《**面向对象程序设计》实验报告**

专业： 空间信息与数字技术 班级： 2002 学号： 202016060205

姓名： 王佳琪 指导老师： 李新广

**实验单元二 面向对象程序设计**

**评分标准 & 得分统计**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **分值**  **观测点** | **90-100分** | **70-89分** | **60-69分** | **0-59分** | **得分** |
| 程序框架和算法设计  （权重30%） | 框架完整，算法正确 | 框架较完整，  算法基本正确 | 框架不完整，  算法部分正确 | 框架不完整，  算法错误较多 |  |
| 数据定义和输入输出  （权重30%） | 数据定义正确、完整，输入输出正确且符合要求 | 数据定义错误较少、较完整，输入输出基本正确和符合要求 | 数据定义错误较多、不完整，输入输出有明显错误 | 数据定义混乱，输入输出错误较多 |  |
| 实验结果分析与总结  （权重30%） | 实验数据、结果、分析和总结完整准确 | 实验数据、结果、分析和总结较为完整准确 | 实验数据、结果、分析和总结部分完整 | 实验数据、结果、分析和总结有错误 |  |
| 实验报告质量  （权重10%） | 实验报告规范完整 | 实验报告较为规范完整 | 实验报告规范但不完整 | 实验报告不规范、不完整 |  |
| 总分 | | | | | |  |

**注意：**

1. 请修改班级、姓名、学号三部分内容，封面中其它内容（含格式）请勿修改；
2. 实验单元二占实验成绩的50%；
3. 请勿修改实验报告的已有内容和格式。

《**面向对象程序设计》实验报告**

专业： 空间信息与数字技术 班级： 2002 学号： 202016060205

姓名： 王佳琪 实验日期： 2021.11.18 指导老师： 李新广

**实验单元二 面向对象程序设计**

1. **实验题目**

实验四 面向对象程序设计（一）

1. **实验目的**

1. 掌握定义类和创建对象的方法。

2. 区分类的不同成员的定义方法，并学会控制其可访问性。

3. 理解方法重载的概念。

4. 熟悉VS中OOP工具的使用。

5. 理解静态类和静态成员的概念。

6. 掌握静态类的定义方法。

7. 掌握静态成员的定义及调用方法。

1. **实验内容**

实践教程第4章的案例4-1、4-2、4-3和4-4。

1. **实验环境**

Visual C# 2017

1. **实验代码及结果（程序运行结果请以截屏给出）**
2. 源程序代码：

using System;

namespace ex4\_1

{

class Student

{

private string id;

private string name;

private string sex;

private int age;

public string Id

{

get { return id; }

}

public string Name

{

get { return name; }

set { name = value; }

}

public string Sex

{

get { return sex; }

set { sex = value; }

}

public int Age

{

get { return age; }

set { age = value; }

}

public Student(string id)

{

this.id = id;

}

public string Show()

{

return String.Format("学号:{0},姓名:{1},性别:{2},年龄:{3}", Id, Name, Sex, Age);

}

}

class Program

{

static void Main(string[] args)

{

Student s = new Student("201302010316");

s.Name = "王雪梅";

s.Sex = "女";

s.Age = 18;

Console.WriteLine(s.Show());

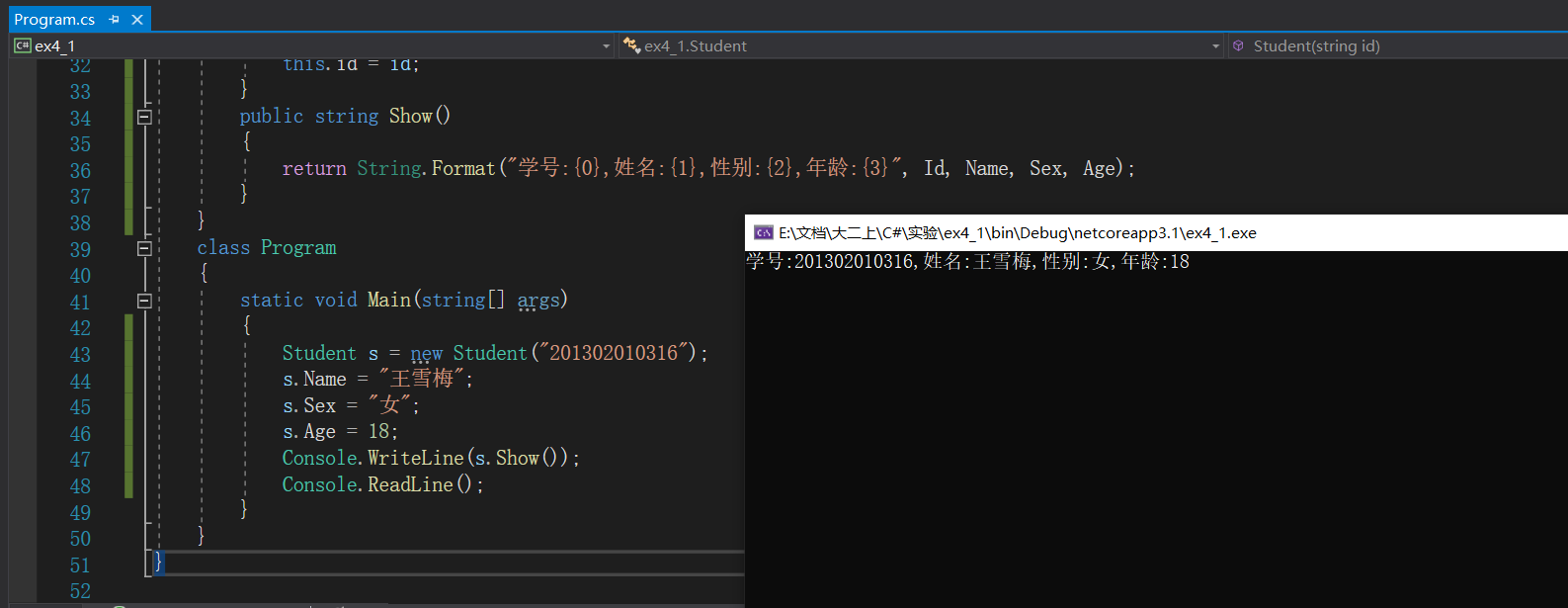
Console.ReadLine();

