```csharp
 1  using System;
 2  using System.Collections.Generic;
 3  using System.ComponentModel;
 4  using System.Data;
 5  using System.Drawing;
 6  using System.IO;
 7  using System.Linq;
 8  using System.Security.Policy;
 9  using System.Text;
10  using System.Threading.Tasks;
11  using System.Windows.Forms;
12
13
14
15  /*
16  _incomes, expenditures 딕셔너리에 접근할 방법이 없어서 restore메소드에서 해당 딕
       셔너리에
17  정보를 저장하질 않았음.
18  파일에 저장된 정보를 "복구하기" 버튼을 눌렀을 때 가져오는 식으로 구성했음.
19  년도별 가계부를 currentFL(FinancialLedger)에 올려두면서 해당년도 가계부만 핸들링하
       게 했음.
20  FLs딕셔너리는 각 년도별 FL들의 인스턴스를 담아놓는 딕셔너리임.. 그런데 딱히
       쓸모는 없음. incomes와 expenditures에 접근이 안되니까..
21
22
23   */
24
25  namespace Proj_FinancialLedger_202001098
26  {
27      public partial class Form1: Form
28      {
29          private const string income = "i";
30          private const string expenditure = "o";
31          private const int fileFormatCnt = 4;
32          private const string _yearFormat = "yyyyMMdd hhmmssfff";
33          private const string BASE_FOLDER_DIR = "..\\..\\financialLedgers\\";
34          FinancialLedger currentFL;
35          Dictionary<int, FinancialLedger> FLs;
36          public Form1()
37          {
38              InitializeComponent();
39              currentFL = null;
40              FLs = new Dictionary<int, FinancialLedger>();
41
42              restore();
43          }
44          private void restore()
45          {
46              makeDirectory();
47              string[] files = Directory.GetFiles(BASE_FOLDER_DIR).Select(p =>
                   Path.GetFileName(p)).ToArray();
48              //Console.WriteLine("불러온 파일의 명칭들 " + files[0]);
49              if (files.Length != 0)
```

```csharp
 50                      {
 51                          foreach (string file in files)
 52                          {
 53                              try
 54                              {
 55                                  using (StreamReader sr = new StreamReader(returnPath    ⮑
                                       (file)))
 56                                  {
 57                                      int year = int.Parse(file.Split(new char[] { '.' })   ⮑
                                           [0]);
 58
 59                                      int targetAmount = int.Parse(sr.ReadLine());
 60                                      FLs[year] = new FinancialLedger(year, targetAmount); ;
 61                                  }
 62                              } catch (Exception ex) { Console.WriteLine(ex); }
 63                          }
 64                      }
 65                  }
 66
 67          private string returnPath(string filename)
 68          {
 69              string path = null;
 70              makeDirectory();
 71              if (filename.Contains(".txt"))
 72              {
 73                  path = Path.Combine(BASE_FOLDER_DIR, filename);
 74              }else
 75              {
 76                  path = Path.Combine(BASE_FOLDER_DIR, filename + ".txt");
 77              }
 78              return path;
 79          }
 80          private void makeDirectory()
 81          {
 82              if (false == Directory.Exists(BASE_FOLDER_DIR))
 83              {
 84                  Directory.CreateDirectory(BASE_FOLDER_DIR);
 85              }
 86          }
 87          private bool InvalidInputCheck(TextBox tbx)
 88          {
 89              if (string.IsNullOrWhiteSpace(tbx.Text))
 90              {
 91                  tbx.Focus();
 92                  return true;
 93              }
 94              return false;
 95          }
 96
 97          private bool InvalidYearCheck(int year)
 98          {
 99              if (year > DateTime.Today.Year + 30 || year < 1900)
100              {
```

```csharp
101                 return true;
102             }
103             return false;
104         }
105         private bool InvalidMonthCheck(int month)
106         {
107             if (month > 12 || month < 1)
108             {
109                 return true;
110             }
111             return false;
112         }
113         private bool InvalidDayCheck(int day)
114         {
115             if (day > 31 || day < 1)
116             {
117                 return true;
118             }
119
120             //public bool IsLeapYear(int year)
121             //{
122             //return (year % 4 == 0) && ((year % 100 != 0) || (year % 400 == 0));
123             //}
124             //if (month == 2 )
125             //{
