

实验报告

课程名称	面向对象程序设计语言			实验名称	设计计算器程序		
系部	地理信息科学系	班级	地信班	姓名	许愿	学号	109092023XXX
成绩							

一、实验目的

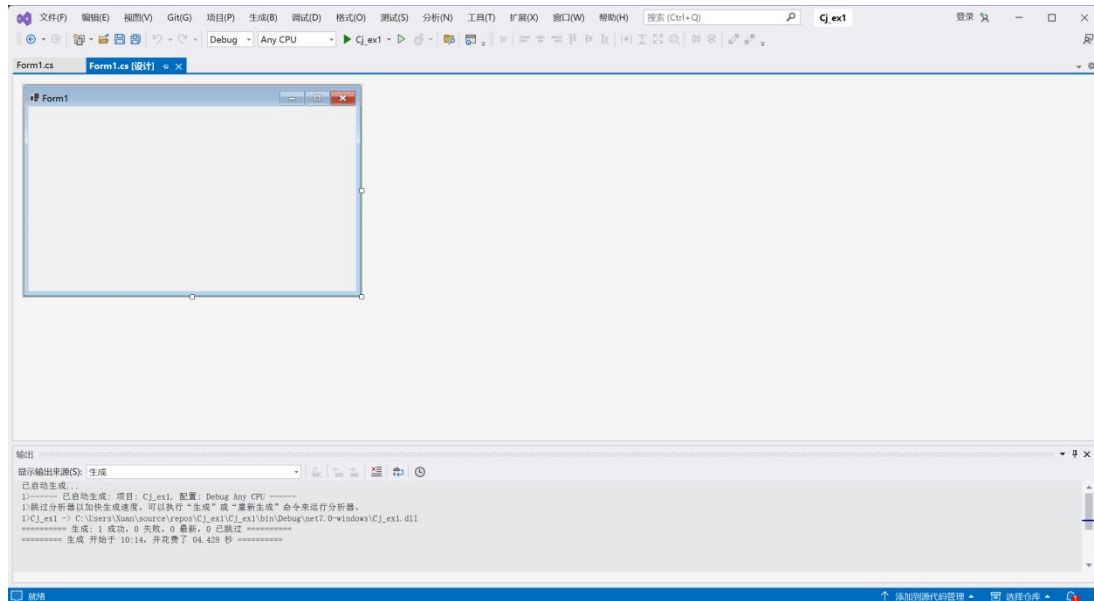
通过设计 Windows Form 计算器程序，了解 C#程序设计的基本过程、学习 C#的基本语法、初步掌握 Math 类的使用方法，提升使用 C#通过程序设计来解决生活中的实际问题的能力。

二、实验内容

1. 设计 Windows Form 计算器程序（参考系统自带计算器），实现加减乘除、求绝对值、Cos、Sin、Tan、Ctan、开方、乘方计算(参考 Math 类)。
2. 编写一个窗体程序，可计算并输出某一数字区间内能同时被 3,5 整除，且不是 100 的倍数的数。数字区间、每行输出个数可由用户输入指定，数与数之间间隔一个 Tab 宽度。

