|  |
| --- |
| Day 11 assignments by lokesh nadella |

1.write a sample program for interface discussed in class Ishape

1.circle

2.square

3.traingle

4.rectangle

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace day11\_morning\_assignment\_ishape

{

interface IShape

{

int CalculatePerimeter();

int CalculateArea();

}

class Circle : IShape

{

private int radius;

public void ReadRadius()

{

Console.WriteLine("Enter radius:");

radius = Convert.ToInt32(Console.ReadLine());

}

public int CalculateArea()

{

return 22 \* radius \* radius / 7;

}

public int CalculatePerimeter()

{

return 2 \* 22 \* radius / 7;

}

}

class Square : IShape

{

private int side;

public void ReadSide()

{

Console.WriteLine("Enter side:");

side = Convert.ToInt32(Console.ReadLine());

}

public int CalculateArea()

{

return side \* side;

}

public int CalculatePerimeter()

{

return 4 \* side;

}

}

class Rectangle : IShape

{

private int length;

public void ReadLength()

{

Console.WriteLine("Enter length:");

length = Convert.ToInt32(Console.ReadLine());

}

private int width;

public void ReadWidth()

{

Console.WriteLine("Enter width:");

width = Convert.ToInt32(Console.ReadLine());

}

public int CalculateArea()

{

return length \* width;

}

public int CalculatePerimeter()

{

return 2 \* (length + width);

}

}

class Triangle : IShape

{

private int Side1;

public void ReadSide1()

{

Console.WriteLine("Enter side 1");

Side1 = Convert.ToInt32(Console.ReadLine());

}

private int Side2;

public void ReadSide2()

{

Console.WriteLine("Enter side2");

Side2 = Convert.ToInt32(Console.ReadLine());

}

private int side3;

public void Readside3()

{

Console.WriteLine("Enter side3");

side3 = Convert.ToInt32(Console.ReadLine());

}

public int CalculateArea()

{

float s = (Side1 + Side2 + side3) / 2;

int Area = (int)Math.Sqrt((s \* (s - Side1) \* (s - Side2) \* (s - side3)));

return Area;

}

public int CalculatePerimeter()

{

return Side1 + Side2 + side3;

}

}

internal class Program

{

static void Main(string[] args)

{

Circle c = new Circle();

c.ReadRadius();

Console.WriteLine(c.CalculatePerimeter());

Console.WriteLine(c.CalculateArea());

Square s = new Square();

s.ReadSide();

Console.WriteLine(s.CalculatePerimeter());

Console.WriteLine(s.CalculateArea());

Rectangle r = new Rectangle();

r.ReadLength();

r.ReadWidth();

Console.WriteLine(r.CalculatePerimeter());

Console.WriteLine(r.CalculateArea());

Triangle t = new Triangle();

t.ReadSide1();

t.ReadSide2();

t.Readside3();

Console.WriteLine(t.CalculatePerimeter());

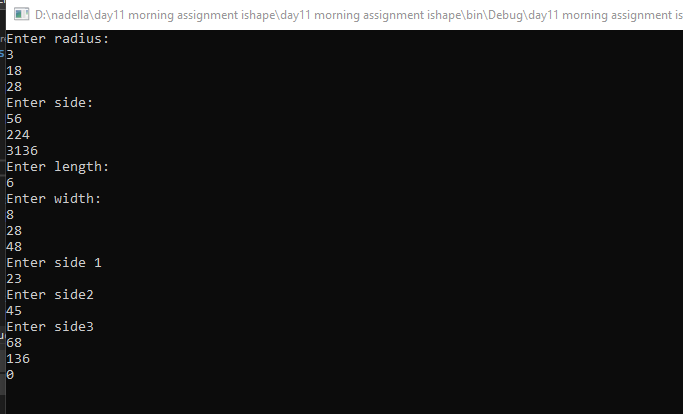
Console.WriteLine(t.CalculateArea());

Console.ReadLine();

}

}

}



2.write the seven points discussed about properties in class??

A property will have get and set methods

It is same like class variable

A property with only get is read only

A property with only set is write only

A property with get and set we can read and assign values

Property names must start with upper case

Properties are used to access private variables

A simple example of properties are::

Class employee

{

Private int id;

Private string name;

Private string designation;

Public int id

{

Get{return id;}

Set{id= value;}

}

}

5. write a sample code for properties as discussed in class

Id,name,designation,salary??

