## 实验6 掌握多态的应用

### 实验目的

理解多态的意义

了解多态的应用

掌握多态的C#实现

### 第一部分 上机训练

1、在俄罗斯方块程序中，有L形，T形，田形等多种形状，它们是图形的多种形态，可以创建一个名为Shape的基类，而后派生L形，T形等，之后可以在运行时动态绘制各种形状。

1）创建一个名位Teris的控制台应用程序、

2）各个类之间的关系如下图所示：



3）创建名为Shape的抽象类，包括ShapeType属性和显示信息方法，以及抽象方法绘制：

using System;

using System.Collections.Generic;

using System.Text;

namespace Teris

{

public abstract class Shape

{

private string shapeType;

public string ShapeType

{

get { return shapeType; }

set { shapeType = value; }

}

public void DisplayInfo()

{

Console.WriteLine("当前图形类型" + shapeType);

}

public abstract void Draw();

}

}

4）创建名为ShapeL的派生类，实现基类的绘制

using System;

using System.Collections.Generic;

using System.Text;

namespace Teris

{

public class ShapeL : Shape

{

public ShapeL()

{

ShapeType = "L形";

}

public override void Draw()

{

Console.WriteLine("|");

Console.WriteLine("|");

Console.Write("|");

Console.WriteLine("\_\_");

}

}

}

5）创建名为ShapeT的派生类，实现基类的绘制

using System;

using System.Collections.Generic;

using System.Text;

namespace Teris

{

public class ShapeT : Shape

{

public ShapeT()

{

ShapeType = "T形";

}

public override void Draw()

{

Console.WriteLine("\_\_\_\_\_\_\_");

Console.WriteLine(" |");

Console.WriteLine(" |");

}

}

}

6）创建名为ShapeZ的派生类，实现基类的绘制

using System;

using System.Collections.Generic;

using System.Text;

namespace Teris

{

public class ShapeZ : Shape

{

public ShapeZ()

{

ShapeType = "Z形";

}

public override void Draw()

{

Console.WriteLine("----");

Console.WriteLine(" |");

Console.WriteLine(" |");

Console.WriteLine(" ----");

}

}

}

7）创建名为ShapeBlock(田字形)的派生类，实现基类的绘制

using System;

using System.Collections.Generic;

using System.Text;

namespace Teris

{

class ShapeBlock : Shape

{

public ShapeBlock()

{

ShapeType = "田形";

}

public override void Draw()

{

Console.WriteLine(" \_\_\_\_\_\_\_\_\_");

Console.WriteLine("| | |");

Console.WriteLine("| | |");

Console.WriteLine(" --------- ");

Console.WriteLine("| | |");

Console.WriteLine("| | |");

Console.WriteLine(" --------- ");

}

}

}

8）在Program.cs中添加如下代码，实现随机生成图形：

using System;

using System.Collections.Generic;

using System.Text;

using System.Threading;

namespace Teris

{

enum ShapeType

{

Block = 0,

L,

T,

Z

}

class Program

{

static void Main(string[] args)

{

Random rnd = new Random();

while (true)

{

Shape shape = null;

ShapeType type = (ShapeType)rnd.Next(4);

switch (type)

{

case ShapeType.Block:

shape = new ShapeBlock();

break;

case ShapeType.L:

shape = new ShapeL();

break;

case ShapeType.T:

shape = new ShapeT();

break;

case ShapeType.Z:

shape = new ShapeZ();

break;

}

if (shape != null)

{

shape.DisplayInfo();

shape.Draw();

Console.WriteLine("----------------------");

Thread.Sleep(1000);//防止刷新太快,让系统停止1秒}

}

}

}

}

}

注意Random类和Thread类的使用

### 第二部分 实战提高

设计一个控制台应用程序，定义一个Shape类，具体要求如下：

（1）类中定义2个私有字段长度（length）、宽度（breadth）。

（2）类中定义相应公有属性分别对应上述2个字段；

（3）类中定义可重载的构造函数初始化上述2个字段；

（4）定义1个默认构造函数；

（5）类中定义公有方法输出对象的长度、宽度等详细信息；

（6）类中定义虚方法Draw，输出当前图型类别。

（7）在main方法中测试Shape类及方法。

定义一个Box类，父类为Shape，具体要求如下：

（1）类中定义3个私有字段长度（length）、宽度（breadth）、高度（height）；

（2）类中定义相应公有属性分别对应上述3个字段；

（3）类中定义可重载的构造函数初始化上述3个字段；

（4）定义1个默认构造函数；

（5）类中定义公有方法输出对象的长度、宽度、高度等详细信息；

（6）类中定义虚方法Draw，输出当前图型类别。

（7）在main方法中测试Box类及方法；

（8）在main方法中测试多态。

**代码和运行结果图请表示在下面：**

**代码如下：**

**using System;**

**using System.Collections.Generic;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**namespace MySoftwareOnDotnetFramework.Shape**