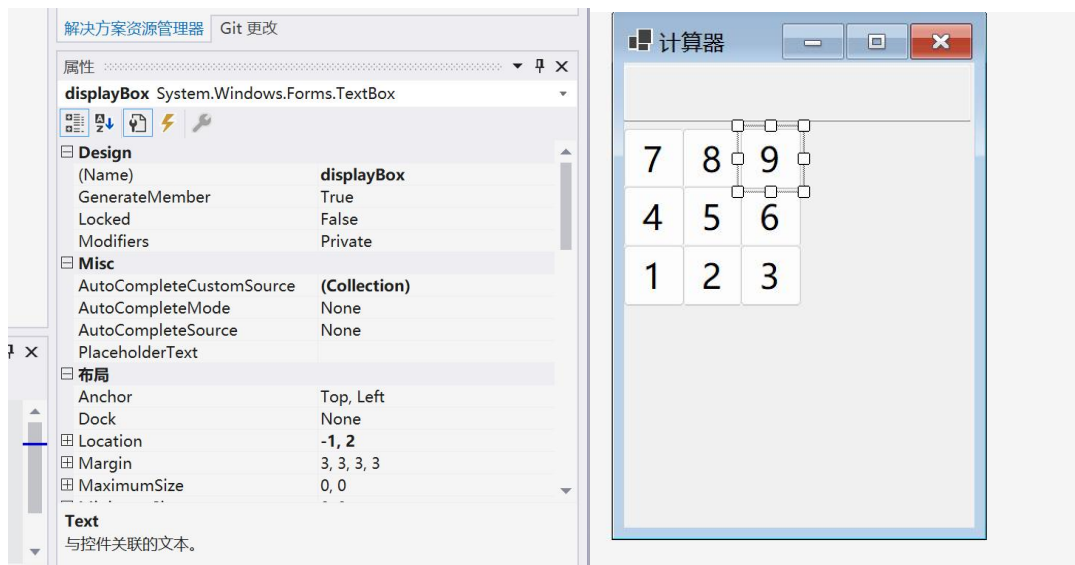
三、实验步骤

1. 设计 Windows Form 计算器程序
 - (1) 新建项目，命名为 Cj_ex1。



(2) 设计用户界面。

a. 从工具箱中添加一个 TextBox, 在属性中将其更名为 displayBox, 设置文本向右对齐、字号增大、只读。



b. 从工具箱中添加一个 Button, 在属性中将其更名为 btn1, 设置文本为 1, 字号增大。复制该按钮并修改, 完成各个按钮的添加。

c. 最终效果如下。



(3) 定义全局变量。计算器需要一些变量来存储当前的状态。

```
1 namespace Cj_ex1
2 {
3     3 个引用
4     public partial class Form1 : Form
5     {
6         double firstNumber = 0; // 第一个数字
7         string operation = ""; // 运算符
8         bool isClicked = false; // 是否点击了运算符按钮
9         1 个引用
10        public Form1 ()
11        {
12            InitializeComponent();
13        }
14    }
15 }
```

(4) 为 0-9 及小数点按钮定义事件，绑定 number_click 函数并编写函数。

```

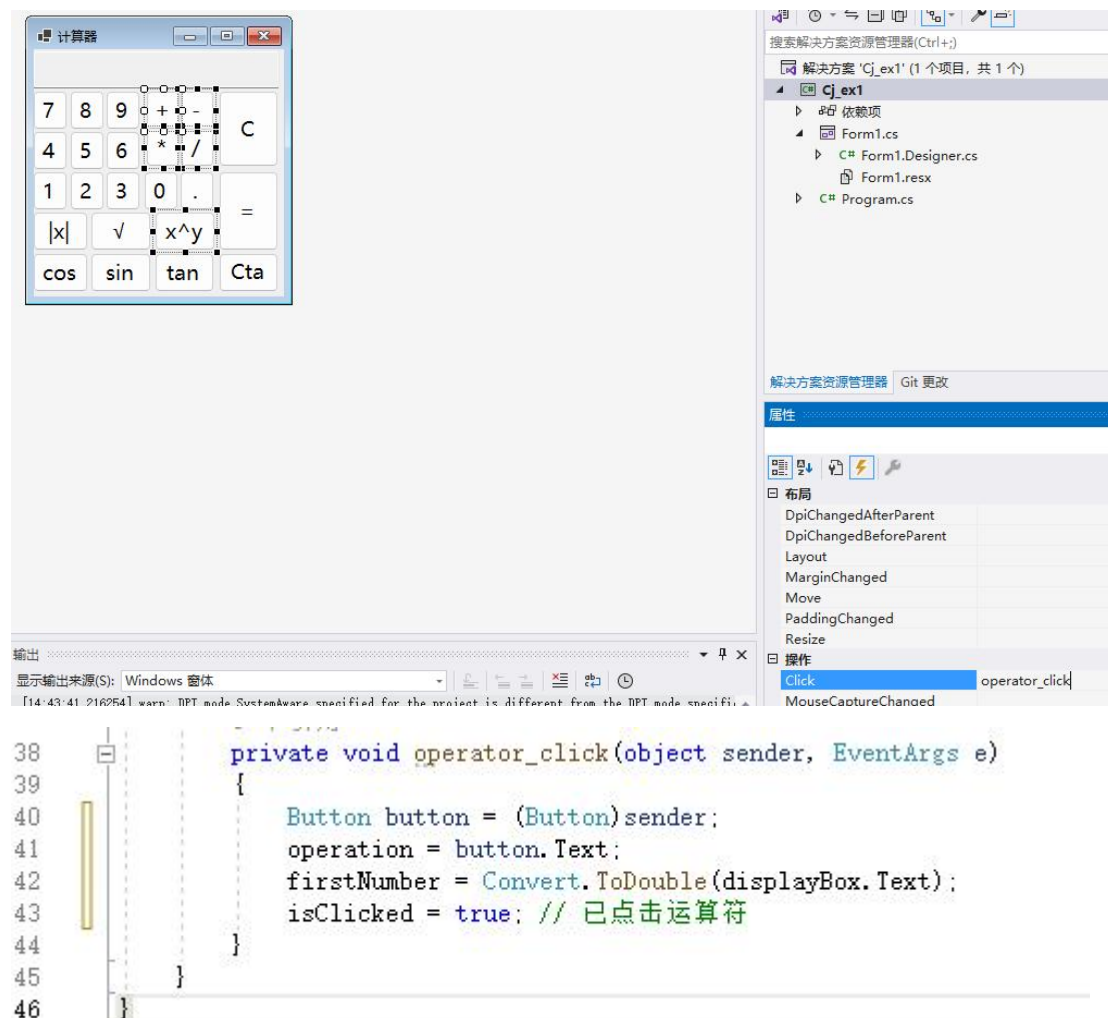
private void number_click(object sender, EventArgs e)
{
    // 如果显示的是0或刚刚按下了运算符，则清空显示屏
    if (displayBox.Text == "0" || isClicked) {
        displayBox.Clear();
    }

    isClicked = false;
    Button button = (Button)sender;

    // 如果按下的是小数点并且已经有小数点了就不再添加
    if (button.Text == ".") {
        if (!displayBox.Text.Contains(".")) {
            displayBox.Text += button.Text;
        }
    }
    else {
        displayBox.Text += button.Text;
    }
}