}

}

}

运行结果为：



1. 源程序代码：

using System;

namespace ex4\_2

{

class Maxer

{

public int Max(params int[] datas)

{

int k = 0;

for (int i = 1; i < datas.Length; i++)

{

if (datas[i] > datas[k])

k = i;

}

return datas[k];

}

public double Max(params double[] datas)

{

int k = 0;

for (int i = 1; i < datas.Length; i++)

{

if (datas[i] > datas[k])

k = i;

}

return datas[k];

}

public string Max(params string[] datas)

{

int k = 0;

for (int i = 1; i < datas.Length; i++)

{

if (datas[i].Length > datas[k].Length)

k = i;

}

return datas[k];

}

}

class Program

{

static void Main(string[] args)

{

Maxer m = new Maxer();

Console.Write( "{125,72,-12,37,89,165}中最大的整数为:");

Console.WriteLine(m.Max(125, 72, -12, 37, 89, 165));

Console.Write("{0.8,2.3,4.6,8.9,7.1}中最大的浮点数为:");

Console.WriteLine(m.Max(0.8, 2.3, 4.6, 8.9, 7.1));

Console.Write("{\"We\",\"are\",\"students\"}中最长的字符串为:");

Console.WriteLine(m.Max("We","are","students"));

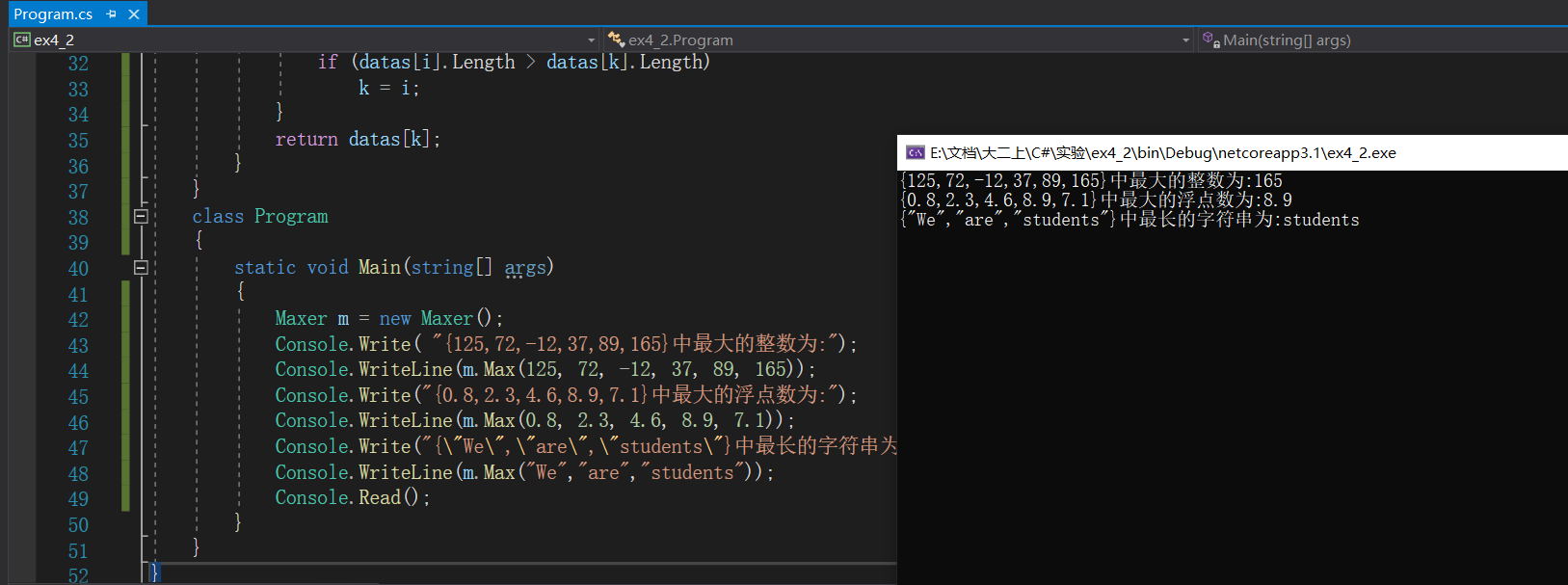
Console.Read();