**{**

**public class Shape**

**{**

**private int length;**

**private int breadth;**

**public int Length**

**{**

**set**

**{**

**length = value;**

**}**

**get**

**{**

**return length;**

**}**

**}**

**public int Breadth**

**{**

**set**

**{**

**breadth = value;**

**}**

**get**

**{**

**return breadth;**

**}**

**}**

**public Shape() { }**

**public Shape(int length,int breadth)**

**{**

**this.length = length;**

**this.breadth = breadth;**

**}**

**public void display()//display the info of the shape**

**{**

**Console.WriteLine("length: " + length);**

**Console.WriteLine("breadth: " + breadth);**

**}**

**public virtual void printType()**

**{**

**Console.WriteLine("Type of the shape.");**

**}**

**}**

**}**

**运行结果如下：**

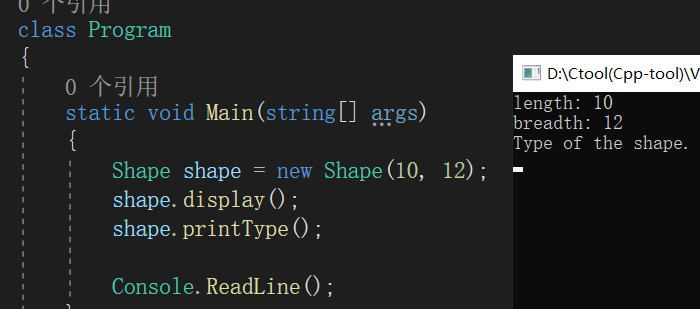


图6-1 对基类Shape的测试

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MySoftwareOnDotnetFramework.Shape

{

public class Box:Shape

{

private int height;

public int Height

{

set

{

height = value;

}

get

{

return height;

}

}

public Box() { }

public Box(int length,int breath,int height) : base(length, breath)

{

this.height = height;

}

public override void display()

{

base.display();

Console.WriteLine("height: " + height);

}

public override void printType()

{

Console.WriteLine("This is a box.");

}

}

}

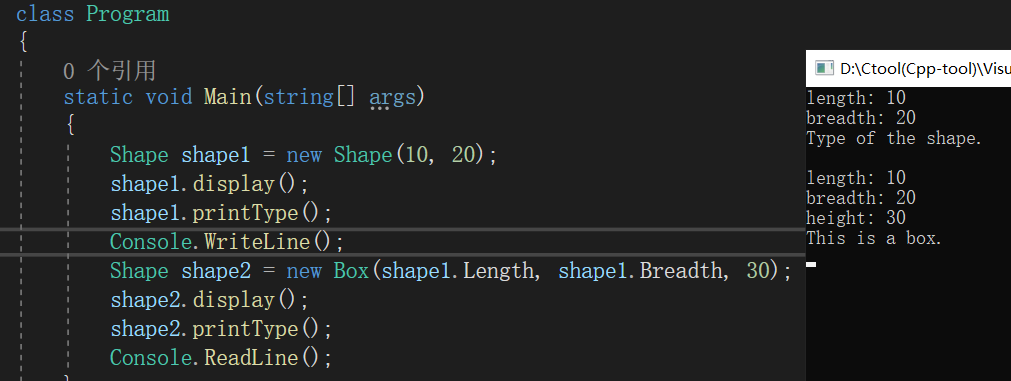


图6-2 对派生类Box的测试

### 第三部分 知识总结

|  |  |
| --- | --- |
| **学生总结** | 请将问题答案写在下面空白处 |
| 完成上机训练模块花费的时间 | 10分钟 |
| 完成实战提高模块花费的时间 | 35分钟 |
| 本次实验对你而言的难点 |  |
| 本次实验的收获 |  |
| 希望老师强化讲解的知识要点 |  |
| 希望老师帮助回答的问题 |  |
| 本次实验自评分数  （五分制） | 4.7 |