```

(5) 为运算符按钮定义事件，绑定 operator_click 函数并编写函数。



The screenshot shows a Windows Forms application titled "计算器" (Calculator). The interface includes a display box and buttons for digits (0-9), operators (+, -, *, /, %, ^), and trigonometric functions (cos, sin, tan, Cta). The code-behind (Form1.cs) defines an event handler for the operator buttons:

```

38 private void operator_click(object sender, EventArgs e)
39 {
40     Button button = (Button)sender;
41     operation = button.Text;
42     firstNumber = Convert.ToDouble(displayBox.Text);
43     isClicked = true; // 已点击运算符
44 }
45
46

```

The right-hand side of the image shows the Visual Studio IDE with the Solution Explorer displaying the project structure for "Cj_ex1", including Form1.cs, Form1.Designer.cs, Form1.resx, and Program.cs. The Properties window shows the "Click" event of the button is bound to the "operator_click" method.

(6) 为等于按钮定义事件，双击按钮绑定 btndengyu_Click 函数并编写

函数。

```
private void btndengyu_Click(object sender, EventArgs e)
{
    double secondNumber = Convert.ToDouble(displayBox.Text);
    double result = 0;

    switch (operation) { // 通过switch函数判断不同的运算符
        case "+":
            result = firstNumber + secondNumber;
            break;
        case "-":
            result = firstNumber - secondNumber;
            break;
        case "*":
            result = firstNumber * secondNumber;
            break;
        case "/":
            if (secondNumber == 0) { // 对除以0的情况做出特殊判断
                MessageBox.Show("不能除0");
                return;
            }
            result = firstNumber / secondNumber;
            break;
        case "x^y":
            result = Math.Pow(firstNumber, secondNumber);
            break;
    }

    displayBox.Text = result.ToString();
    operation = "";
}
```

(7) 为清除按钮定义事件，双击按钮绑定 btnC_Click 函数并编写函数。

```
1 个引用
79 private void btnC_Click(object sender, EventArgs e)
80 {
81     displayBox.Text = "0";
82     firstNumber = 0;
83     operation = "";
84     isClicked = false;
85 }
86
87 }
```

