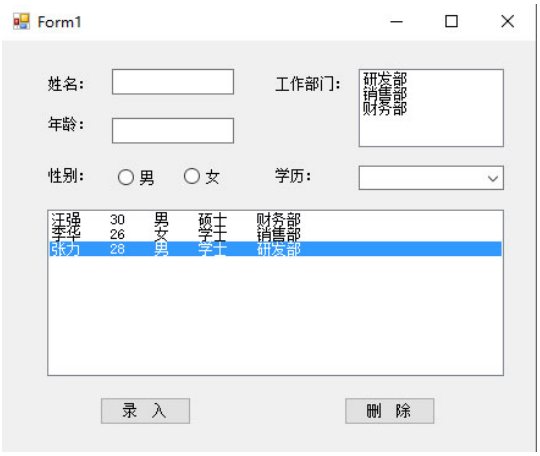
题目 1：

按下面图示程序。功能包括：

- 在窗体中设置姓名、年龄、性别、工作部门、学历之后，点击录入按钮，个人信息以一行字符串的形式显示的下方 ListBox 中。（注：由于姓名长度不同，可能各行信息中的每个字段不能对齐显示，因此本题目不要求各个字段对齐）
- 选中下方 ListBox 中某一行，点击删除按钮，可以将此行信息删除。（教材 p246）



第一步：

按照题目要求布局窗体，并设置合理的对象名。对于 departList（工作部门 listBox）对象，使用“编辑项”添加要求的项目。在 degreeCombo（学历 comboBox）对象也使用“编辑项”添加合理的项目。

第二步：

设计 Data 类用于封装每个人的信息。设计如下：

```
public class Data: Object {
    public string Name { get; set; }
    public int Age { get; set; }
    public bool Gender { get; set; }    // true male, false female
    public string Degree { get; set; }
    public string Depart { get; set; }
}
```

注意定义时要放在 Form1 类之下，否则会引起 IDE 报错。实例化 Data 类私有字段于 Form1 类中，并在 Form1 的构造函数中进行初始化。

第三步：

设计 buttonLeft 点击事件的响应函数。当这个函数被调用时，分别将 nameText, ageText, radioButton, departList, degreeCombo 中的数据记录于私有字段 data 中。检查有无空值，若无空值，则记录于 showList 中。注意这里需要调用自定义类 Data 的 ToString 虚函数。

```
private void button1_Click(object sender, EventArgs e) {
    bool ok = true;
    ok = ok && (nameText.Text.Length > 0);  data.Name = nameText.Text;
    ok = ok && (ageText.Text.Length > 0);
    try {
        data.Age = Convert.ToInt32(ageText.Text);
```

```
            } catch (FormatException) {
                ok = false;
            }
            ok = ok && (maleRadio.Checked || feRadio.Checked);
            if (maleRadio.Checked) data.Gender = true;
            else data.Gender = false;
            if (ok = ok && (degreeCombo.SelectedIndex >= 0))
                data.Degree = degreeCombo.SelectedItem.ToString();
            if (ok = ok && (departList.SelectedIndex >= 0))
                data.Depart = departList.SelectedItem.ToString();
            if (ok) showList.Items.Add(data.ToString());
        }

    public class Data: Object {
        public string Name { get; set; }
        public int Age { get; set; }
        public bool Gender { get; set; }    // true male, false female
        public string Degree { get; set; }
        public string Depart { get; set; }
        public override string ToString() {
            return String.Format("{0}\t{1}\t{2}\t{3}\t{4}", Name, Age, Gender ? "男" :
"女", Degree, Depart);
        }
    }
```

