

系统分析与设计课程设计实验报告

学号：

姓名：

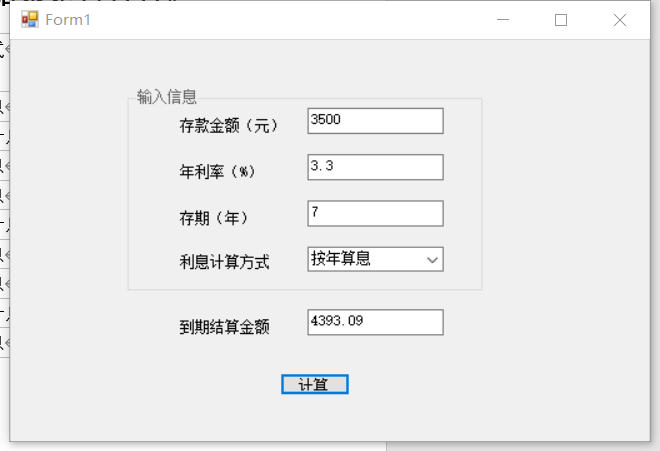
指导老师：

完成时间：

# 实验二(一级标题)

## 实验结果（二级标题）



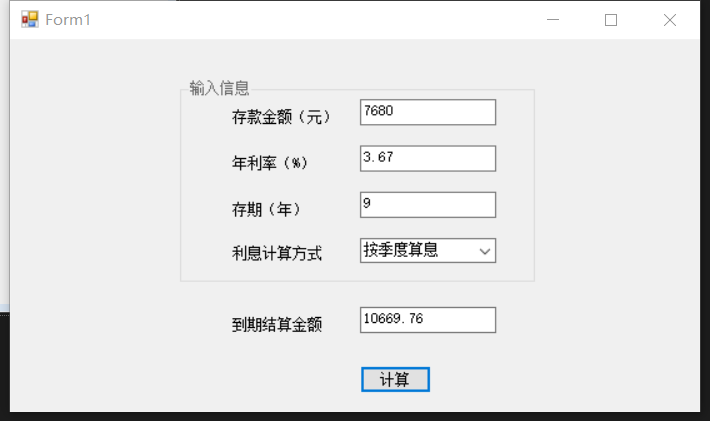


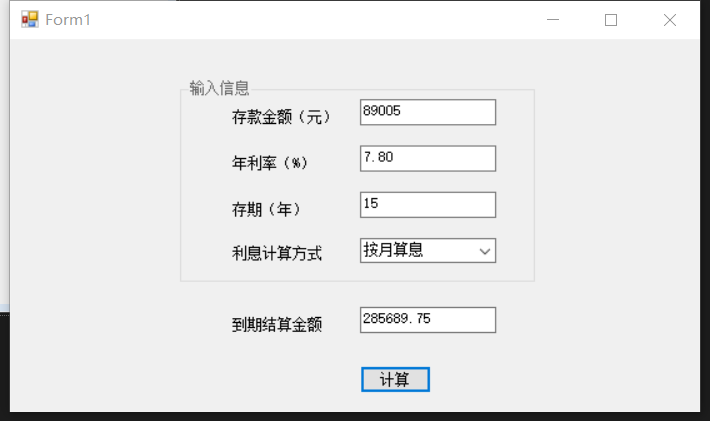












## 主要代码（二级标题）（共三个不同版本代码，可任选其一更改）

版本（1）

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 SuperCalculator

{

public partial class FormMain : Form

{

public FormMain()

{

InitializeComponent();

this.StartPosition = FormStartPosition.CenterScreen;

}

private bool ConvertStringToNumber(string str,bool mustGreatThanZero,out int result)

{

bool isValid = false;

if(int.TryParse(str,out result) == false)

{

MessageBox.Show(string.Format("无法将{0}转换为整数", str));

}

else

{

if(result<=0)

{

MessageBox.Show(string.Format("{0}不是正数", str));

}

else

{

isValid = true;

}

}

return isValid;

}

private bool ConvertStringToNumber(string str, bool mustGreatThanZero, out double result)

{

bool isValid = false;

if (double.TryParse(str, out result) == false)

{

MessageBox.Show(string.Format("无法将{0}转换为实数", str));

}

else

{

if (result <= 0)

{

MessageBox.Show(string.Format("{0}不是正数", str));

}

else

{

isValid = true;

}

}

return isValid;

}

private void label1\_Click(object sender, EventArgs e)

{

}

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

{

}

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

{

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

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

{

int startAmount;

double yearRate;

int years;

if (ConvertStringToNumber(textBoxStartAmount.Text, true, out startAmount) == false) return;

if(startAmount<100)