}

}

}

运行结果为：



1. 源程序代码：

using System;

namespace ex4\_3

{

class Student

{

private static int numbers;

private string id;

private string name;

private string sex;

private int age;

public string Id

{

get { return id; }

}

public string Name

{

get { return name; }

set { name = value; }

}

public string Sex

{

get { return sex; }

set { sex = value; }

}

public int Age

{

get { return age; }

set { age = value; }

}

public Student(string id)

{

this.id = id;

numbers++;

}

public string Show()

{

return String.Format("学号:{0},姓名:{1},性别:{2},年龄:{3}", Id, Name, Sex, Age);

}

public static int ReturnNumber()

{

return numbers;

}

}

class Program

{

static void Main(string[] args)

{

Student s1 = new Student("201302010316");

s1.Name = "王雪梅";

s1.Sex = "女";

s1.Age = 18;

Console.WriteLine("学生"+Student.ReturnNumber().ToString()+"信息");

Console.WriteLine(s1.Show());

Student s2 = new Student("201210010105");

s2.Name = "张亮";

s2.Sex = "男";

s2.Age = 19;

Console.WriteLine("学生"+Student.ReturnNumber().ToString()+"信息");

Console.WriteLine(s2.Show());

Console.ReadLine();

}

}

}

运行结果为：

电脑萤幕的截图

描述已自动生成

1. 源程序代码：

using System;

namespace ex4\_4

{

static class Maxer

{

public static int Max(params int[] datas)

{

int k = 0;

for(int i = 0;i< datas.Length; i++)

{

if(datas[i] > datas[k])

{

k = i;

}

}

return datas[k];

}

public static double Max(params double[] datas)

{

int k = 0;

for(int i = 0;i<datas.Length;i++)

{

if(datas[i] > datas[k])

{

k = i;

}

}

return datas[k];

}

public static string Max(params string[] datas)

{

int k = 0;

for(int i=0;i<datas.Length;i++)

{

if(datas[i].Length > datas[k].Length)

{

k = i;

}

}

return datas[k];

}

}

class Program

{

static void Main()

{

Console.Write("{125,72,-12,37,89,165}中最大的整数为:");

Console.WriteLine(Maxer.Max(125, 72, -12, 37, 89, 165));

Console.Write("{0.8,2.3,4.6,8.9,7.1}中最大的浮点数为:");

Console.WriteLine(Maxer.Max(0.8, 2.3, 4.6, 8.9, 7.1));

Console.Write("{\"We\",\"are\",\"students\"}中最长的字符串为:");

Console.WriteLine(Maxer.Max("We", "are", "students"));

Console.Read();

}

}

}

运行结果为：

电脑萤幕的截图

描述已自动生成

**六、实验心得（请对每个实验题目编程及调试运行中遇到的问题及解决方案进行简单总结）**

这次的实验中学习了面向对象程序设计部分，在这次实验的学习中，我感受到了实验带来的很多学习经验，首先是实验一中的关于类的创建的部分，第一次使用了面向对象的思想进行程序设计，掌握了定义类和创建对象的方法。并且能够区分类的不同成员的定义方法，学会了控制其可访问性。而且在这次实验中我理解了方法重载的概念。方法的重载也就是在同一个类中声明了两个以上名称相同，但是参数类型或者参数个数不同的方法，用来实现对不同数据类型的相同处理。而在实验案例三四的部分了解学习了静态类和静态成员的概念，并且了解使用了静态类的定义方法。掌握静态成员的定义以及调用方法。静态类只能包含静态成员，而且静态类不能被实例化。这次的实验让我感受到了面向程序设计的魅力，也初步了解了面向对象程序设计的思维。

《**面向对象程序设计》实验报告**

专业： 空间信息与数字技术 班级： 2002 学号： 202016060205

姓名： 王佳琪 实验日期： 2021.11.20 指导老师： 李新广

**实验单元二 面向对象程序设计**

1. **实验题目**

实验四 面向对象程序设计（二）

1. **实验目的**

1. 理解类的继承性和多态性。

2. 掌握定义派生类的方法。

3. 理解构造函数的执行顺序。

4. 理解抽象类和接口的概念。

5. 掌握抽象类和接口的定义方法。

6. 掌握抽象方法的定义和实现。

1. **实验内容**

实践教程第5章的案例5-1、5-2、5-3和5-4。

1. **实验环境**

Visual C# 2017

1. **实验代码及结果（程序运行结果请以截屏给出）**
2. 源程序代码：

using System;

namespace ex5\_1

{

class Person

{

public string name;

public string Name

{

get { return name; }

set { name = value; }

}

private string sex;

public string Sex

{

get { return sex; }

set { sex = value; }

}

private DateTime birthday;

public DateTime Birthday

{

get { return birthday; }

set { birthday = value; }

}

public Person(string name,string sex)