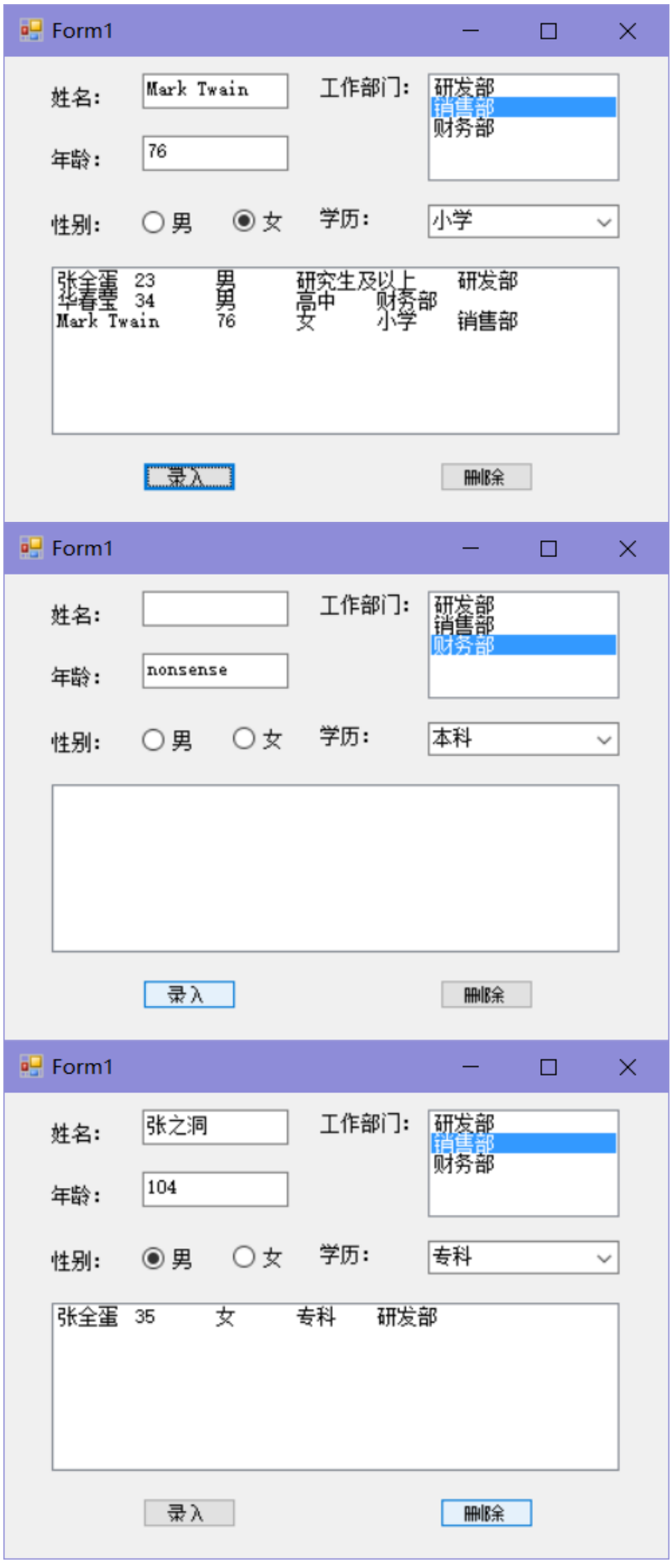
第四步：

设计删除按钮的功能，其点击事件的响应函数移除 showList 中的选中项。实现如下：

```
        private void button2_Click(object sender, EventArgs e) {
            if (showList.SelectedIndex >= 0) {
                showList.Items.Remove(showList.SelectedItem);
            }
```

注意删除之前需要检查是否有选中项。（无选中项时 showList.SelectItem == -1）。当然如果不嫌麻烦也可以用 try-catch 实现。

运行截图：


Form1
姓名:
Mark Twain
工作部门:
研发部
销售部
财务部
年龄:
76
性别:
男
女
学历:
小学
张全蛋 23 男 研究生及以上 研发部
华春莹 34 男 高中 财务部
Mark Twain 76 女 小学 销售部
录入
删除
Form1
姓名:
工作部门:
研发部
销售部
财务部
年龄:
nonsense
性别:
男
女
学历:
本科
录入
删除
Form1
姓名:
张之洞
工作部门:
研发部
销售部
财务部
年龄:
104
性别:
男
女
学历:
专科
张全蛋 35 女 专科 研发部
录入
删除

题目 2：

实现如下图所示计算器：

1) 输入数字后，数字在第一行文本框中显示（图 1 为初始态，图 2 正在输入）；
2) 按下加号或乘号后，第二行显示部分算式，如图 3；
3) 继续输入第二个数字时，仍在第一行文本框中显示（图 4 正在输入）；
4) 按下等号后，计算结果，如图 5 所示。