{

MessageBox.Show("金额不能小于100元");

}

if (ConvertStringToNumber(textBoxYearRate.Text, true, out yearRate) == false) return;

yearRate = yearRate / 100;

if (ConvertStringToNumber(textBoxYears.Text, true, out years) == false) return;

if (comboBoxCalculateFrequency.SelectedIndex == -1)

{

MessageBox.Show("请选择提供的利息计算方式");

return;

}

string calculateFrequency = comboBoxCalculateFrequency.SelectedItem.ToString();

switch(calculateFrequency)

{

case "按月计算":

textBoxTotal.Text = string.Format("{0:F2}元", Caculate(startAmount, yearRate / 12, years \* 12));

break;

case "按季计算":

textBoxTotal.Text = string.Format("{0:F2}元", Caculate(startAmount, yearRate / 4, years \* 4));

break;

case "按年计算":

textBoxTotal.Text = string.Format("{0:F2}元", Caculate(startAmount, yearRate, years));

break;

}

}

private double Caculate(double startAmount,double rate,int count)

{

double total = startAmount;

for(int i=1;i<=count;i++)

{

total += total \* rate;

}

return total;

}

private void textBoxYearRate\_TextChanged(object sender, EventArgs e)

{

}

private void textBoxStartAmount\_TextChanged(object sender, EventArgs e)

{

}

private void textBoxYears\_TextChanged(object sender, EventArgs e)

{

}

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

{

}

private void textBoxTotal\_TextChanged(object sender, EventArgs e)

{

}

private void FormMain\_Shown(object sender, EventArgs e)

{

textBoxStartAmount.Focus();

}

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

{

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

//保证修改任一输入值时，不显示计算结果

textBoxTotal.Clear();

}

}

}

版本（2）

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 ex2

{

public partial class SuperCalculator : Form

{

public SuperCalculator()

{

InitializeComponent();

}

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

{

//存款金额

int startAmount;

//年利率

double yearRate;

//年数

int years;

if (!ConvertStringToNumber(textBoxStartAmount.Text, true, out startAmount))

{

MessageBox.Show("存款金额输入有错！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

if (startAmount < 100)

{

MessageBox.Show("存款金额不能小于100元", "提示", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

if (ConvertStringToNumber(textBoxRate.Text, true, out yearRate) == false)

{

MessageBox.Show("年利率输入有错！", "提示",

MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

yearRate /= 100.0f;

if (ConvertStringToNumber(textBoxYears.Text, true, out years) == false)

{

MessageBox.Show("存期（年）输入有错！", "提示",

MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

if (comboBoxCalculateType.SelectedIndex == -1)

{

MessageBox.Show("请选择提供的利息计算方式", "提示",

MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

switch (comboBoxCalculateType.SelectedItem.ToString())

{

case "按月算息":

labelResult.Text = string.Format("{0:F2}元",

Caculate(startAmount, yearRate / 12, years \* 12));

break;

case "按季度算息":

labelResult.Text = string.Format("{0:F2}元",

Caculate(startAmount, yearRate / 4, years \* 4));

break;

case "按年算息":

labelResult.Text = string.Format("{0:F2}元",

Caculate(startAmount, yearRate, years));

break;

}

}

// 计算到期结算日期

private double Caculate(double startAmount, double rate, int count)

{

double total = startAmount;

for (int i = 1; i <= count; i++)

{

total += total \* rate;

}

return total;

}

//将大于零的字符串转换为 32 位整数或者 64 位浮点数。并指明转换是否成功。

private bool ConvertStringToNumber(string s, bool mustGreatThanZero, out int result)

{

if (int.TryParse(s, out result) == false)

{

return false;

}

else if (mustGreatThanZero && result <= 0)

{

return false;

}

return true;

}

private bool ConvertStringToNumber(string s, bool mustGreatThanZero, out double result)

{

if (double.TryParse(s, out result) == false)

{

return false;

}

if (mustGreatThanZero && result <= 0)

{

return false;

}

return true;

}

}

}

版本（3）

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 SuperCalculator

{

public partial class MainForm : Form

{

public MainForm()

 {

InitializeComponent();

this.StartPosition = FormStartPosition.CenterScreen;

string[] caclType = { "按月计息", "按季度计息", "按年计息" };         comboBoxCalculateType.Items.AddRange(caclType);         comboBoxCalculateType.SelectedIndex = 0;

labelResult.Text = string.Empty;

}

private void MainForm\_Shown(object sender, EventArgs e)

   {

textBoxStartAmount.Focus();

}

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

  {

//存款金额

int startAmount;

 //年利率

float yearRate;

//存期

int years;

if (!ConvertStringToNumber(textBoxStartAmount.Text, true, out startAmount))