{

this.name = name;

this.sex = sex;

}

public virtual string Output()

{

return String.Format("姓名:{0},性别:{1}", name, sex);

}

}

class Teacher : Person

{

private string school;

public string School

{

get { return school; }

set { school = value; }

}

private string profTitle;

public string ProfTitle

{

get { return profTitle; }

set { profTitle = value; }

}

public Teacher(string name, string sex, string school, string title) : base(name, sex)

{

this.school = school;

profTitle = title;

}

private int Age()

{

return DateTime.Now.Year - Birthday.Year;

}

public override string Output()

{

return String.Format("姓名:{0},性别:{1},年龄:{2},职称:{3},学校:{4}",Name,Sex,Age(),profTitle,School);

}

}

class Program

{

static void Main()

{

Person[] p = new Person[2];

p[0] = new Person("张艳", "女");

p[1] = new Teacher("王东", "男", "河北联合大学", "教授");

p[1].Birthday = new DateTime(1970,5,10);

foreach(Person p1 in p)

{

Console.WriteLine(p1.Output());

}

Console.Read();

}

}

}

运行结果为：

电脑萤幕的截图

描述已自动生成

1. 源程序代码：

using System;

namespace ex5\_2

{

class Program

{

static void Main()

{

Bora b1 = new Bora("冀 B s6882");

Console.Read();

}

}

class Car

{

public Car()

{

Console.WriteLine("汽车实例：");

}

}

class Bora:Car

{

public Bora()

{

Console.Write("宝来");

}

public Bora(string platenumber):this()

{

Console.WriteLine(platenumber);

}

}

}

运行结果为：

电脑萤幕的截图

描述已自动生成

1. 源程序代码：

using System;

namespace ex5\_2

{

class Program

{

abstract class Shape

{

public abstract double Perimeter();

public abstract double Area();

}

class Rectangle : Shape

{

private double len;

private double wid;

public Rectangle (double len,double wid)

{

this.len = len;

this.wid = wid;

}

public override double Area()

{

return (len + wid) / 2;

}

public override double Perimeter()

{

return len \* wid;

}

}

class Circle : Shape

{

private double radius;

public Circle(double r)

{

radius = r;

}

public override double Perimeter()

{

return 2 \* Math.PI \* radius;

}

public override double Area()

{

return Math.PI \* radius \* radius;

}

}

static void Main()

{

Rectangle r = new Rectangle(6, 8);

Console.WriteLine("长方形的周长为：{0}，面积为{1}",r.Perimeter(),r.Area());

Circle c = new Circle(5);

Console.WriteLine("圆的周长为：{0}，面积为{1}", Math.Round(c.Perimeter(), 2), Math.Round(c.Area(), 2));

Console.Read();

}

}

}

运行结果为：

电脑萤幕的截图

描述已自动生成

1. 源程序代码：

using System;

namespace ex5\_4

{

public class Program

{

interface IRegister

{

string Register();

}

class Student : IRegister

{

string name;

string sex;

string sno;

public Student (string name, string sex, string no)

{

this.name = name;

this.sex = sex;

this.sno = no;

}

string IRegister.Register()

{

return String.Format("学生注册信息如下:\n姓名：{0}，性别：{1}，学号：{2}",name,sex,sno);

}

}

class Teacher : IRegister

{

string name;

string sex;

string tno;

public Teacher (string name, string sex,string no)

{

this.name = name;

this.sex = sex;

this.tno = no;

}

string IRegister.Register()

{

return String.Format("教师注册信息如下:\n姓名：{0}，性别：{1}，学号：{2}", name, sex, tno);

}

}

static void Main()

{

Student s1 = new Student("赵晨", "男", "201306020316");

Teacher t1 = new Teacher("王晓", "女", "1032");

IRegister[] r = { s1, t1 };

foreach(IRegister r1 in r)

{

Console.WriteLine(r1.Register());

}

Console.Read();

}

}

}

运行结果为：

电脑萤幕的截图

描述已自动生成

**六、实验心得（请对每个实验题目编程及调试运行中遇到的问题及解决方案进行简单总结）**

这次的实验主要进行了关于类的继承和多态的部分，首先需要去理解类的继承性和多态性，才能更好的进行程序的设计，不过在这次得实验中，我领悟到了关于定义派生类的方法，理解了构造函数的执行顺序。构造函数在一个类中也可以多次进行调用，实例化派生类时，必须先实例化它的基类，单参构造函数中使用this关键字会先去调用它的无参构造函数。在后两个实验中，要求使用了抽象类与接口的内容，在这些练习过后，我理解了关于抽象类和接口的概念，掌握了抽象类和接口的定义方法，掌握了抽象方法的定义和实现。抽象类和抽象方法使用关键字abstract声明。抽象类除了拥有普通的类成员外，还有抽象类成员。

