

第 3 章補足 . ファイルを用いたデータ入出力の補足と応用

【学習のねらい】

- 1 行に複数のデータがある場合のデータ読み込みの方法を学習する。
- ファイル入出力を用いた応用プログラムを学習する。

先週は、Windows の Update 作業等で時間を取られ、第 3 章の基礎課題を終了できなかった人が相当数いました。その進捗状況に配慮して、今週は新しい内容（第 4 章）には進まずに、ファイルからのデータ入力の補足事項を学習することにしました。基礎課題は 1 題だけです。各自確実にこれまでの基礎課題を終了させて下さい。基礎課題が終了した学生は、3-6 節の応用課題にチャレンジして下さい。これまで学習した範囲内でも、工夫次第で色々な場面に応用できる、ということが理解できると思います。

3 - 5 1 行に複数のデータがある場合のデータ読み込み

3-4 節で、ファイルからデータを読み込む際の一般的な方法を学習しました。しかし、ここでは、1 行に一つのデータが書き込まれていることが想定されていた。ところが実際の場面では、ファイルの 1 行に複数のデータが書き込まれている場合が少なくありません。その際は、個々のデータを区切って読み込む事が必要になります。本節では、このようにファイルの 1 行に複数のデータが存在する場合の読み込み方を学習します。

【基礎課題 3-7】

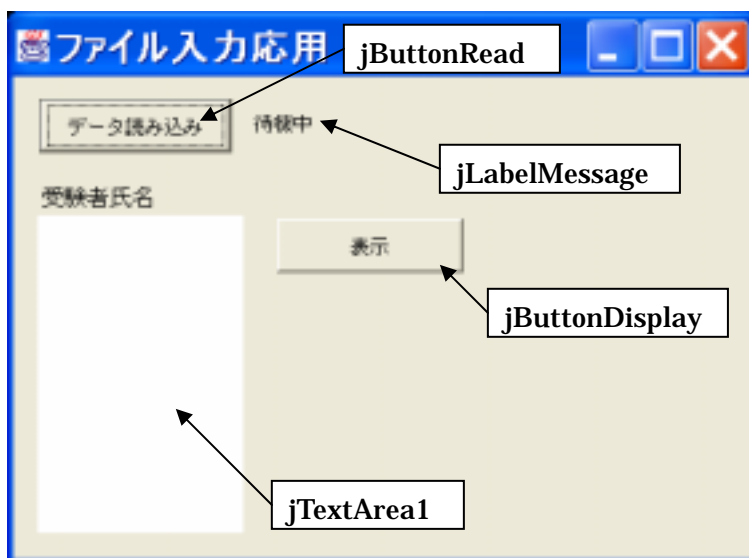
HP の該当部分から、ファイル「score2.txt」をダウンロードして下さい。このファイルには次のように、あるテストの結果が、氏名と得点という形で保管されています。そして、氏名と得点はカンマ記号「,」で区切られています。

そこで、このファイルから、氏名と得点を（別々に）読み込み、その氏名だけを表示するプログラムを作成しましょう。（基本的な部分は【基礎課題 3-5】のプログラムと同様です。異なるのは、「氏名」と「得点」という 2 つのデータを読み込むという点です。）