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace day11\_morning\_assignmentid\_name\_salary

{

class Employee

{

private int id;

private string name;

private string designation;

private int salary;

public int Id

{

get { return id; }

set { id = value; }

}

public string Name

{

get { return name; }

set { name = value; }

}

public string Designation

{

get { return designation; }

set { designation = value; }

}

public int Salary

{

get

{

salary = (designation == "s") ? 30000 : 60000;

return salary;

}

}

}

internal class Program

{

static void Main(string[] args)

{

Employee emp = new Employee();

emp.Designation = "m";

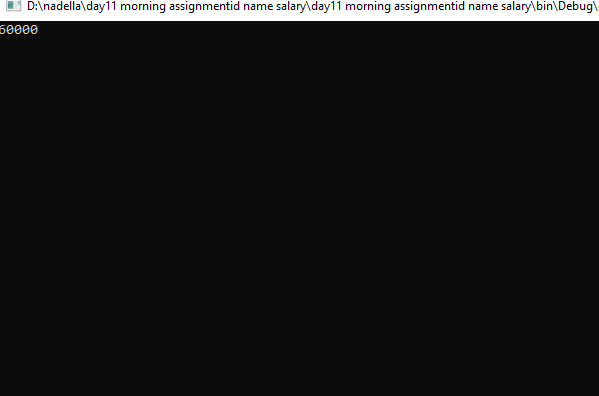
Console.WriteLine(emp.Salary);

Console.ReadLine();

}

}

}



7.create mathematics class and add three static methods???

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace day\_11\_assignments\_mathematics

{

class Mathematics

{

public static int Add(int a, int b)

{

return a + b;

}

public static int sub(int a, int b)

{

return a - b;

}

public static int mul(int a, int b)

{

return a \* b;

}

}

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine(Mathematics.Add(1045, 1578));

Console.WriteLine(Mathematics.sub(1007, 505));

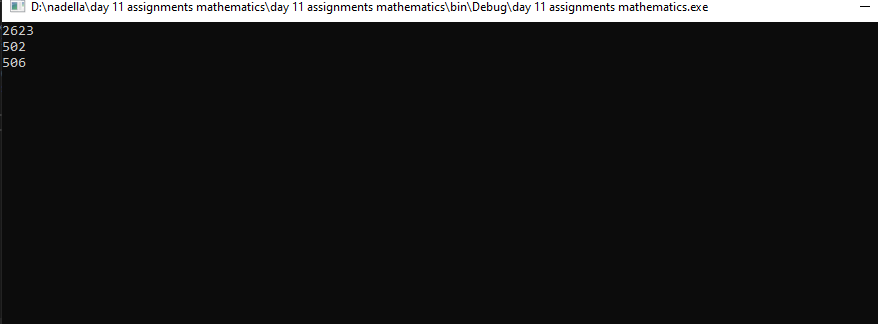
Console.WriteLine(Mathematics.mul(253, 2));

Console.ReadLine();

}

}

}



6. create a class employee with only properties??

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace day11\_assignment\_with\_properties

{

class Employee

{

private int id;

private string name;

private string designation;

private int salary;

public int Id { get; set; }

public string Name { get; set; }

public string Designation { get; set; }

public int Salary { get; set; }

}

internal class Program

{

static void Main(string[] args)

{

Employee emp = new Employee();

emp.Id = 11001;

Console.WriteLine($"id = {emp.Id}");

emp.Name = "lokesh";

Console.WriteLine($"name = {emp.Name}");

emp.Designation = "testing";

Console.WriteLine($"designation = {emp.Designation}");

emp.Salary = 80000;

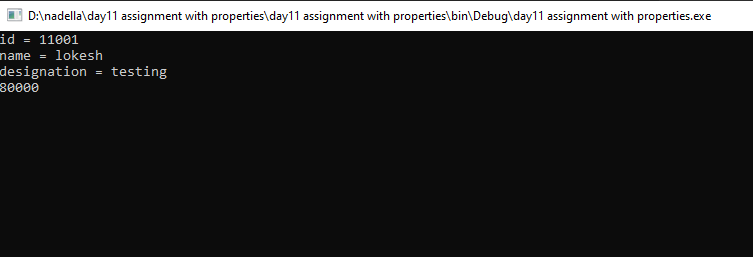
Console.WriteLine(emp.Salary);

Console.ReadLine();

}

}

}



6. six points discussed in class about interface???

Interface name should start with I

Interface acts like a contract

Any class that is implementing interface must override all the methods

Interface is a pure abstract class

By default the methods in interface are public

Interface support multiple interface