.Net Programming Lab

Assingment-2

Name: V. Sai Nikhil

Reg No: 16MIS0257

Create a dll for watercooler object with necessary methods, properties, fields, and constructor. List all the types available using the concept of reflection and store the count in registry.

Watercoolerclass.dll

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Microsoft.Win32;

namespace watercoolerclass
{
    public class cooler
    {
        public RegistryKey rk;
        public string cname,cap;
        public int cmodelno,rem;
        public cooler()
        {
            rk=Registry.CurrentUser.OpenSubKey("Software\\Microsoft\\cooler",true);
            if(rk==null)
                rk=Registry.CurrentUser.CreateSubKey("Software\\Microsoft\\cooler");
        }
        public string name
        {
            set { cname = value; }
            get { return cname; }
        }
        public int modelno
        {
            set { cmodelno = value; }
            get { return cmodelno; }
        }
        public void capacity()
        {
            cap = "1000 litres";
            Console.WriteLine("The capacity of the water is:{0}", cap);
            rk.SetValue("capacity", cap);
        }
        public void fillwater()
        {
            rem = 600;
            rk.SetValue("fillwater", rem);
```

```
        }
        public void collecteater()
        {
            rem = 300;
            rk.SetValue("collectwater", rem);
        }
        public void wastewater()
        {
            rem = 100;
            rk.SetValue("wastewater", rem);
        }


    }
}
```

Gendral.dll

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Reflection;

namespace gendral
{
    public class reflect
    {
        public void get_m(object obj)
        {
            Type t = obj.GetType();
            MethodInfo[] mi = t.GetMethods();
            foreach (MethodInfo m in mi)
            {
                Console.WriteLine("Method Names:{0}",m.Name);
                ParameterInfo[] pi = m.GetParameters();
                foreach (ParameterInfo p in pi)
                {
                    Console.WriteLine("Parameter Name:{0}", p.Name);
                    Console.WriteLine("Parameter Position:{0}", p.Position);
                    Console.WriteLine("Parameter Type:{0}", p.ParameterType);
                    Console.WriteLine("parameter Member:{0}", p.Member);
                    Console.WriteLine("Parameter Raw Default Value:{0}",
p.RawDefaultValue);
                }
            }
        }
        public void get_f(object obj)
        {
            Type t = obj.GetType();
            FieldInfo[] fi = t.GetFields();
            foreach(FieldInfo f in fi)
            {
                Console.WriteLine("Fiels Names:{0}", f.Name);
            }
```

```
        }
        public void get_p(object obj)
        {
            Type t = obj.GetType();
            PropertyInfo[] ppi = t.GetProperties();
            foreach (PropertyInfo p in ppi)
            {
                Console.WriteLine("Property names:{0}", p.Name);
            }
        }
        public void get_c(object obj)
        {
            Type t = obj.GetType();
            ConstructorInfo[] ci = t.GetConstructors();
            foreach (ConstructorInfo c in ci)
            {
                Console.WriteLine("Constructor Names:{0}", c.Name);
            }
        }
        public void get_i(object obj)
        {
            Type t = obj.GetType();
            Type[] ti = t.GetInterfaces();
            foreach (Type o in ti)
            {
                Console.WriteLine(o.Name);
                MethodInfo[] mi = o.GetMethods();
                foreach (MethodInfo m in mi)
                    Console.WriteLine("Interface Method Names:{0}", m.Name);
                FieldInfo[] fi = o.GetFields();
                foreach (FieldInfo f in fi)
                    Console.WriteLine("Interface Fiels Namws:{0}", f.Name);
                PropertyInfo[] ppi = t.GetProperties();
                foreach (PropertyInfo p in ppi)
                    Console.WriteLine("Interface Property Names:{0}", p.Name);
                ConstructorInfo[] ci = t.GetConstructors();
                foreach (ConstructorInfo c in ci)
                    Console.WriteLine("Interface COnstructor Names:{0}", c.Name);
            }
        }
        public void get_oth(object obj)
        {
            Type t = obj.GetType();
            Console.WriteLine("Is CLass:{0}", t.IsClass);
            Console.WriteLine("Is Abstract:{0}", t.IsAbstract);
            Console.WriteLine("Is Sealed:{0}", t.IsSealed);
            Console.WriteLine("Is Serializable:{0}", t.IsSerializable);
            Console.WriteLine("Is Array:{0}", t.IsArray);
            Console.WriteLine("Is Interface:{0}", t.IsInterface);
            Console.WriteLine("Is Nested Private:{0}", t.IsNestedPrivate);
            Console.WriteLine("Is Nested Public:{0}", t.IsNestedPublic);
            Console.WriteLine("Is Value Type:{0}", t.IsValueType);
            Console.WriteLine("Is Enum:{0}", t.IsEnum);
        }
    }

}
```