《**面向对象程序设计》实验报告**

专业： 空间信息与数字技术 班级： 2002 学号： 202016060205

姓名： 王佳琪 实验日期： 2021.11.20 指导老师： 李新广

**实验单元二 面向对象程序设计**

1. **实验题目**

实验五 Windows编程（一）

1. **实验目的**

1. 掌握Windows应用程序的开发步骤。

2. 掌握界面布局的步骤和技术。

3. 掌握标签、按钮、文本框、列表框和组合框的应用。

4. 掌握单选按钮、复选按钮和分组框的应用。

1. **实验内容**

实践教程第6章的案例6-1、6-2和6-3。

1. **实验环境**

Visual C# 2017

1. **实验代码及结果（程序运行结果请以截屏给出）**
2. 源程序代码：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ex6\_1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void btnPay\_Click(object sender, EventArgs e)

{

double yfk, sfk, zk, yh;//应付款，实付款，折扣，优惠

try

{

yfk = Convert.ToDouble(textBox1.Text);

}

catch

{

MessageBox.Show("输入数据为空或有误，请重新输入！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

textBox1.Focus();

return;

}

if (yfk < 250)

zk = 0;

else if (yfk < 500)

zk = 0.05;

else if (yfk < 1000)

zk = 0.075;

else if (yfk < 2000)

zk = 0.1;

else

zk = 0.15;

yh = Math.Round(yfk \* zk, 2);

sfk = yfk - yh;

textBox2.Text = yfk.ToString();

textBox3.Text = zk.ToString();

textBox4.Text = yh.ToString();

textBox5.Text = sfk.ToString();

}

}

}

运行结果为：

电脑萤幕的截图

描述已自动生成

1. 源程序代码：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ex6\_2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void btnAdd\_Click(object sender, EventArgs e)

{

if(textBox1.Text != "")

{

listBox1.Items.Add(textBox1.Text);

textBox1.Clear();

}

else

{

MessageBox.Show("请输入要添加的列表项的内容！","提示",MessageBoxButtons.OK,MessageBoxIcon.Information);

textBox1.Focus();

}

}

private void btnModify\_Click(object sender, EventArgs e)

{

textBox1.Text = listBox1.Text;

textBox1.Focus();

btnAdd.Enabled = btnModify.Enabled = btnDelete.Enabled = false;

btnConfirm.Enabled = true;

}

private void btnConfirm\_Click(object sender, EventArgs e)

{

listBox1.Items[listBox1.SelectedIndex] = textBox1.Text;

textBox1.Clear();

btnConfirm.Enabled = false;

btnAdd.Enabled = btnModify.Enabled = btnDelete.Enabled = true;

}

private void btnDelete\_Click(object sender, EventArgs e)

{

listBox1.Items.Remove(listBox1.SelectedItem);

}

}

}

运行结果为：

电脑萤幕的截图

描述已自动生成

1. 源程序代码：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ex6\_3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

cmbFontName.SelectedIndex = 0;

cmbFontSize.SelectedIndex = 1;

}

private void cmbFontName\_SelectedIndexChanged(object sender, EventArgs e)

{

textBox1.Font = new Font(cmbFontName.Text,textBox1.Font.Size,textBox1.Font.Style);

}

private void cmbFontName\_KeyPress(object sender, KeyPressEventArgs e)

{

if(e.KeyChar == 13)

cmbFontName\_SelectedIndexChanged(sender, e);

}

private void cmbFontSize\_SelectedIndexChanged(object sender, EventArgs e)

{

textBox1.Font = new Font(textBox1.Font.FontFamily, Convert.ToInt32(cmbFontSize.Text), textBox1.Font.Style);

}

private void cmbFontSize\_KeyPress(object sender, KeyPressEventArgs e)

{

if(e.KeyChar == 13)

cmbFontSize\_SelectedIndexChanged(sender, e);

}

private void rdbRed\_Click(object sender, EventArgs e)

{

textBox1.ForeColor = Color.Red;

}

private void rdbGreen\_Click(object sender, EventArgs e)

{

textBox1.ForeColor = Color.Green;

}

private void rdbBlue\_Click(object sender, EventArgs e)

{

textBox1.ForeColor = Color.Blue;

}

private void rdbBlack\_Click(object sender, EventArgs e)

{

textBox1.ForeColor = Color.Black;

}

private void rdbGray\_Click(object sender, EventArgs e)

{

textBox1.ForeColor = Color.Gray;

}

private void rdbYellow\_Click(object sender, EventArgs e)

{

textBox1.ForeColor = Color.Yellow;

}

private void checkBox\_Click(object sender, EventArgs e)

{

Font currentFont = textBox1.Font;

FontStyle style = FontStyle.Regular;

if(chkBold.Checked)

{

style |= FontStyle.Bold;

}

if(chkItalic.Checked)

{

style |= FontStyle.Italic;

}

if (chkStrikeout.Checked)

{

style |= FontStyle.Strikeout;

}

if(chkUnderline.Checked)

{

style |= FontStyle.Underline;

}

textBox1.Font = new Font(currentFont.FontFamily, currentFont.Size, style);