(8) 为 cos 等特殊函数按钮定义事件，绑定 spOperator_Click 函数并编写函数。

```

private void spOperator_Click(object sender, EventArgs e)
{
    Button button = (Button)sender;
    double number = Convert.ToDouble(displayBox.Text); // 转换为双精度
    double result = 0;

    double ang = number * Math.PI / 180.0; // 转换为弧度

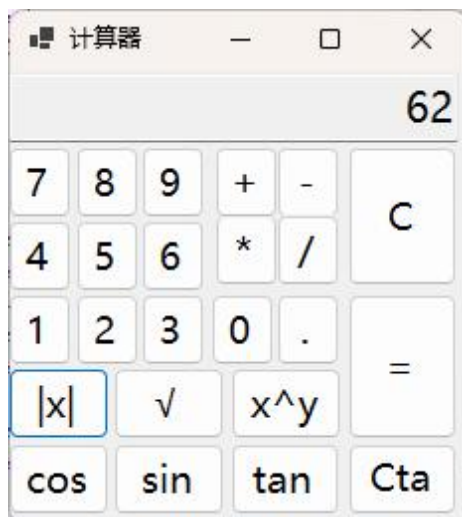
    switch (button.Text) {
        case "cos":
            result = Math.Cos(ang);
            break;
        case "sin":
            result = Math.Sin(ang);
            break;
        case "tan":
            result = Math.Tan(ang);
            break;
        case "Ctan":
            if (Math.Tan(ang) == 0) {
                MessageBox.Show("错误: 输入值无效!");
                return;
            }
            result = 1 / Math.Tan(ang);
            break;
        case "/":
            result = Math.Sqrt(number);
            break;
        case "|x|":
            result = Math.Abs(number);
            break;
    }

    displayBox.Text = result.ToString();
    isClicked = true;
}

```

(9) 运行测试。

- 输入 $12 + 8$ ，正常显示 20。
- 输入 $100 / 9$ ，正常显示 11.111111。
- 输入 $5 \times^y 3$ ，正常显示 125。
- 输入 90 sin，正常显示 1。
- 输入 - 55，输入时不显示负号，但点击=号后正常显示。点击|x|，正常变为 55。



2. 编写计算并输出能同时被 3,5 整除且不被 100 整除的区间数

(1) 新建项目，命名为 Cj_ex1_2。

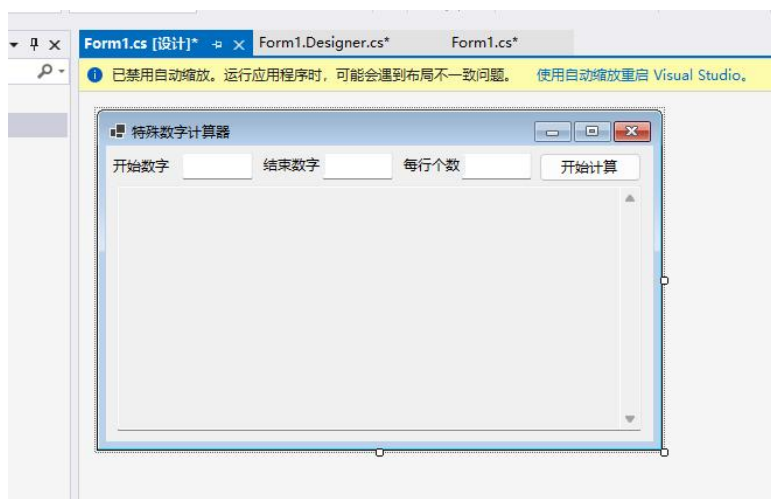
(2) 设计用户界面。

a. 从工具箱中拖出 TextBox 作为输入框和结果框（作为结果框时，需要开启 ReadOnly 只读、ScrollBars 改为 Vertical 以增加滚动条、开启 Multiline 以显示多行）。