注意：只计算 A * B =C 或 A + B =C 的形式，不计算多个数据连续运算。

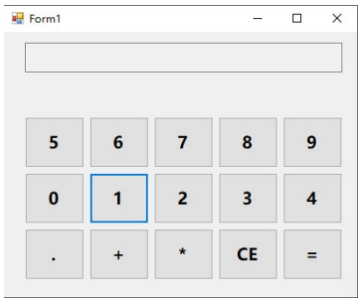


图 1

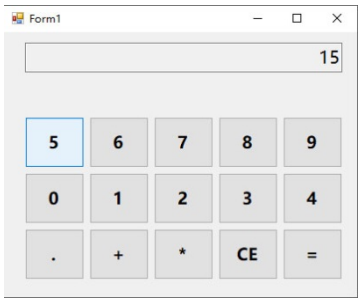


图 2

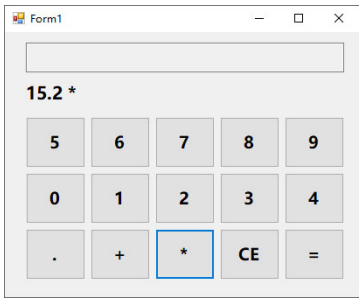


图 3

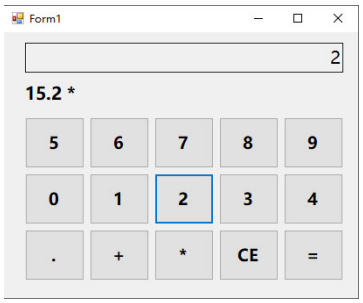


图 4

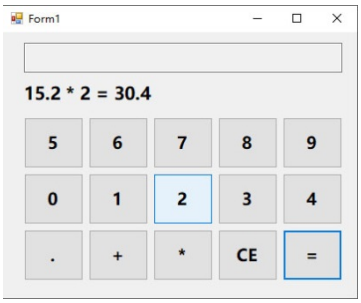


图 5

第一步：

按照图示建立窗口与图形界面。创建按钮的具体方法：创建一个 3x5 的 TableLayout，创建一个 Button，放置在左上角的格中，调节其 Font, TextAlign 属性及其大小，设置 Dock 为 None 以居中在单元格内。将设置好的 Button 复制 14 份，并逐一修改其 Text。

创建两个 Label，将顶部 Label 的 AutoSize 取消，设置为右对齐。底部 Label 的 AutoSize 取消，设置为左对齐。

第二步：

创建计算器所需的数据结构。由于本计算器最多只用到 3 个变量，故在 Form1 类中声明 3 个 double 类型的私有字段 paraLeft, paraRight, result，并在构造函数中均初始化为 0。再另声明一个 sign 私有字段用于存储符号。实现如下：

```
public Form1() {
    InitializeComponent();
    paraLeft = 0.0; paraRight = 0.0; result = 0.0;
    sign = true;
}
```

```
private double paraLeft;
private double paraRight;
private double result;
private bool sign;
```

第三步：

设计 button_Click 私有方法，将其作为所有数字按钮 Click 事件的响应函数。其实现的功能为：数字键被按下后，向 labelTop 中添加此字符，以便后续转化为浮点数。实现如下：

```
private void button_Click(object sender, EventArgs e) {
    labelTop.Text = labelTop.Text + ((Button)sender).Text;
}
```

第四步：

设计对于加号和乘号按钮的功能。设计这两个按钮共同的点击事件响应函数，完成下述功能：将 labelTop 中的字符串转化为浮点数并保存到 paraLeft 中，清空 labelTop，在 labelBottom 中写入该数字。若捕获到异常则不记录任何数据，清空 labelTop 并回复到初始状态。若无异常，则在 sign 中存储该运算符号，并在 labelBottom 中添加该字符。

```
private void sign_Click(object sender, EventArgs e) {
    try {
        paraLeft = Convert.ToDouble(labelTop.Text);
        labelTop.Text = "";
        labelBottom.Text = paraLeft.ToString();
        sign = ((Button)sender).Text[0] == '+';
        labelBottom.Text = labelBottom.Text + (sign ? "+" : "*");
    } catch (FormatException) {
        labelTop.Text = "";
        paraLeft = 0.0;
    }
}
```