126             //    if (IsLeapYear(year))
127             //    {
128             //        if (day > 29) { MessageBox.Show("윤년이면서 2월달은 29일까지 ↵
             입니다."); return false; }
129             //    }else
130             //    {
131             //        if (day > 28) { MessageBox.Show("2월달은 28일까지입니다."); ↵
             return false; }
132             //    }
133             //}
134             return false;
135         }
136
137         private void BtnReset_Click(object sender, EventArgs e)
138         {
139             MakeDefault();
140         }
141
142         private void MakeDefault()
143         {
144             int thisYear = DateTime.Today.Year;
145             int thisMonth = DateTime.Today.Month;
146             int thisDay = DateTime.Today.Day;
147             tbxYear.Text = thisYear.ToString();
148             tbxTargetAmount.Text = "0";
149
150             lblTotalIncome.Text = "0원";
151             lblTotalExpenditure.Text = "0원";
```

```csharp
152              lblTotalAmount.Text = "0원";
153              lblIsBlack.Text = " ";
154
155              tbxIncomeYear.Text = thisYear.ToString();
156              tbxIncomeMonth.Text = thisMonth.ToString();
157              tbxIncomeDay.Text = thisDay.ToString();
158              tbxIncomeContent.Text = string.Empty;
159              tbxIncomeMoney.Text = "0";
160
161              tbxExpenditureYear.Text = thisYear.ToString();
162              tbxExpenditureMonth.Text = thisMonth.ToString();
163              tbxExpenditureDay.Text = thisDay.ToString();
164              tbxExpenditureContent.Text = string.Empty;
165              tbxExpenditureMoney.Text = "0";
166
167              tbxFastContent.Text = string.Empty;
168              tbxFastMoney.Text = "0";
169
170              BlockingInput(true);
171
172
173              lbxIncome.Items.Clear();
174              lbxExpenditure.Items.Clear();
175              lbxAll.Items.Clear();
176          }
177
178          private void btnStore_Click(object sender, EventArgs e)
179          {
180
181
182              if (InvalidInputCheck(tbxYear) || InvalidInputCheck(tbxTargetAmount)) ⮡
                   { return; }
183
184
185              int year, targetAmount;
186              if (false == int.TryParse(tbxYear.Text, out year)) { tbxYear.Focus(); ⮡
                    return; }
187              if (false == int.TryParse(tbxTargetAmount.Text, out targetAmount)) ⮡
                   { tbxTargetAmount.Focus(); return; }
188
189              if (InvalidYearCheck(year)) { return; }
190
191              FinancialLedger fl = new FinancialLedger(year, targetAmount);
192              currentFL = fl;
193
194              //파일이 없다면 만든다
195              if (false == File.Exists(returnPath(year.ToString())))
196              {
197                  try {
198                      using (FileStream fs = new FileStream(returnPath(year.ToString ⮡
                        ()), FileMode.Create)) {
199                          using (StreamWriter sw = new StreamWriter(fs)) {
200                              sw.WriteLine(targetAmount);
```

```csharp
201                                    }
202                                }
203                            } catch (Exception ex ) { Console.WriteLine(ex); }
204                        } else
205                        {
206                            string[] lines = File.ReadAllLines(returnPath(year.ToString()));
207
208                            if (lines.Length > 0)
209                            {
210                                lines[0] = targetAmount.ToString(); // 첫 줄만 수정
211                            }
212                            File.WriteAllLines(returnPath(year.ToString()), lines);
213                        }
214
215
216                        MakeDefault();
217                        BlockingInput(year, false);
218                    }
219
220                    private void BlockingInput(int year, bool b)
221                    {
222                        tbxYear.Enabled = b;
223                        tbxTargetAmount.Enabled = b;
224                        tbxIncomeYear.Text = year.ToString();
225                        tbxIncomeYear.Enabled = b;
226                        tbxExpenditureYear.Text = year.ToString();
227                        tbxExpenditureYear.Enabled = b;
228                        if (DateTime.Today.Year == year)
229                        {
230                            tbxFastContent.Enabled = !b;
231                            tbxFastMoney.Enabled = !b;
232                        }