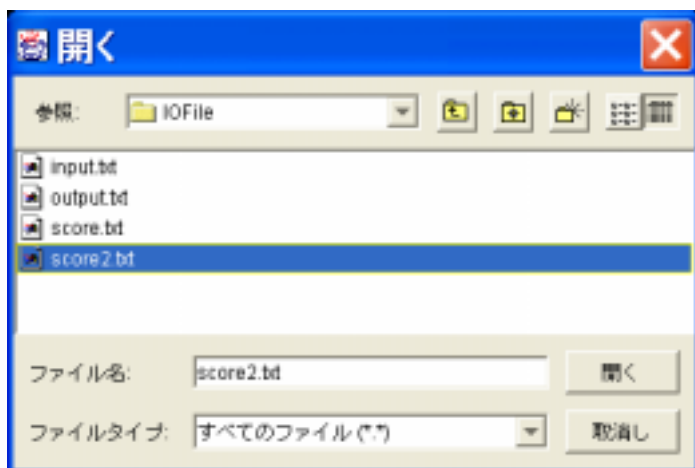
作成するプログラムの動作結果は次ページの通りです。

< score2.txt >

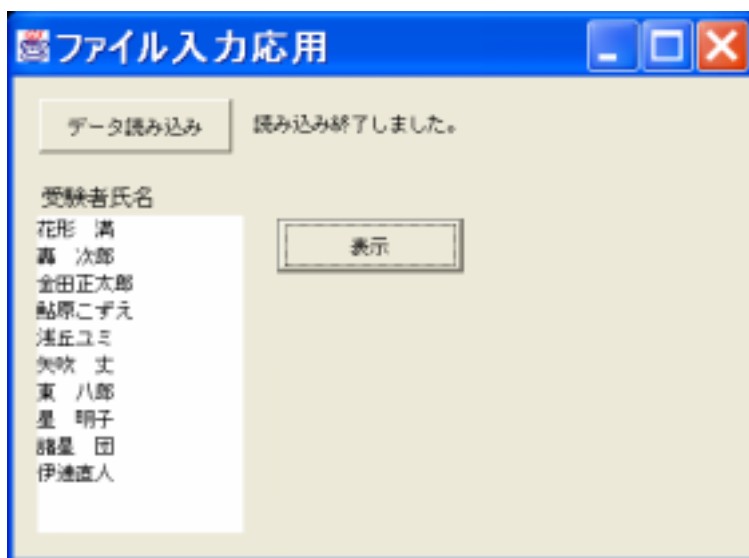
花形 満	,55
轟 次郎	,63
金田正太郎	,39
鮎原こずえ	,87
浅丘ユミ	,48
矢吹 丈	,70
東 八郎	,35
星 明子	,77
諸星 団	,59
伊達直人	,44



プログラムを起動すると、次の画面が現れます。
ここで、[データ読み込み] ボタンをクリックすると・・・



入力ファイルを指定するためのダイアログボックスが現れます。
ここで、ダウンロードしたファイル「score2.txt」を指定し、[開く] ボタンをクリックします。



すると、データの読み込みが完了し「読み込み終了しました。」というメッセージが表示されます。

続いて [表示] ボタンをクリックすると、テキストエリアに、読み込んだ受験者の全員の氏名が表示されます。

なお、フレームの設計に当たっては、3-2 節で学習した通り、JFileChooser コンポーネントを構造ペインの UI フォルダに追加しておく事を忘れないようにして下さい。

それでは、プログラムの作成に取りかかりましょう。まず、プログラムの冒頭部に以下の 2 行（波線部）を追加して下さい。

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;
import java.util.*;
```

「java.io.*」が必要なのはこれまで通りですが、「java.util.*」が必要なのは、この中に、複数のデータを区切って読み込む機能を有するクラスが入っている（そしてここでそれを利用する）からです。

[データ読み込み] ボタンのイベントハンドラ、および、氏名と得点を保管する配列、そしてデータの個数を保管する変数の宣言は次の通りです。

```
String Name[]=new String[100]; //氏名を保管する配列
int Tokuten[]=new int[100]; //得点を保管する配列
int Num; //データの配列
```

```
void jButtonRead_actionPerformed(ActionEvent e) {
    String Data;
    try {
        jFileChooser1.showOpenDialog(this);
        File FName=jFileChooser1.getSelectedFile();
        BufferedReader fin=new BufferedReader
            (new FileReader(FName));

        int i=0;
        while ((Data=fin.readLine())!=null) {
            StringTokenizer tk=new StringTokenizer(Data, ",");
            Name[i]=tk.nextToken();
            Tokuten[i]=Integer.parseInt(tk.nextToken());
            i++;
        }
        Num=i;
        jLabelMessage.setText("読み込み終了しました。");
        fin.close();
    }
    catch (Exception em) {
        jLabelMessage.setText("エラー発生："+em);
    }
}
```

枠線部以外は、基本的に【基礎課題 3-5】と同様なので説明は省きます。ここでは、新しく出てきた「StringTokenizer」クラスについて説明すれば十分でしょう。

< StringTokenizer クラスについて >