b. 从工具箱中拖出 Label 作为显示输入框的提示词。

c. 从工具型中拖出 TextBox 作为开始计算按钮。

d. 最终效果如图所示。



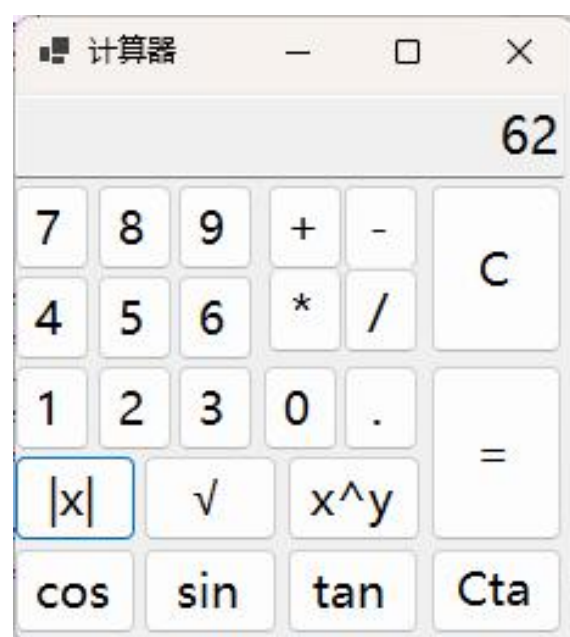
(3) 双击开始计算按钮进入事件函数 button1_Click，为其编写代码。

对输入的数据进行处理，防止出现错误。


```
Cj_ex1_2 Cj_ex1_2.Form1 button1_Click
11 {
12     int start, end, Geshu;
13     string result = "";
14
15     if (!int.TryParse(textBoxStart.Text, out start)){
16         MessageBox.Show("起始数字必须是一个有效的整数!");
17         return; // 中断执行
18     }
19     if (!int.TryParse(textBoxEnd.Text, out end)){
20         MessageBox.Show("结束数字必须是一个有效的整数!");
21         return;
22     }
23     if (!int.TryParse(textBoxGeshu.Text, out Geshu)){
24         MessageBox.Show("每行个数必须是一个有效的整数!");
25         return;
26     }
27     if (start > end){
28         MessageBox.Show("起始数字不能大于结束数字!");
29         return;
30     }
31     if (Geshu <= 0){
32         MessageBox.Show("每行个数必须大于0!");
33         return;
34     }
35
36     textBoxResult.Clear();
37
38     // 用于计算当前行已经输出了多少个数字
39     int countOnCurrentLine = 0;
40
41     // 使用 for 循环遍历从起始到结束的每一个数字
42     for (int i = start; i <= end; i++)
43     {
44         if ((i % 3 == 0) && (i % 5 == 0) && (i % 100 != 0)){
45             result += i;
46             result += "\t";
47             countOnCurrentLine++;
48             if (countOnCurrentLine >= Geshu){ // 达到换行条件
49                 result += "\r\n";
50                 countOnCurrentLine = 0;
51             }
52         }
53     }
54
55     textBoxResult.Text = result;
56
57     // 如果一个数都没找到，就提示
58     if (string.IsNullOrEmpty(textBoxResult.Text)){
59         textBoxResult.Text = "没有符合条件的数字";
60     }
61 }
62 }
```

(4) 运行测试，结果正常。

四、程序运行结果截图





五、程序源代码

1. 计算器

```
namespace Cj_ex1
{
    public partial class Form1 : Form
    {
        double firstNumber = 0; // 第一个数字
        string operation = ""; // 运算符
        bool isClicked = false; // 是否点击了运算符按钮
        public Form1()
        {
            InitializeComponent();
        }

        private void number_click(object sender, EventArgs e)
        {
            // 如果显示的是 0 或刚刚按下了运算符，则清空显示屏
            if (displayBox.Text == "0" || isClicked) {
                displayBox.Clear();
            }

            isClicked = false;
            Button button = (Button)sender;

            // 如果按下的是小数点并且已经有小数点了就不再添加
            if (button.Text == ".") {
                if (!displayBox.Text.Contains(".")) {
                    displayBox.Text += button.Text;
                }
            }
        }
    }
}
```

```

        }
    }
    else{
        displayBox.Text += button.Text;
    }
}

private void operator_click(object sender, EventArgs e)
{
    Button button = (Button)sender;
    operation = button.Text;
    firstNumber = Convert.ToDouble(displayBox.Text);
    isClicked = true; // 已点击运算符
}

private void btndengyu_Click(object sender, EventArgs e)
{
    double secondNumber = Convert.ToDouble(displayBox.Text);
    double result = 0;

    switch (operation){ // 通过 switch 函数判断不同的运算符
        case "+":
            result = firstNumber + secondNumber;
            break;
        case "-":
            result = firstNumber - secondNumber;
            break;
        case "*":
            result = firstNumber * secondNumber;
            break;
        case "/":
            if (secondNumber == 0){ // 对除以 0 的情况做出特殊判断
                MessageBox.Show("不能除 0");
                return;
            }
            result = firstNumber / secondNumber;
            break;
        case "x^y":
            result = Math.Pow(firstNumber, secondNumber);
            break;
    }

    displayBox.Text = result.ToString();
    operation = "";
}