233                        else
234                        {
235                            tbxFastContent.Enabled = b;
236                            tbxFastMoney.Enabled = b;
237                        }
238                    }
239
240                    private void BlockingInput(bool b)
241                    {
242                        tbxYear.Enabled = b;
243                        tbxTargetAmount.Enabled = b;
244                        tbxIncomeYear.Enabled = b;
245                        tbxExpenditureYear.Enabled = b;
246                        tbxFastContent.Enabled = b;
247                        tbxFastMoney.Enabled = b;
248                    }
249
250
251                    private void btnExit_Click(object sender, EventArgs e)
252                    {
253                        Application.Exit();
```

```
254            }
255
256        private void btnRegIncome_Click(object sender, EventArgs e)
257        {
258            if (currentFL == null)
259            {
260                MessageBox.Show("가계부를 저장하거나 불러와야 합니다.");
261                return;
262            }
263
264            if (InvalidInputCheck(tbxIncomeYear))    { return; }
265            if (InvalidInputCheck(tbxIncomeMonth))   { return; }
266            if (InvalidInputCheck(tbxIncomeDay))     { return; }
267            if (InvalidInputCheck(tbxIncomeContent)) { return; }
268            if (InvalidInputCheck(tbxIncomeMoney))   { return; }
269
270            string content = tbxIncomeContent.Text;
271            string displayStr = null;
272            int year = int.Parse(tbxIncomeYear.Text); //Year는 유저가 수정할 수 없
                   으니 반드시 올바른값임.
273            int month, day, money;
274            if (false == int.TryParse(tbxIncomeMonth.Text, out month))
                 { tbxIncomeMonth.Focus(); return; }
275            if (false == int.TryParse(tbxIncomeDay.Text, out day))
                 { tbxIncomeDay.Focus();   return; }
276            if (false == int.TryParse(tbxIncomeMoney.Text, out money))
                 { tbxIncomeMoney.Focus(); return; }
277
278            if (InvalidMonthCheck(month)) { return; }
279            if (InvalidDayCheck(day))     { return; }
280
281            DateTime dt = new DateTime(year, month, day);
282            TimeSpan ts = DateTime.Now - DateTime.Today;
283            dt = dt.Add(ts);
284
285            if (currentFL != null)
286            {
287                currentFL.RegIncome(dt, content, money);
288            }
289
290            try
291            {
292                using (FileStream fs = new FileStream(returnPath(year.
                    ToString()), FileMode.Append))
293                {
294                    using (StreamWriter sw = new StreamWriter(fs))
295                    {
296                        displayStr = $"{income},{dt.ToString(_yearFormat)},
                            {content},{money}";
297                        sw.WriteLine(displayStr);
298                    }
299                }
300            }
```

```
301                    catch (Exception ex) { Console.WriteLine(ex); }
302
303                    DisplayInfo(displayStr);
304                }
305
306            private void DisplayInfo(string datas)
307            {
308                if (currentFL == null) return;
309                int amount = currentFL.TotalIncome - currentFL.TotalExpenditure;
310                lblTotalIncome.Text = currentFL.TotalIncome.ToString("N0") + "원";
311                lblTotalExpenditure.Text = currentFL.TotalExpenditure.ToString("
                      N0") + "원";
312                lblTotalAmount.Text = amount.ToString("N0") + "원";
313                lblIsBlack.Text = currentFL.IsBlack ? "흑자" : "적자";
314
315                string[] temp = datas.Split(new char[] { ',' });
316                string date = temp[1].Substring(0, 8);
317                int money = int.Parse(temp[3]);
318                if (temp[0] == income)
319                {
320                    lbxIncome.Items.Add($"[수입]({date}) {temp[2]} : {money.ToString
                          ("N0")}원");
321                    lbxAll.Items.Add($"[수입]({date}) {temp[2]} : {money.ToString
                          ("N0")}원");
322                }else
323                {
324                    lbxExpenditure.Items.Add($"[지출]({date}) {temp[2]} :
                          {money.ToString("N0")}원");
325                    lbxAll.Items.Add($"[지출]({date}) {temp[2]} : {money.ToString
                          ("N0")}원");
326                }