```
StringTokenizer tk=new StringTokenizer(Data,",");
```

によって、StringTokenizer クラスのオブジェクト（変数と考えて結構です）を「tk」という名前で生成しています。

一般に、「new StringTokenizer(文字列,区切り記号)」という形で生成したオブジェクトには、「文字列」の中身が「区切り記号」で分解され、各々が順に（文字列として）保管されます。上の例で具体的に説明しましょう。ファイルの最初の 1 行が変数 Data に読み込まれとき、Data の中身は次のようになっています。

変数 Data の中身
花形 満 ,55

そして、「StringTokenizer tk=new StringTokenizer(Data,",");」によって、Data の中身は区切り記号「,」によって分解され、tk の中身は次のようになります。

	1 番目のトークン	2 番目のトークン
tk	花形 満	55

区切られた個々の文字列を「トークン」と呼びます。

StringTokenizer クラスには、「nextToken()」メソッドがあり、これにより tk に保管されているトークンを順に取り出すことができます。今の場合、最初の nextToken()メソッドで「氏名」が取り出され、2 番目の nextToken()メソッドで得点を取り出される訳です。

以上のように、StringTokenizer クラスを利用することにより、任意の区切り記号で区切られたデータを順次取り出すことができます。

次に、[表示] ボタンのイベントハンドラは次の通りです。

```
void jButtonDisplay_actionPerformed(ActionEvent e) {
    String Data="";
    for (int i=0;i<Num;i++) {
        Data=Data+Name[i]+"¥n";
    }
    jTextArea1.setText(Data);
}
```

氏名を（1 行に 1 名ずつ）順に表示させるので、「Name[i]+改行記号」を連結させた結果を、jTextArea1 に表示させています。

プログラムを作成したら実行し、動作を確認して下さい。

3 - 7 ファイル入出力の応用

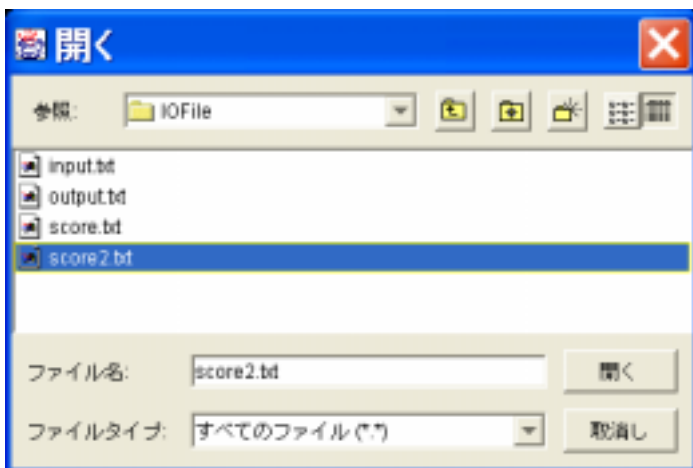
【応用課題 3-D】

【基礎課題 3-7】を修正して、次のように、最高点をとった人の氏名とその得点を表示させるプログラムを作成して下さい。



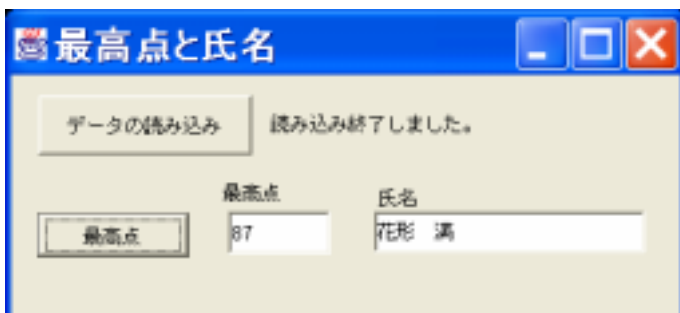
プログラムを起動すると次の画面が現れます。

ここで、[データの読み込み]ボタンをクリックすると・・・



入力ファイル指定のダイアログボックスが現れるので、ここで、「score2.txt」を指定します。

ここまでの動作は【基礎課題 3-7】と同じです。



データ読み込み完了後、[最高点]ボタンをクリックすると、最高点と、それをとった人の氏名が表示されます。

<ヒント>

最高点を求める部分は【基礎課題 3-6】と同様です。そこでは、「勝ち抜き戦」方式で変数 Max に最高点候補が順次代入されて行きました。最高点候補者の氏名についても、同様のことを行えばよいのです。

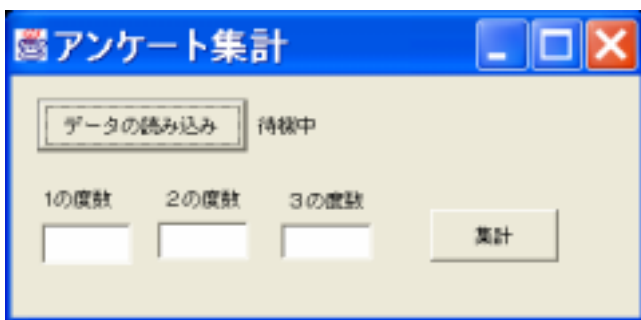
【応用課題 3-E】

【応用課題 2-B】では、3 択のアンケートの集計を行うプログラムを考えました。その際回答データはテキストエリアに直接書き込んだのですが、その回答データ < Ank.txt > をファイルから読み込むようにしましょう。HP の該当部分からファイル「Ank.txt」をダウンロードして下さい。このファイルには 150 名分の回答結果が書き込まれています。

