Day 11th Assignment

By

B.P.N.V.S.Sudheer

07-02-22

|  |  |
| --- | --- |
| 1.Research and write the difference between abstract class and interface in c# | |
| Abstract class | Interface |
| 1.It can contain different types of access modifiers like public,private etc.. | 1.It only contain public access modifier because everything in the interface is public |
| 2.The performance of an abstract class is fast | 2.The performance of interface is slow |
| 3.A class can only use abstract class | 3.A class can use multiple interface |
| 4.Multiple inheritance is not achieved by abstract class | 4.Multiple inheritance is acheived by interface |

|  |
| --- |
| 2.Write the 6 points about interface discussed in the class |
| * Interface is a pure abstract class |
| * Interface name should be start with “I” |
| * Interface acts like a contract |
| * By default the methods in interface are public and abstract |
| * Any class that is implementing interface must override all the methods |
| * Interface support multiple inheritance |

|  |
| --- |
| 3.Write the 7 points discussed about properties |
| * Properties are almost same as class variables with get and set |
| * A property with only get readonly |
| * A property with only set writeonly |
| * A property with get and set you can read value and assign value |
| * Properties are introduced to deal with private variables |
| * Example :   Class employee  {  Private int id;  Private int salary;  Public int id  {  get {return id;}  set {id = value;}  }  } |
| * Property name start with upper class |

|  |
| --- |
| 4.Research and understand when to create static methods |
| The static keyword is used to create method that will exist independently of any instances created for the class **s**tatic methods do not use any instance variables of any object of the class they are defined in. Static methods take all the data from parameters and compute something from those parameters, with no reference to variables. |

|  |
| --- |
| 5.Write sample code to illustrate properties as discussed in class  Id-get, set  Name – get,set  Designation-set (writeonly)  Salary-get (get with some functionality) |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;    namespace \_11th\_day\_project2  {  class employee  {  private int id;  private string name;  private string designation;  private int salary;  public int Id  {  get { return id; }  set { Id = value; }  }  public int Name  {  get { return Name; }  set { Name = value; }  }  public string Designation  {  set { Designation = value; }    }  public int Salary  {  get  {  salary = (designation == "s") ? 3000 : 6000;  return salary;  }  }  }  internal class Program  {  static void Main(string[] args)  {  employee e = new employee();  e.Designation = "t";  Console.WriteLine(e.Salary);  e.Id = 89;  Console.ReadLine();  }  }  } |

|  |
| --- |
| 6.Write example program for interfaces discussed in the class  Ishape  Include the classes  Circle,square,triangle |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;    namespace \_11th\_day\_project1  {  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 Triangle : Ishape  {  private int a;  private int b;  private int c;        public int calculateperimeter()  {  return a + b + c;  }    public int calculatearea()  {  return a \* b \* c;  }  public void ReadTriangle()  {    Console.WriteLine("enter a value");  a = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("enter b value");  b = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("enter c value");  c = Convert.ToInt32(Console.ReadLine());  }  }  class square : Ishape  {  private int side;    public int calculateperimeter()  {  return 4 \* side;  }    public int calculatearea()  {  return side \* side;  }  public void Readside()  {  Console.WriteLine("enter side");  side = Convert.ToInt32(Console.ReadLine());  }    }    internal class Program  {  static void Main(string[] args)  {  circle c = new circle();  c.Readradius();  Console.WriteLine(c.calculateperimeter());  Console.WriteLine(c.calculatearea());  Triangle t = new Triangle();  t.ReadTriangle();  Console.WriteLine(t.calculateperimeter());  Console.WriteLine(t.calculatearea());  square s = new square();  s.Readside();  Console.WriteLine(s.calculateperimeter());  Console.WriteLine(s.calculatearea());            Console.ReadLine();          }  }  } |
| Output: |
|  |

|  |
| --- |
| 7.Create Mathematics class and add 3 static methods and call the methods in main method |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;    namespace \_11\_th\_day\_project3  {  class mathematics  {  public static int add(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(5, 6));  Console.WriteLine(mathematics.mul(5, 6));  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| 8.create a class employee with only properties |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;    namespace \_11th\_day\_project2  {  class employee  {  private int id;  private string name;  private string designation;  private int salary;  public int Id  {  get { return id; }  set {Id= value; }  }  public int Name  {  get { return Name; }  set { Name = value; }  }  public string Designation  {  Get { return Designation;}  set { Designation = value; }    }  public int Salary  {  Get { return Salary;}  Set { Salary =Value;}  }  } |