327
328            }
329
330            private void DisplayInfo()
331            {
332                if (currentFL == null) return;
333                int amount = currentFL.TotalIncome - currentFL.TotalExpenditure;
334                lblTotalIncome.Text = currentFL.TotalIncome.ToString("N0") + "원";
335                lblTotalExpenditure.Text = currentFL.TotalExpenditure.ToString("
                      N0") + "원";
336                lblTotalAmount.Text = amount.ToString("N0") + "원";
337                lblIsBlack.Text = currentFL.IsBlack ? "흑자" : "적자";
338
339            }
340
341            private void btnRegExpenditure_Click(object sender, EventArgs e)
342            {
343                if (currentFL == null)
344                {
345                    MessageBox.Show("가계부를 저장하거나 불러와야 합니다.");
346                    return;
347                }
```

```
348
349            if (InvalidInputCheck(tbxExpenditureYear)) { return; }
350            if (InvalidInputCheck(tbxExpenditureMonth)) { return; }
351            if (InvalidInputCheck(tbxExpenditureDay)) { return; }
352            if (InvalidInputCheck(tbxExpenditureContent)) { return; }
353            if (InvalidInputCheck(tbxExpenditureMoney)) { return; }
354
355            string content = tbxExpenditureContent.Text;
356            string displayStr = null;
357            int year = int.Parse(tbxExpenditureYear.Text); //Year는 유저가 수정할
                 수 없으니 반드시 올바른값임.
358            int month, day, money;
359            if (false == int.TryParse(tbxExpenditureMonth.Text, out month))
                 { tbxExpenditureMonth.Focus(); return; }
360            if (false == int.TryParse(tbxExpenditureDay.Text, out day))
                 { tbxExpenditureDay.Focus(); return; }
361            if (false == int.TryParse(tbxExpenditureMoney.Text, out money))
                 { tbxExpenditureMoney.Focus(); return; }
362
363            if (InvalidMonthCheck(month)) { return; }
364            if (InvalidDayCheck(day)) { return; }
365
366            DateTime dt = new DateTime(year, month, day);
367            TimeSpan ts = DateTime.Now - DateTime.Today;
368            dt = dt.Add(ts);
369
370            if (currentFL != null)
371            {
372                currentFL.RegExpenditure(dt, content, money);
373            }
374
375
376            try
377            {
378                using (FileStream fs = new FileStream(returnPath(year.
                     ToString()), FileMode.Append))
379                {
380                    using (StreamWriter sw = new StreamWriter(fs))
381                    {
382                        displayStr = $"{expenditure},{dt.ToString(_yearFormat)},
                         {content},{money}";
383                        sw.WriteLine(displayStr);
384                    }
385                }
386            }
387            catch (Exception ex) { Console.WriteLine(ex); }
388
389            DisplayInfo(displayStr);
390        }
391
392        private void btnFastRegIncome_Click(object sender, EventArgs e)
393        {
394            if (currentFL == null)
```

```csharp
395                    {
396                        MessageBox.Show("가계부를 저장하거나 불러와야 합니다.");
397                        return;
398                    }
399
400                if (InvalidInputCheck(tbxFastContent)) { return; }
401                if (InvalidInputCheck(tbxFastMoney)) { return; }
402
403                DateTime now = DateTime.Now;
404                string content = tbxFastContent.Text;
405                string displayStr = null;
406                int money;
407                if (false == int.TryParse(tbxFastMoney.Text, out money))          ⮑
                     { tbxFastMoney.Focus(); return; }
408
409                if (currentFL != null)
410                {
411                    currentFL.RegIncome(content, money);
412                }
413
414
415                try
416                {
417                    using (FileStream fs = new FileStream(returnPath(now.Year.ToString ⮑
                       ()), FileMode.Append))
418                    {
419                        using (StreamWriter sw = new StreamWriter(fs))
420                        {
421                            displayStr = $"{income},{now.ToString(_yearFormat)},        ⮑
                              {content},{money}";
422                            sw.WriteLine(displayStr);
423                        }
424                    }
425                }
426                catch (Exception ex) { Console.WriteLine(ex); }