}

}

}

运行结果为：

电脑萤幕的截图

描述已自动生成

**六、实验心得（请对每个实验题目编程及调试运行中遇到的问题及解决方案进行简单总结）**

这次的实验主要进行了一些关于windows应用程序的一些开发设置，在这些操作中需要灵活掌握多种控件的使用，经过反复多次练习之后，自己掌握了关于界面布局的步骤和技术，掌握了标签，按钮，文本框，列表框和组合框的应用，掌握了单选按钮，复选框和分组框的应用。在第一个实验中，需要进行一些简单的操作，过程还算顺利，而在第二个实验中发现修改的时候需要选中要修改的项，第三个实验的控件比较多，因此进行的时间也比较长，但也没有遇到什么大的问题。

《**面向对象程序设计》实验报告**

专业： 空间信息与数字技术 班级： 2002 学号： 202016060205

姓名： 王佳琪 实验日期： 2021.11.25 指导老师： 李新广

**实验单元二 面向对象程序设计**

1. **实验题目**

实验五 Windows编程（二）

1. **实验目的**

1. 掌握图片框的应用。

2. 掌握定时器的应用。

3. 掌握跟踪条、滚动条和进度条的应用。

4. 理解富文本框的应用。

1. **实验内容**

实践教程第6章的案例6-4、6-5和6-6。

1. **实验环境**

Visual C# 2017

1. **实验代码及结果（程序运行结果请以截屏给出）**
2. 源程序代码：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ex6\_4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

listBox1.SelectedIndex = 0;

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

string filename = Application.StartupPath + "\\image\\" + listBox1.SelectedItem.ToString() + ".jpg";

pictureBox1.Image = Image.FromFile(filename);

}

private void pictureBox1\_DoubleClick(object sender, EventArgs e)

{

timer1.Enabled = !timer1.Enabled;

}

private void timer1\_Tick(object sender, EventArgs e)

{

if(listBox1.SelectedIndex<listBox1.Items.Count-1)

{

listBox1.SelectedIndex += 1;

}

else

{

listBox1.SelectedIndex = 0;

}

}

}

}

运行结果为：

图形用户界面, 文本

描述已自动生成

1. 源程序代码：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ex6\_5

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

if(this.ClientRectangle.Height%10==0)

{

progressBar1.Maximum = this.ClientRectangle.Height / 10;

}

else

{

progressBar1.Maximum = this.ClientRectangle.Height / 10 + 1;

}

}

private void button1\_Click(object sender, EventArgs e)

{

label1.BackColor = Color.FromArgb(hScrollBar1.Value,hScrollBar2.Value,hScrollBar3.Value);

timer1.Enabled = true;

timer1.Interval = trackBar1.Value;

progressBar1.Visible= true;

}

private void timer1\_Tick(object sender, EventArgs e)

{

if(label1.Height>=this.ClientRectangle.Height)

{

timer1.Enabled=false;

progressBar1.Visible = false;

MessageBox.Show("填充完毕", "提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

progressBar1.Value++;

label1.Height += 10;

}

}

}

}

运行结果为：

电脑萤幕的截图

描述已自动生成

1. 源程序代码：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ex6\_6

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

richTextBox1.LoadFile("1.txt", RichTextBoxStreamType.PlainText);

comboBox1.SelectedIndex = 0;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

switch(comboBox1.SelectedIndex)

{

case 0:

richTextBox1.SelectionAlignment = HorizontalAlignment.Left;

break;

case 1:

richTextBox1.SelectionAlignment = HorizontalAlignment.Center;

break;

case 2:

richTextBox1.SelectionAlignment = HorizontalAlignment.Right;

break;

}

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

richTextBox1.SelectionBullet = !richTextBox1.SelectionBullet;

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

if(e.KeyChar == 13)

{

if (sender.Equals(textBox1))

{

richTextBox1.SelectionIndent = Convert.ToInt16(textBox1.Text);

}

else if (sender.Equals(textBox2))

{

richTextBox1.SelectionRightIndent = Convert.ToInt16(textBox2.Text);

}

else if (sender.Equals(textBox3))

{

richTextBox1.SelectionHangingIndent = Convert.ToInt16(textBox3.Text);

}

}

else if(!(e.KeyChar==8||e.KeyChar==45||e.KeyChar>='0'&&e.KeyChar<='9'))

{

MessageBox.Show("只能输入数字！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

e.Handled = true;

}

}

private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == 13)

{

if (sender.Equals(textBox1))

{

richTextBox1.SelectionIndent = Convert.ToInt16(textBox1.Text);

}

else if (sender.Equals(textBox2))

{

richTextBox1.SelectionRightIndent = Convert.ToInt16(textBox2.Text);

}

else if (sender.Equals(textBox3))

{

richTextBox1.SelectionHangingIndent = Convert.ToInt16(textBox3.Text);

}

}

else if (!(e.KeyChar == 8 || e.KeyChar == 45 || e.KeyChar >= '0' && e.KeyChar <= '9'))

{

MessageBox.Show("只能输入数字！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

e.Handled = true;

