

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：

# 实验四 界面设计与文件存取

## 一、实验目的

(1). 练习界面功能的设计方法。

(2). 练习文本文件的存取方法。

(3). 练习对话框的使用方法。

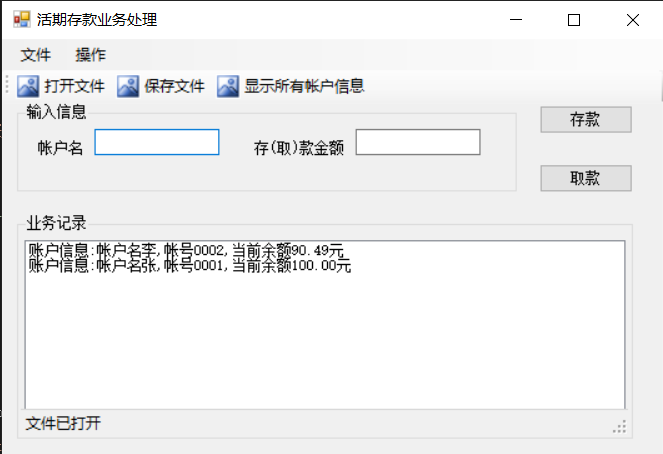
(4). 练习菜单的设计方法。

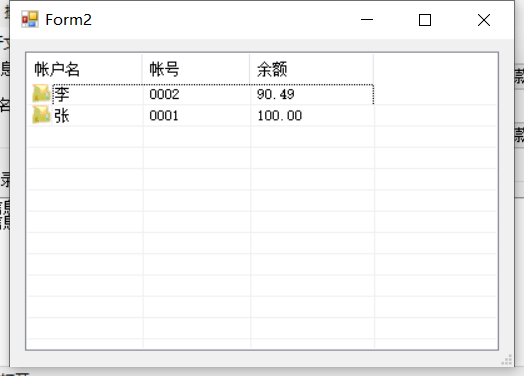
(5). 练习工具条和状态条的设计方法。

(6). 练习一个窗体调用另外一个窗体以及窗体间参数传递的方法。

## 二、实验结果

界面运行截图：





## 三、主要代码

FormMain.cs:

1. **namespace** MenuBankCustoms
2. {
4. **public** partial **class** FormMain : Form
5. {
6. **double** starmoney;
7. Customers cus = **new** Customers();
8. Account custom;
10. SortedList<**string**, Account> Customs = **new** SortedList<**string**, Account>();
11. **public** FormMain()
12. {
13. InitializeComponent();
15. **this**.StartPosition = FormStartPosition.CenterScreen;
16. }