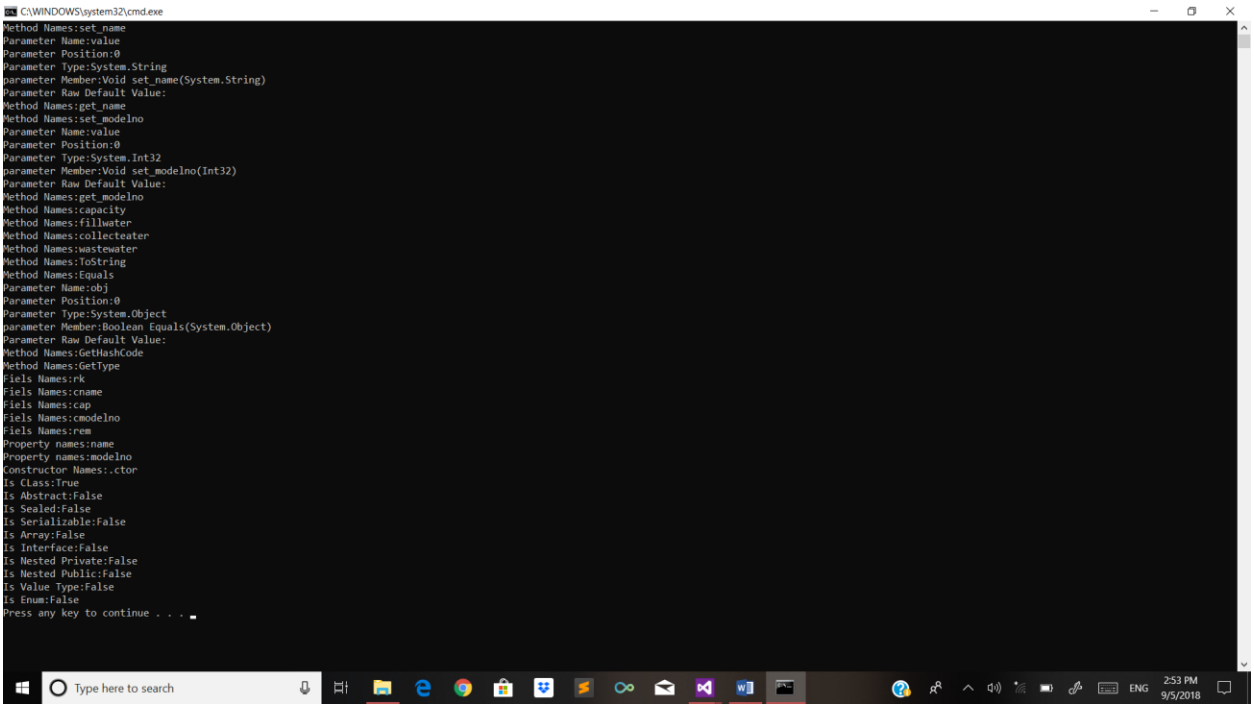
Watercooler console application:

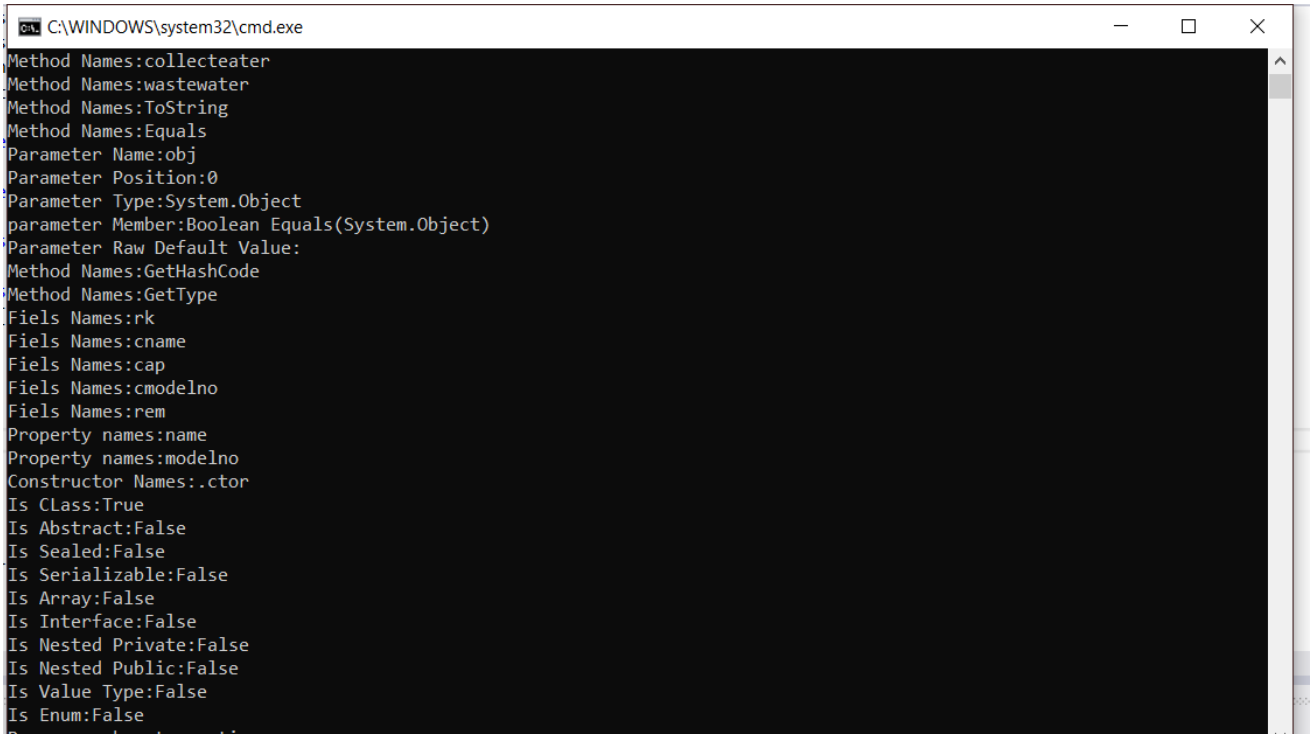
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using gendral;
using watercoolerclass;

public delegate void sd(object obj);

namespace watercooler
{
    class Program
    {
        static void Main(string[] args)
        {
            gendral.reflect r1 = new gendral.reflect();
            sd sd1 = new sd(r1.get_m);
            sd1 += new sd(r1.get_f);
            sd1 += new sd(r1.get_p);
            sd1 += new sd(r1.get_c);
            sd1 += new sd(r1.get_i);
            sd1 += new sd(r1.get_oth);
            watercoolerclass.cooler c1 = new watercoolerclass.cooler();
            sd1(c1);
        }
    }
}
```

Output:


C:\WINDOWS\system32\cmd.exe
Method Names:set_name
Parameter Name:value
Parameter Position:0
Parameter Type:System.String
parameter Member:Void set_name(System.String)
Parameter Raw Default Value:
Method Names:get_name
Method Names:set_modelno
Parameter Name:value
Parameter Position:0
Parameter Type:System.Int32
parameter Member:Void set_modelno(Int32)
Parameter Raw Default Value:
Method Names:get_modelno
Method Names:capacity
Method Names:fillwater
Method Names:collecteater
Method Names:wastewater
Method Names:ToString
Method Names:Equals
Parameter Name:obj
Parameter Position:0
Parameter Type:System.Object
parameter Member:Boolean Equals(System.Object)
Parameter Raw Default Value:
Method Names:GetHashCode
Method Names:GetType
Fiels Names:rk
Fiels Names:cname
Fiels Names:cap
Fiels Names:cmodelno
Fiels Names:rem
Property names:name
Property names:modelno
Constructor Names:.ctor
Is Class:True
Is Abstract:False
Is Sealed:False
Is Serializable:False
Is Array:False
Is Interface:False
Is Nested Private:False
Is Nested Public:False
Is Value Type:False
Is Enum:False
Press any key to continue . . .
Type here to search
ENG
2:53 PM
9/5/2018


C:\WINDOWS\system32\cmd.exe
Method Names:collecteater
Method Names:wastewater
Method Names:ToString
Method Names:Equals
Parameter Name:obj
Parameter Position:0
Parameter Type:System.Object
parameter Member:Boolean Equals(System.Object)
Parameter Raw Default Value:
Method Names:GetHashCode
Method Names:GetType
Fiels Names:rk
Fiels Names:cname
Fiels Names:cap
Fiels Names:cmodelno
Fiels Names:rem
Property names:name
Property names:modelno
Constructor Names:.ctor
Is CLass:True
Is Abstract:False
Is Sealed:False
Is Serializable:False
Is Array:False
Is Interface:False
Is Nested Private:False
Is Nested Public:False
Is Value Type:False
Is Enum:False