**POLYMORPHISM &Interfaces**

**Objective**

In this lab our main target is to familiarize ourselves with interfaces and to implement interface inheritance. By using classes we can access interfaces.

**Software Tools**

Microsoft Visual Studio

**Procedure**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplicationpolymorphisim

{

class Program

{

class Birds

{

public virtual void speak()

{

Console.WriteLine("Birds name");

}

}

class sparrow : Birds

{

public override void speak()

{

Console.WriteLine("Sparrow speak");

}

}

class pigeon : Birds

{

public override void speak()

{

Console.WriteLine("\nPigeon speak");

}

}

class parrot : Birds

{

public override void speak()

{

Console.WriteLine("\nparrot speak");

}

}

static void Main(string[] args)

{

Birds s1 = new Birds();

Console.WriteLine("Select speak\n\n1) sparrow \n2) pigeon \n3) parrot\n");

int a = Convert.ToInt32(Console.ReadLine());

switch (a)

{

case 1:

{

sparrow t = new sparrow();

s1 = t;

s1.speak();

break;

}

case 2:

{

pigeon c = new pigeon();

s1 = c;

s1.speak();

break;

}

case 3:

{

parrot c = new parrot();

s1 = c;

s1.speak();

break;

}

}

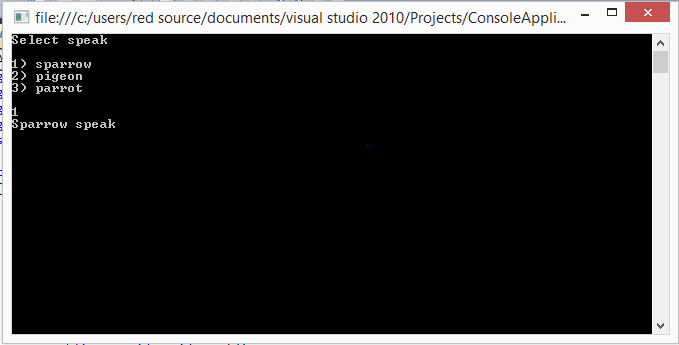
Console.ReadKey();

}

}

}

**Output:**



1. **Task 2:**

Interface SimpleRemote :

**Input:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace wwww

{

public interface IsimpleRemote

{

void on();

void off();

void VolUP();

void VolDown();

void Reverse();

void Forward();

void Pause();

}

}

1. **Task 3:**

Interface ExtendedRemoteControl:

Input:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleInterfacelab6

{

public interface IExtendedRemote

{

void fineTune();

void View();

void setSource();

void openBrowser();

void nextChanel();

void prevChanel();

}

}

**Input:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplicationwwww

{

class Program

{

public interface IExtendedRemote

{

void fineTune();

void View();

void setSource();

void openBrowser();

void nextChanel();

void prevChanel();

}

public class samsung : IExtendedRemote

{

public void fineTune()

{

Console.WriteLine("now you are in the fine\_tune mode");

}

public void View()

{

Console.WriteLine("now you are in the view mode");

}

public void setSource()

{

Console.WriteLine("now you are in the set-source mode");

}

public void openBrowser()

{

Console.WriteLine("now you are in the open-browser mode");

}

public void nextChanel()

{

Console.WriteLine("now you are in the next chanel mode");

}

public void prevChanel()

{

Console.WriteLine("now you are in the previous chanel mode");

}

}

class sony : IExtendedRemote

{

public void fineTune()

{

Console.WriteLine("now you are in the fine\_tune mode");

}

public void View()

{

Console.WriteLine("now you are in the view mode");

}

public void setSource()

{

Console.WriteLine("now you are in the set-source mode");

}

public void openBrowser()

{

Console.WriteLine("now you are in the open-browser mode");

}

public void nextChanel()

{

Console.WriteLine("now you are in the next chanel mode");

}

public void prevChanel()

{

Console.WriteLine("now you are in the previous chanel mode");

}

static void Main(string[] args)

{

IExtendedRemote m = new samsung();

m.fineTune();

m.View();

m.setSource();

m.openBrowser();

m.nextChanel();

m.prevChanel();

Console.ReadLine();

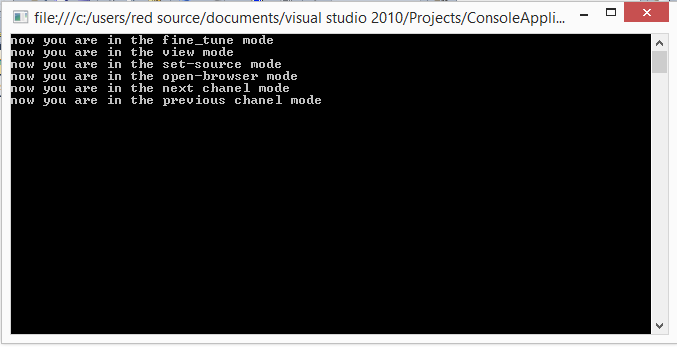
}

}

}

}

**Output:**

****

**Conclusion**

By the end of the lab I understood all the concepts regarding polymorphism and interfaces, I completed all my task by using those concepts.