第五步：

实现等号按钮点击的响应函数。实现以下功能：与第四步类似地处理 labelTop 中的数据，捕获异常，并将数据存储在 paraRight 中。根据 sign 决定 paraLeft 与 paraRight 之间的运算，并显示结果于 labelBottom 中

```
private void button15_Click(object sender, EventArgs e) {
    try {
        paraRight = Convert.ToDouble(labelTop.Text);
        labelTop.Text = "";
        labelBottom.Text = labelBottom.Text + paraRight.ToString();
        result = sign ? paraLeft + paraRight : paraLeft * paraRight;
        labelBottom.Text = labelBottom.Text + "=" + result.ToString();
    } catch (FormatException) {
        labelTop.Text = "";
        paraRight = 0.0;
```

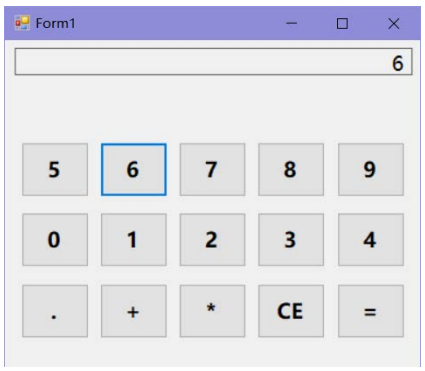
```
        }
    }
```

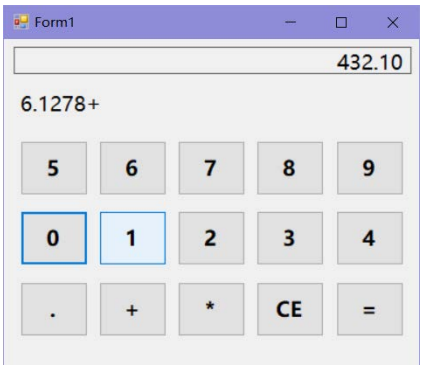
第六步：

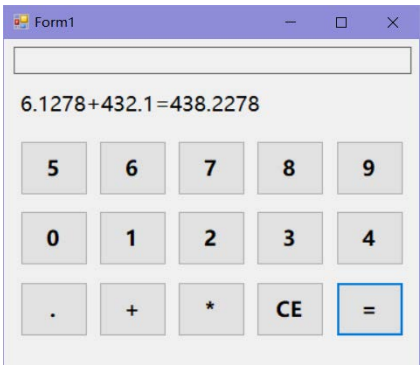
实现 CE 按钮功能。实现其 Click 事件响应函数如下：

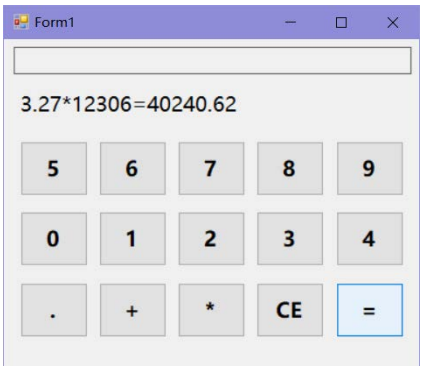
```
    private void button14_Click(object sender, EventArgs e) {
        labelTop.Text = labelBottom.Text = "";
        paraLeft = paraRight = result = 0.0;
        sign = true;
    }
```

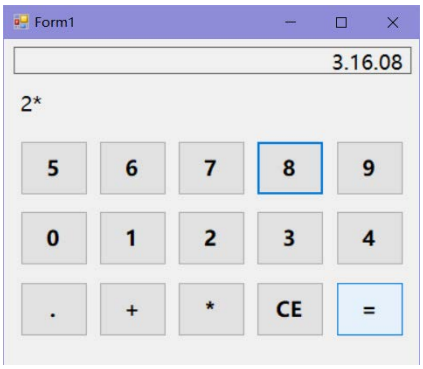
运行截图：











最后一幅截图演示了错误的输入格式不会带来报错。