     {

MessageBox.Show("存款金额输入有错！", "提示",

                 MessageBoxButtons.OK, MessageBoxIcon.Warning);

 return;

 }

if (startAmount < 100)

  {

   MessageBox.Show("存款金额不能小于100元", "提示",                  MessageBoxButtons.OK, MessageBoxIcon.Warning);

        return;

   }

 if (ConvertStringToNumber(textBoxYearRate.Text, true, out yearRate) == false)

     {

    MessageBox.Show("年利率输入有错！", "提示",                 MessageBoxButtons.OK, MessageBoxIcon.Warning);

        return;

  }

  yearRate /= 100.0f;

  if (ConvertStringToNumber(textBoxYears.Text, true, out years) == false)

   {

   MessageBox.Show("存期（年）输入有错！", "提示",                 MessageBoxButtons.OK, MessageBoxIcon.Warning);

        return;

   }

     if (comboBoxCalculateType.SelectedIndex == -1)

    {

            MessageBox.Show("请选择提供的利息计算方式", "提示",                 MessageBoxButtons.OK, MessageBoxIcon.Warning);

        return;

    }

   labelParameter.Text =

       string.Format("存款金额：{0}元{3}{3}年利率：{1}%{3}{3}存期：{2}年",             startAmount, yearRate \* 100, years, Environment.NewLine);

       switch (comboBoxCalculateType.SelectedItem.ToString())

      {

  case "按月计息":

labelResult.Text = string.Format("{0:F2}元",

Caculate(startAmount, yearRate / 12, years \* 12));

break;

case "按季度计息":

labelResult.Text = string.Format("{0:F2}元",

Caculate(startAmount, yearRate / 4, years \* 4));

break;

 case "按年计息":

labelResult.Text = string.Format("{0:F2}元",

 Caculate(startAmount, yearRate, years));

break;

   }

  }

private void groupBox1\_Enter(object sender, EventArgs e)

  {

//保证修改任一输入值时，不显示结果

 labelParameter.Text = string.Empty;

 labelResult.Text = string.Empty;

    }

 /// <summary>

 /// 将字符串转换为32位整数

/// </summary>

 /// <param name="s">被转换的字符串</param>

   /// <param name="mustGreatThanZero">是否有必须大于零的要求</param>

    /// <param name="result">转换后的结果</param>

   /// <returns></returns>

 private bool ConvertStringToNumber(

string s, bool mustGreatThanZero, out int result)

 {

if (int.TryParse(s, out result) == false)

{

 return false;

 }

 else if (mustGreatThanZero && result <= 0)

 {

 return false;

  }

 return true;

   }

 /// <summary>

 /// 将字符串转换为64位浮点数

 /// </summary>

 /// <param name="s">被转换的字符串</param>

/// <param name="mustGreatThanZero">是否有必须大于零的要求</param>

 /// <param name="result">转换后的结果</param>

    /// <returns></returns>

  private bool ConvertStringToNumber(

     string s, bool mustGreatThanZero, out float result)

   {

 if (float.TryParse(s, out result) == false)

   {

     return false;

    }

  if (mustGreatThanZero && result <= 0)

    {

   return false;

     }

   return true;

   }

 /// <summary>

 /// 计算到期结算金额

  /// </summary>

  /// <param name="startAmount">存款金额</param>

  /// <param name="rate">利率</param>

 /// <param name="count">叠加次数</param>

 /// <returns></returns>

 private float Caculate(float startAmount, float rate, int count)

   {

   float total = startAmount;

    for (int i = 1; i <= count; i++)

    {

  total += total \* rate;

    }

    return total;

  }

}

}

## 问题及解答（二级标题）

### ~~(1). 画出包含输入和输出数据的程序运行界面示例。~~

### (2). 运行设计的程序，在实验报告中列出计算后的表 4-1 的内容，并说明程序计算结果和手工计算结果是否符合。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 初始金额（元） | 利率（%） | 年数 | 计算方式 | 到期结算总额（元） |
| 1000 | 2 | 5 | 按月计息 | 1105.08 |
| 按季度计息 | 1104.90 |
| 按年计息 | 1104.08 |
| 3500 | 3.3 | 7 | 按月计息 | 4408.11 |
| 按季度计息 | 4405.33 |
| 按年计息 | 4393.09 |
| 5000 | 6.25 | 10 | 按月计息 | 9326.09 |
| 按季度计息 | 9296.20 |
| 按年计息 | 9167.68 |

程序计算结果和手工计算结果相同

## 实验小结（二级标题）

写出实验中遇到的问题，解决途径、方法以及跟参考代码思考不一样的地方