19. ///判断金额格式是否正确
20. **private** **void** textBoxAccountBalance\_Leave(**object** sender, EventArgs e)
21. {
22. **bool** ok = **double**.TryParse(textBoxAccountBalance.Text, **out** starmoney);
23. **if** (!ok)
24. {
25. MessageBox.Show("存（取）款金额格式不正确，请检查后重新输入");
26. }
27. **else**
28. {
29. **if** (starmoney < 0)
30. MessageBox.Show("存（取）款金额不能小于零");
31. }
32. }
34. **private** **void** Form1\_Shown(**object** sender, EventArgs e)
35. {
36. textBoxAccountName.Focus();
37. //设置状态栏显示
38. toolStripStatusLabel1.Text = "文件还没打开，请先打开文件，再操作";
39. }
41. /// 向listbox中添加字符串
42. **private** **void** AddItemTolistBox(Account Custom)
43. {
44. listBoxshow.Items.Add(**string**.Format("账户信息:帐户名{0},帐号{1:d4},当前余额{2:n2}元",custom.AccountName,custom.AccountNumber,custom.AccountBalance));
45. }
47. **private** **void** 打开ToolStripMenuItem\_Click(**object** sender, EventArgs e)
48. {
49. **try**
50. {
51. Account[] b = **new** Account[100];
52. b = cus.Load();
53. **int**[] number = **new** **int**[100];
54. **for** (**int** i = 0; i < Customers.m; i++)
55. {
56. Customs.Add(b[i].AccountName, b[i]);
57. listBoxshow.Items.Add(**string**.Format("账户信息:帐户名{0},帐号{1:d4},当前余额{2:n2}元", b[i].AccountName, b[i].AccountNumber, b[i].AccountBalance));
58. number[i] = b[i].AccountNumber;
59. }
61. Account.newAccountNumber = number.Max() + 1;
62. toolStripStatusLabel1.Text = "文件已打开";
63. }
64. **catch** (Exception ex)
65. {
66. MessageBox.Show(ex.Message);
67. }
68. }
70. /// 文件保存。
71. **private** **void** 保存ToolStripMenuItem\_Click(**object** sender, EventArgs e)
72. {
73. **if** (toolStripStatusLabel1.Text == "文件还没打开，请先打开文件，再操作")
74. MessageBox.Show("文件还没打开", "操作失败", MessageBoxButtons.OK, MessageBoxIcon.Warning);
75. **else**
76. {
77. **if** (MessageBox.Show("是否将所有信息保存到文件中？", "提示", MessageBoxButtons.YesNo) == DialogResult.Yes)
78. {
79. Account[] acc = **new** Account[Customs.Count];
80. **for** (**int** i = 0; i < Customs.Count; i++)
81. acc[i] = Customs.Values[i];
82. cus.Save(acc);
83. toolStripStatusLabel1.Text = "文件已保存";
84. MessageBox.Show("保存文件成功！");
85. }
87. }
88. }
90. **private** **void** toolStripButton1\_Click(**object** sender, EventArgs e)
91. {
92. 打开ToolStripMenuItem.PerformClick();
93. }
95. **private** **void** toolStripButton2\_Click(**object** sender, EventArgs e)
96. {
97. 保存ToolStripMenuItem.PerformClick();
98. }
100. **private** **void** toolStripButton3\_Click(**object** sender, EventArgs e)
101. {
102. 显示所有帐户信息ToolStripMenuItem.PerformClick();
103. }
105. /// 取款按钮处理事件
106. **private** **void** buttonWithdraw\_Click(**object** sender, EventArgs e)
107. {
108. **if** (toolStripStatusLabel1.Text == "文件还没打开，请先打开文件，再操作")
109. MessageBox.Show("文件还没打开", "操作失败", MessageBoxButtons.OK, MessageBoxIcon.Warning);
110. **else**
111. {
112. **if** (Customs.ContainsKey(textBoxAccountName.Text) == **false**)
113. {
114. MessageBox.Show("无此账户，无法取款", "操作失败", MessageBoxButtons.OK, MessageBoxIcon.Question);
115. }
116. **else**
117. {
118. custom = Customs[textBoxAccountName.Text];
119. custom.Withdraw(starmoney);
120. AddItemTolistBox(custom);
121. }
122. }
123. }
125. /// 存款按钮处理事件
126. **private** **void** buttonDesposit\_Click(**object** sender, EventArgs e)
127. {
128. **if** (toolStripStatusLabel1.Text == "文件还没打开，请先打开文件，再操作")
129. MessageBox.Show("文件还没打开", "操作失败", MessageBoxButtons.OK, MessageBoxIcon.Warning);
130. **else**
131. {
132. **if** (Customs.ContainsKey(textBoxAccountName.Text) == **false**)
133. {
134. custom = **new** Account(textBoxAccountName.Text, starmoney);
135. Customs.Add(textBoxAccountName.Text, custom);
136. AddItemTolistBox(custom);
138. }
139. **else**
140. {
141. custom = Customs[textBoxAccountName.Text];
142. custom.Deposit(starmoney);
143. AddItemTolistBox(custom);
144. }
145. }
146. }
148. /// 显示所有用户按钮处理事件
149. **private** **void** 显示所有帐户信息ToolStripMenuItem\_Click(**object** sender, EventArgs e)
150. {
151. **if** (toolStripStatusLabel1.Text == "文件还没打开，请先打开文件，再操作")
152. MessageBox.Show("文件还没打开", "操作失败", MessageBoxButtons.OK, MessageBoxIcon.Warning);
153. **else**
154. {
155. **try**
156. {
157. Account[] b = **new** Account[100];
158. b = cus.Load();
159. **string**[] str = **new** **string**[Customers.m];
161. **for** (**int** i = 0; i < Customers.m; i++)
162. {
163. str[i] = **string**.Format("{0},{1:d4},{2:f2}", b[i].AccountName, b[i].AccountNumber, b[i].AccountBalance);
164. }
165. Form2 fm2 = **new** Form2(str);
166. fm2.ShowDialog();
167. }
168. **catch** (Exception ex)
169. {
170. MessageBox.Show(ex.Message);
171. }
172. }
173. }
175. **private** **void** 存款ToolStripMenuItem\_Click(**object** sender, EventArgs e)
176. {
177. buttonDesposit.PerformClick();
178. }
180. **private** **void** 取款ToolStripMenuItem\_Click(**object** sender, EventArgs e)
181. {
182. buttonWithdraw.PerformClick();
183. }
185. **private** **void** listBoxshow\_SelectedIndexChanged(**object** sender, EventArgs e)
186. {
188. }
189. }
190. }

Account.cs:

1. **namespace** MenuBankCustoms
2. {
3. **class** Account
4. {
5. **public** **static** **int** newAccountNumber = 0001;
6. **private** **string** accountName;//账户名
7. **public** **string** AccountName
8. {
9. **get** { **return** accountName; }
10. **set** { accountName = value; }
11. }
12. **private** **int** accountNumber;//账户号
13. **public** **int** AccountNumber
14. {
15. **get** { **return** accountNumber; }
16. **set** { accountNumber = value; }
17. }
18. **private** **double** accountBalance;//账户余额
20. **public** **double** AccountBalance
21. {
22. **get** { **return** accountBalance; }
23. **set** { accountBalance = value; }
24. }
25. **public** **float** YearRate
26. {
27. **get** { **return** 0.005f; }
28. }
29. **public** Account(**string** name, **double** money)
30. {
31. accountName = name;
32. accountBalance = money;
33. accountNumber = newAccountNumber;
34. newAccountNumber++;
35. }
36. **public** Account()
37. {
38. }
40. **public** **void** Deposit(**double** money)
41. {
42. accountBalance += money;
43. }
45. **public** **virtual** **void** Withdraw(**double** money)
46. {
47. **if** (money > accountBalance)
48. **throw** **new** Exception("取款金额不能大于存款金额，请检查是否输入错误");
49. **else**
50. {
51. accountBalance += accountBalance \* YearRate;
52. accountBalance -= money;
53. }
54. }
55. }
56. }

Customers.cs

1. **namespace** MenuBankCustoms
2. {
3. **class** Customers
4. {
5. **public** **static** **int** m;
6. Account[] a = **new** Account[100];
7. **string** path = @"C:\Users\xiaoshu12138\Desktop\account.txt";
8. **public** Account[] Load()
9. {
10. m = 0;
11. **string** line;
12. //检查文件是否存在，存在打开，不存在就创建空文件。
13. **if** (File.Exists(path))
14. {
15. StreamReader sr = **new** StreamReader(path);
16. **while** ((line = sr.ReadLine()) != **null**)
17. {
18. **string**[] ln = line.Split(' ');
19. a[m] = **new** Account();
20. a[m].AccountName = Convert.ToString(ln[0]);
21. a[m].AccountNumber = Convert.ToInt32(ln[1]);
22. a[m].AccountBalance = Convert.ToDouble(ln[2]);
23. m++;
24. }
25. sr.Close();
26. }
27. **else**
28. {
29. File.AppendAllText(path, "", Encoding.Default);
30. }
31. **return** a;
32. }
34. **public** **void** Save(Account[] cust)
35. {
36. **if** (File.Exists(path))
37. File.Delete(path);
38. **for** (**int** n = 0; n < cust.Length; n++)
39. {
40. **string** str=**string**.Format("{0} {1} {2} {3}",cust[n].AccountName,cust[n].AccountNumber,cust[n].AccountBalance,Environment.NewLine);
41. File.AppendAllText(path, str, Encoding.Default);
42. }
43. }
44. }
45. }

Form2.cs

1. **namespace** MenuBankCustoms
2. {
3. **public** partial **class** Form2 : Form
4. {
5. **public** Form2(**string**[] str)
6. {
7. InitializeComponent();
8. **this**.StartPosition = FormStartPosition.CenterScreen;
9. listView1.GridLines = **true**;
10. listView1.FullRowSelect = **true**;
11. //listView1.View = View.Details;
12. // listView1.Columns.Add("帐户名",80,HorizontalAlignment.Left);
13. //listView1.Columns.Add("帐号", 100, HorizontalAlignment.Left);
14. // listView1.Columns.Add("剩余金额", 80, HorizontalAlignment.Left);
15. **for** (**int** i = 0; i < str.Length; i++)
16. {
17. **string**[] s = str[i].Split(',');
18. AddlistviewItem(s[0],s[1],s[2]);
19. }
20. }
21. **private** **void** AddlistviewItem(**string** s1,**string** s2,**string** s3)
22. {
23. **int** imageIndex = 0;
24. ListViewItem item = **new** ListViewItem(
25. **new** **string**[] {s1,s2,s3}, imageIndex);
26. listView1.Items.Add(item);
27. listView1.View = View.Details;
28. }
29. }
30. }

## 四、问题及解答

(1). 写出你认为有必要解释的关键步骤或代码。

见上述代码。

(2). 写出实验中遇到的问题及解决方法。

(3). 如果通过Internet 连网，即可以在任一地点存款和取款，简要回答需要考虑哪些问题？

应该考虑系统的安全性，同时应该记录地点时间等信息。

## 五、实验小结

通过此次实验基本掌握了界面设计和文件存储，理解了一个窗体调用另外一个窗体以及窗体间参数传递的方法，后续还需要练习。