```

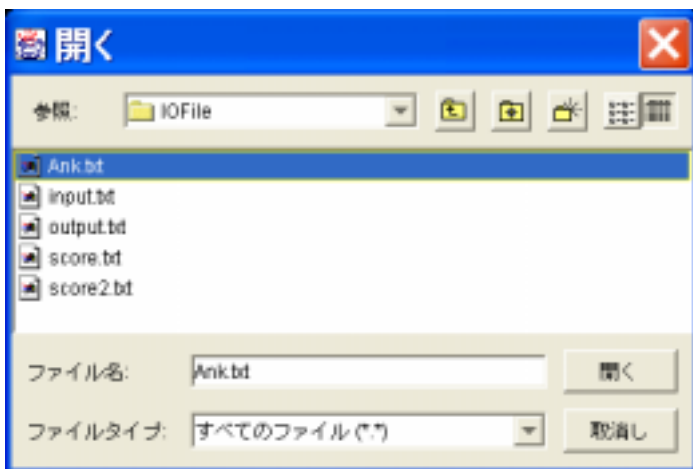
3
2
2
2
. . .
    
```

作成するプログラムの動作は次の通りです。

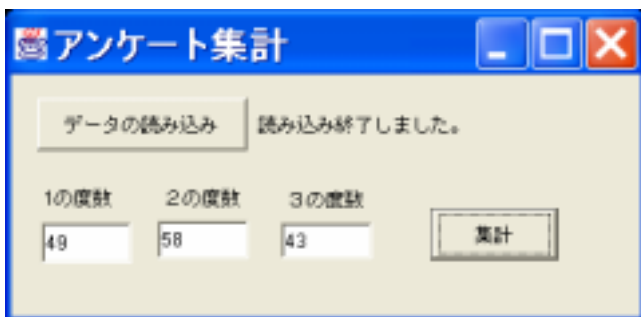


プログラムを起動すると、次のような画面が現れます。

ここで、[データ読み込み] ボタンをクリックすると・・・



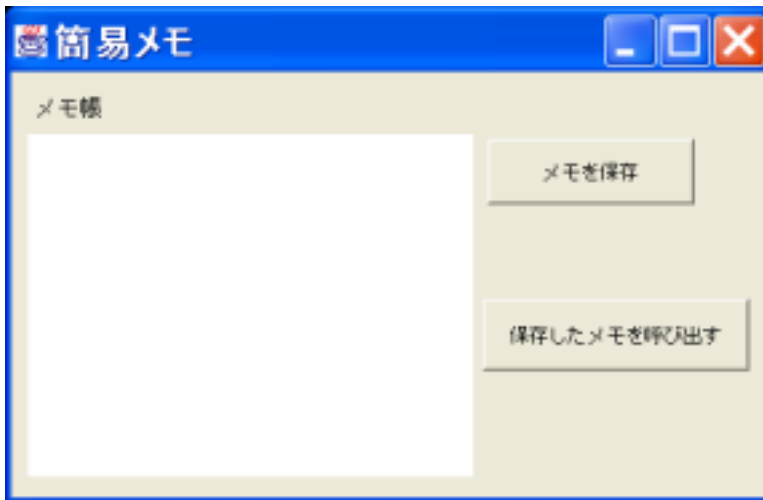
入力ファイル指定のダイアログボックスが現れるので、ここで、ファイル「Ank.txt」を指定します。



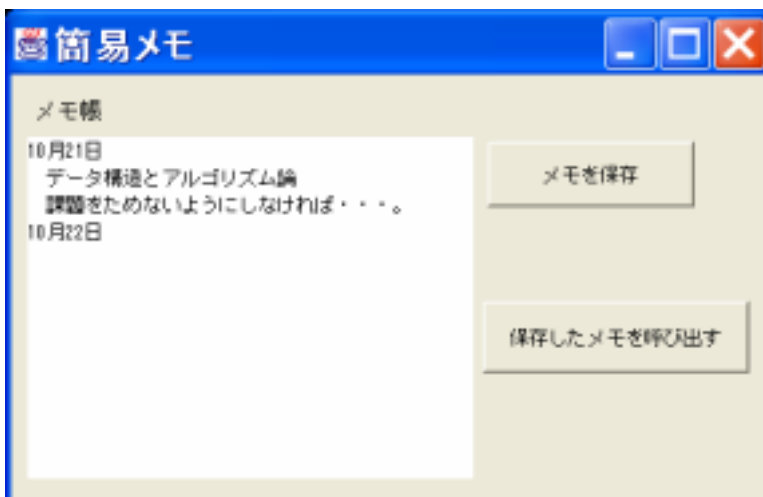
データの読み込み完了後、[集計] ボタンをクリックすると、選択肢 1 ~ 3 それぞれの回答に対する度数が集計され表示されます。

【応用課題 3-F】

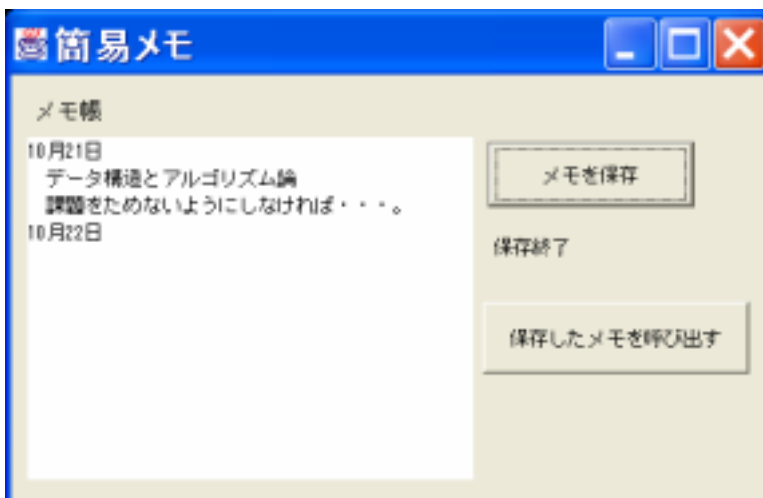
次のような、クイック保存機能のある簡易メモ帳を作って下さい。