427
428                DisplayInfo(displayStr);
429            }
430
431            private void btnFastRegExpenditure_Click(object sender, EventArgs e)
432            {
433
434                if (currentFL == null)
435                {
436                    MessageBox.Show("가계부를 저장하거나 불러와야 합니다.");
437                    return;
438                }
439                if (InvalidInputCheck(tbxFastContent)) { return; }
440                if (InvalidInputCheck(tbxFastMoney)) { return; }
441
442                DateTime now = DateTime.Now;
443                string displayStr = null;
444                string content = tbxFastContent.Text;
```

```
445                int money;
446                if (false == int.TryParse(tbxFastMoney.Text, out money))                    ⮑
                     { tbxFastMoney.Focus(); return; }
447
448                if (currentFL != null)
449                {
450                    currentFL.RegExpenditure(content, money);
451                }
452
453                try
454                {
455                    using (FileStream fs = new FileStream(returnPath(now.Year.ToString ⮑
                         ()), FileMode.Append))
456                    {
457                        using (StreamWriter sw = new StreamWriter(fs))
458                        {
459                            displayStr = $"{expenditure},{now.ToString(_yearFormat)},     ⮑
                               {content},{money}";
460                            sw.WriteLine(displayStr);
461                        }
462                    }
463                }
464                catch (Exception ex) { Console.WriteLine(ex); }
465
466                DisplayInfo(displayStr);
467            }
468
469        private void btnRestore_Click(object sender, EventArgs e)
470        {
471
472
473            if (InvalidInputCheck(tbxYear)) { return; }
474            int year;
475            if (false == int.TryParse(tbxYear.Text, out year)) { tbxYear.Focus();        ⮑
                 return; }
476
477            if (InvalidYearCheck(year)) { return; }
478
479
480            if (FLs.ContainsKey(year))
481                currentFL = FLs[year];
482            else
483            {
484                MessageBox.Show("해당 연도의 가계부가 존재하지 않습니다.");
485                return;
486            }
487
488            MakeDefault();
489            string[] datas = File.ReadAllLines(returnPath(year.ToString()));
490            if (datas.Length > 1)
491            {
492                for (int i = 1; i < datas.Length; i++)
493                {
```

```
494                    string[] temp = datas[i].Trim().Split(new char[] { ',' });
495                    if (temp.Length != fileFormatCnt) { return; }
496
497                    string content = null;
498                    int money = 0;
499                    if (temp[0] == income)
500                    {
501                        // 20240908 = 8자리, 시간까지는 표시하지 않겠다.
502                        string date = temp[1].Substring(0, 8);
503                        content = temp[2];
504                        money = int.Parse(temp[3]);
505
506                        lbxIncome.Items.Add($"[수입]({date}) {content} :          ↵
                             {money.ToString("N0")}원");
507                        lbxAll.Items.Add($"[수입]({date}) {content} :             ↵
                             {money.ToString("N0")}원");
508
509                        DateTime dt = DateTime.ParseExact(temp[1], _yearFormat,   ↵
                             null);
510                        currentFL.RegIncome(dt, content, money);
511                    }
512                    else //expenditure
513                    {
514                        string date = temp[1].Substring(0, 8);
515                        content = temp[2];
516                        money = int.Parse(temp[3]);
517
518                        lbxExpenditure.Items.Add($"[지출]({date}) {content} :      ↵
                             {money.ToString("N0")}원");
519                        lbxAll.Items.Add($"[지출]({date}) {content} :             ↵
                             {money.ToString("N0")}원");
520
521                        DateTime dt = DateTime.ParseExact(temp[1], _yearFormat,   ↵
                             null);
522                        currentFL.RegExpenditure(dt, content, money);
523                    }
524                }
525
526            }
527            BlockingInput(year, false);
528            DisplayInfo();
529        }
530
531
532    }
533 }
534
```