```

```

    }

    private void btnC_Click(object sender, EventArgs e)
    {
        displayBox.Text = "0";
        firstNumber = 0;
        operation = "";
        isClicked = false;
    }

    private void spOperator_Click(object sender, EventArgs e)
    {
        Button button = (Button)sender;
        double number = Convert.ToDouble(displayBox.Text); // 转换为双精度
        double result = 0;

        double ang = number * Math.PI / 180.0; // 转换为弧度

        switch (button.Text) {
            case "cos":
                result = Math.Cos(ang);
                break;
            case "sin":
                result = Math.Sin(ang);
                break;
            case "tan":
                result = Math.Tan(ang);
                break;
            case "Ctan":
                if (Math.Tan(ang) == 0) {
                    MessageBox.Show("错误: 输入值无效!");
                    return;
                }
                result = 1 / Math.Tan(ang);
                break;
            case "√":
                result = Math.Sqrt(number);
                break;
            case "|x|":
                result = Math.Abs(number);
                break;
        }

        displayBox.Text = result.ToString();
    }

```

```

        isClicked = true;
    }
}

```

2. 获取区间指定数

```

namespace Cj_ex1_2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            int start, end, Geshu;
            string result = "";

            if (!int.TryParse(textBoxStart.Text, out start)) {
                MessageBox.Show("起始数字必须是一个有效的整数!");
                return; // 中断执行
            }
            if (!int.TryParse(textBoxEnd.Text, out end)) {
                MessageBox.Show("结束数字必须是一个有效的整数!");
                return;
            }
            if (!int.TryParse(textBoxGeshu.Text, out Geshu)) {
                MessageBox.Show("每行个数必须是一个有效的整数!");
                return;
            }
            if (start > end) {
                MessageBox.Show("起始数字不能大于结束数字!");
                return;
            }
            if (Geshu <= 0) {
                MessageBox.Show("每行个数必须大于 0!");
                return;
            }

            textBoxResult.Clear();

            // 用于计算当前行已经输出了多少个数

```

```

int countOnCurrentLine = 0;

// 使用 for 循环遍历从起始到结束的每一个数字
for (int i = start; i <= end; i++)
{
    if ((i % 3 == 0) && (i % 5 == 0) && (i % 100 != 0)) {
        result += i;
        result += "\t";
        countOnCurrentLine++;
        if (countOnCurrentLine >= Geshu) { // 达到换行条件
            result += "\r\n";
            countOnCurrentLine = 0;
        }
    }
}

textBoxResult.Text = result;

// 如果一个数都没找到，就提示
if (string.IsNullOrEmpty(textBoxResult.Text)) {
    textBoxResult.Text = "没有符合条件的数字";
}
}
}
}

```

六、收获，体会及问题

通过完成计算器与数字查找计算器这两个 C#窗体项目，我初步掌握了 Visual Studio 的可视化界面设计、控件使用及属性配置，尝试使用了循环、条件判断等逻辑来解决生活中可能会遇到的实际问题。

实践中，我也发现了我编写的程序依然存在不足，可能还需要进一步的修改和优化，故这些问题成为了我深入学习的动力。例如，计算器在计算如 $\sin 0.5$ 的数字时会生成 0.49999999994，此为 double 类型会产生的浮点数精度误差；计算器无法处理三个及以上数据连续的计算；数字查找器在处理大数据（比如起始 3 结束 99999999）时则会阻塞卡死一段时间。

这次经历让我认识到，要开发出高质量的应用，除了实现基本功能，还必须深入理解和选择功能模块，不断优化程序，采用更好的办法解决问题。这为我后续的学习指明了清晰的方向。