectangle r = new rectangle();

r.GetDimensions();

r.CalclateArea();

r.DisplayArea();

}

}

}

Monday : Exception Handling

Interfaces

Linq