}

}

private void textBox3\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == 13)

{

if (sender.Equals(textBox1))

{

richTextBox1.SelectionIndent = Convert.ToInt16(textBox1.Text);

}

else if (sender.Equals(textBox2))

{

richTextBox1.SelectionRightIndent = Convert.ToInt16(textBox2.Text);

}

else if (sender.Equals(textBox3))

{

richTextBox1.SelectionHangingIndent = Convert.ToInt16(textBox3.Text);

}

}

else if (!(e.KeyChar == 8 || e.KeyChar == 45 || e.KeyChar >= '0' && e.KeyChar <= '9'))

{

MessageBox.Show("只能输入数字！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

e.Handled = true;

}

}

private void richTextBox1\_SelectionChanged(object sender, EventArgs e)

{

if(richTextBox1.SelectionBullet)

{

checkBox1.Checked = true;

}

else

{

checkBox1.Checked = false;

}

switch(richTextBox1.SelectionAlignment)

{

case HorizontalAlignment.Left:

comboBox1.SelectedIndex = 0;

break;

case HorizontalAlignment.Center:

comboBox1.SelectedIndex = 1;

break;

case HorizontalAlignment.Right:

comboBox1.SelectedIndex = 2;

break;

}

if(richTextBox1.SelectionIndent!=0)

textBox1.Text=richTextBox1.SelectionIndent.ToString();

else

textBox1.Text = "";

if (richTextBox1.SelectionRightIndent != 0)

textBox2.Text = richTextBox1.SelectionRightIndent.ToString();

else

textBox2.Text = "";

if (richTextBox1.SelectionRightIndent != 0)

textBox3.Text = richTextBox1.SelectionRightIndent.ToString();

else

textBox3.Text = "";

}

}

}

运行结果为：

图形用户界面, 文本, 应用程序

描述已自动生成

**六、实验心得（请对每个实验题目编程及调试运行中遇到的问题及解决方案进行简单总结）**

这次的三个实验代码量比较大，进行编程的时候也用了较多的时间，不过在一步步的进行下也算成功完成了实验。第一个实验是一个简单的图片播放器，比较简单，但是要注意图片的路径下要有对应的文件，然后就跟着操作一步一步进行就能轻松完成实验，第二个实验中要注意在界面显示出的情况下要注意不能手动修改界面的大小，否则会出现错误。第三个实验的代码量较大，但是难度并不算高。在这次的实验过后基本上熟悉了常用控件的使用，包括图片框，定时器，跟踪条等控件的使用。

《**面向对象程序设计》实验报告**

专业： 空间信息与数字技术 班级： 2002 学号： 202016060205

姓名： 王佳琪 实验日期： 2021.11.27 指导老师： 李新广

**实验单元二 面向对象程序设计**

1. **实验题目**

实验五 Windows编程（三）

1. **实验目的**

1. 掌握下拉菜单的设计和使用方法。

2. 掌握工具栏和状态栏的使用方法。

3. 掌握打开文件对话框和保存文件对话框的使用方法。

4. 掌握浏览文件夹对话框的使用方法。

5. 掌握字体对话框和颜色对话框的使用方法。

1. **实验内容**

实践教程第7章的案例7-2、7-3和7-4。

1. **实验环境**

Visual C# 2017

1. **实验代码及结果（程序运行结果请以截屏给出）**
2. 源程序代码：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

namespace ex7\_2

{

public partial class Form1 : Form

{

bool newFile = true;

string filename = "";

public Form1()

{

InitializeComponent();

}

private void tsbtnNew\_Click(object sender, EventArgs e)

{

this.Text = "File1";

richTextBox1.Clear();

newFile = true;

}

private void tsbtnOpen\_Click(object sender, EventArgs e)

{

if(tstxtFilename.Text=="")

{

MessageBox.Show("请在左侧的文本框输入文件名！");

tstxtFilename.Focus();

}

else

{

if(Path.GetExtension(tstxtFilename.Text.ToLower())==".txt")

filename = tstxtFilename.Text;

else if(Path.GetExtension(tstxtFilename.Text.ToLower())=="")

filename=tstxtFilename.Text+".txt";

else

{

MessageBox.Show("只能打开TXT文件", "提示");

return;

}

if(!File.Exists(filename))

{

MessageBox.Show("文件不存在，请重新输入！", "提示");

tstxtFilename.Focus();

return;

}

richTextBox1.LoadFile(filename,RichTextBoxStreamType.PlainText);

this.Text = filename;

tstxtFilename.Clear();

newFile=false;

}

}

private void tsbtnSave\_Click(object sender, EventArgs e)

{

if(!newFile)

{

richTextBox1.SaveFile(this.Text, RichTextBoxStreamType.PlainText);

return;

}

if(tstxtFilename.Text=="")

{

MessageBox.Show("请在左侧的文本框中输入文件名！", "提示");

tstxtFilename.Focus();

return;

}

filename = tstxtFilename.Text;

FileInfo f = new FileInfo(filename);

if(!f.Directory.Exists)

