

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace PaternNVI

{

internal abstract class Car

{

public string Name { get; set; }

protected Car(string name)

{

Name = name;

}

public abstract void Motor();

}

class BmwCar:Car

{

public BmwCar(string name) : base(name)

{

}

public override void Motor()

{

Console.WriteLine("Work {0} Motor",Name);

ColorCar();

Klapan();

MaxSpeed();

}

private void ColorCar()

{

ConsoleColor result;

result=(ConsoleColor) Enum.Parse(typeof(ConsoleColor), "Green");

//ConsoleColor.TryParse("green", true, out result);

Console.WriteLine(result);

Console.ForegroundColor = result;

Console.WriteLine("This Color Car - green");

Console.ResetColor();

}

protected virtual void Klapan()

{

Console.WriteLine("Klapana 4");

}

protected virtual void MaxSpeed()

{

Console.WriteLine("Max Speed 210");

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace PaternNVI

{

class Program

{

class BMWX5:BmwCar

{

public BMWX5(string name) : base(name)

{

}

protected override void MaxSpeed()

{

Console.WriteLine("Speed 250");

}

}

static void Main(string[] args)

{

BMWX5 x5=new BMWX5("BMW X5");

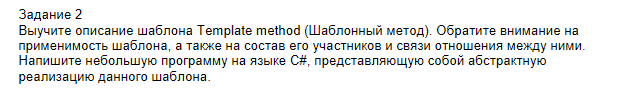
x5.Motor();

Console.ReadKey();

}

}

}



using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Ad\_hoc\_polimorf

{

class TextRedactor

{

public string NameFile { get; set; }

public TextRedactor(string nameFile)

{

NameFile = nameFile;

}

public void Write(string str)

{

FileStream fr = File.Open(NameFile, FileMode.OpenOrCreate,FileAccess.Write);

fr.Seek(0, SeekOrigin.End);

using (StreamWriter sw = new StreamWriter(fr))

{

sw.Write(str);

}

}

public void Read()

{

FileStream fr = File.Open(NameFile, FileMode.OpenOrCreate,FileAccess.Read);

using (StreamReader sr=new StreamReader(fr))

{

while (!sr.EndOfStream)

{

Console.WriteLine(sr.ReadLine());

}

}

}

}

}

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Xml;

namespace Ad\_hoc\_polimorf

{

class XMLRedactor

{

public string NameFile { get; set; }

protected virtual void CreateDocument()

{

using (XmlWriter writer = new XmlTextWriter(NameFile,Encoding.Default)

{

Formatting = Formatting.Indented,

Indentation = 1,

IndentChar = '\t'

})

{

writer.WriteStartDocument();

writer.WriteStartElement("Root");

writer.WriteAttributeString("myText",**DateTime**.Now.ToShortDateString());

//writer.WriteEndAttribute();

writer.WriteStartElement("textXML");

writer.WriteEndElement();

writer.WriteEndElement();

writer.WriteEndDocument();

}

}

public XMLRedactor(string nameFile)

{

NameFile = nameFile;

}

public void Write(string str)

{

CreateDocument();

XmlDocument doc = new XmlDocument();

doc.Load(NameFile);

XmlNode n=doc.SelectSingleNode("Root/textXML");

n.InnerText = str;

doc.Save(NameFile);

}

public void Read()

{

FileStream fr = File.Open(NameFile, FileMode.OpenOrCreate,FileAccess.Read);

XmlTextReader tr=new XmlTextReader(fr);

while (tr.Read())

{

Console.WriteLine(tr.ReadString());

}

tr.Close();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Ad\_hoc\_polimorf

{

class Program

{

static void Main(string[] args)

{

dynamic[] arr = {new TextRedactor("text.txt"), new XMLRedactor("textXml.xml")};

foreach (dynamic item in arr)

{

item.Write("static void Main(string[] args)" +Environment.NewLine+

"{" + Environment.NewLine +

"dynamic[] arr = {new TextRedactor('text.txt'), new XMLRedactor('textXml.xml')};" + Environment.NewLine +

"foreach (dynamic item in arr)" + Environment.NewLine +

" {" + Environment.NewLine +

" item.Write();" + Environment.NewLine +

" }" + Environment.NewLine +

"}");

}

Console.WriteLine(new string('-',50));

foreach (var item in arr)

{

item.Read();

Console.WriteLine();

}

Console.ReadKey();

}

}

}