{

MessageBox.Show("指定路径不存在！", "提示");

tstxtFilename.Focus();

return;

}

if (Path.GetExtension(tstxtFilename.Text.ToLower()) == "")

filename += ".txt";

else if(Path.GetExtension(tstxtFilename.Text.ToLower()) != ".txt")

{

MessageBox.Show("只能保存成TXT格式！");

tstxtFilename.Focus();

return;

}

richTextBox1.SaveFile(filename,RichTextBoxStreamType.PlainText);

newFile=false;

this.Text = filename;

tstxtFilename.Clear();

}

private void tsbtnUndo\_Click(object sender, EventArgs e)

{

richTextBox1.Undo();

}

private void tsbtnRedo\_Click(object sender, EventArgs e)

{

richTextBox1.Redo();

}

private void tsbtnCopy\_Click(object sender, EventArgs e)

{

richTextBox1.Copy();

}

private void tsbtnCut\_Click(object sender, EventArgs e)

{

richTextBox1.Cut();

}

private void tsbtnPaste\_Click(object sender, EventArgs e)

{

richTextBox1.Paste();

}

private void richTextBox1\_SelectionChanged(object sender, EventArgs e)

{

if (richTextBox1.SelectedText == "")

tsbtnCopy.Enabled = tsbtnCut.Enabled = false;

else

tsbtnCopy.Enabled = tsbtnCut.Enabled = true;

}

}

}

运行结果为：

电脑萤幕的截图

描述已自动生成

1. 源程序代码：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ex7\_3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

toolStripStatusLabel3.Text = DateTime.Now.ToLongDateString();

toolStripStatusLabel4.Text = DateTime.Now.ToLongTimeString();

}

private void Form1\_MouseMove(object sender, MouseEventArgs e)

{

toolStripStatusLabel1.Text = "X:" + e.X.ToString();

toolStripStatusLabel2.Text = "Y:" + e.Y.ToString();

}

private void timer1\_Tick(object sender, EventArgs e)

{

toolStripStatusLabel4.Text = DateTime.Now.ToLongTimeString();

}

}

}

运行结果为：

电脑萤幕的截图

描述已自动生成

1. 源程序代码：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ex7\_4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void mnuOpenFile\_Click(object sender, EventArgs e)

{

openFileDialog1.Filter = "所有文件(\*.\*)|\*.\*|rtf文件(\*.rtf)|\*.rtf|文本文件(\*.txt)|\*.txt";

if(openFileDialog1.ShowDialog() == DialogResult.OK)

{

string filename = openFileDialog1.FileName;

if (filename.EndsWith(".rtf"))

richTextBox1.LoadFile(filename);

else if (filename.EndsWith(".txt"))

richTextBox1.LoadFile(filename, RichTextBoxStreamType.PlainText);

else

MessageBox.Show("只能打开TXT或RTF文件！", "提示");

}

}

private void mnuSaveFile\_Click(object sender, EventArgs e)

{

saveFileDialog1.Filter = "rtf文件(\*.rtf)|\*.rtf|文本文件(\*.txt)|\*.txt";

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

{

string filename = saveFileDialog1.FileName;

if (filename.EndsWith(".rtf"))

richTextBox1.SaveFile(filename);

else if (filename.EndsWith(".txt"))

richTextBox1.SaveFile(filename, RichTextBoxStreamType.PlainText);

else

MessageBox.Show("只能保存成TXT或RTF格式！", "提示");

}

}

private void munQuit\_Click(object sender, EventArgs e)

{

this.Close();

}

private void mnuPicture\_Click(object sender, EventArgs e)

{

openFileDialog1.Filter = "所有文件(\*.\*)|\*.\*|图片文件(\*.jpg;\*.bmp;\*.gif)|\*.jpg;\*.bmp;\*.gif";

if(openFileDialog1.ShowDialog()==DialogResult.OK)

{

Clipboard.SetDataObject(Image.FromFile(openFileDialog1.FileName));

richTextBox1.Paste();

}

}

private void mnuFont\_Click(object sender, EventArgs e)

{

fontDialog1.ShowDialog();

richTextBox1.SelectionFont = fontDialog1.Font;

richTextBox1.SelectionColor = fontDialog1.Color;

}

private void mnuBackColor\_Click(object sender, EventArgs e)

{

colorDialog1.ShowDialog();

richTextBox1.SelectionBackColor = colorDialog1.Color;

}

}

}

运行结果为：

电脑萤幕的截图

描述已自动生成

**六、实验心得（请对每个实验题目编程及调试运行中遇到的问题及解决方案进行简单总结）**

这次的实验是这一单元的最后一个实验，用到了大部分之前学习的内容，在今天的操作学习之后，也感觉自己对C#这个语言的代码越来越熟练了，对于面向对象的思想有了充分的认知，也了解了大部分控件的操作使用，今天完成这些实验后感受有许多，但主要还是学习到了许多东西。在这个实验第三个小实验中了解到了控件也是一个对象，使用了这个控件就相当于进行了实例化，在这些实践操作的帮助下，也算是对面向对象的思想有了进一步的了解。