プログラムを起動すると次の画面が現れます。

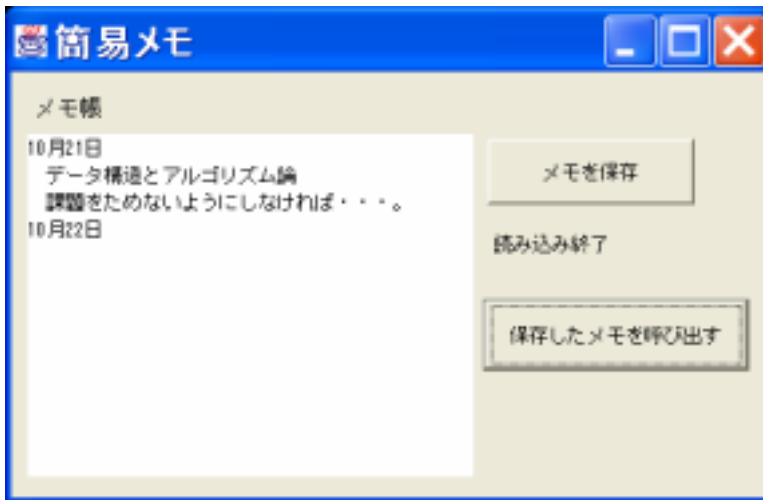


起動後、「メモ帳」欄に適当にメモを記入します。



ここで、作業を中断しなければならなくなったとします。そこで、[メモを保存] ボタンをクリックし、今のメモの内容を保存しておきます。

そしてプログラムを終了させます。



再び、プログラムを起動し、作業を再開します。起動後、[保存したメモを呼び出す]ボタンをクリックします。すると、先程保存したメモの内容が表示されます。

<ヒント>

[メモを保存]ボタンをクリックしたときに、適当なファイル(プログラム中で指定)に、jTextArea コンポーネントの内容を保管する(書き込む)ようにします。

[保存したメモを呼び出す]ボタンをクリックしたときに、上で書き込んだファイルから1行ずつ読み込みます。要領は【基礎課題 3-5】と同じです。そして、読み込んだ内容を、jTextArea コンポーネントに表示させれば良いのです。要領は【基礎課題 3-7】の[表示]ボタンのイベントハンドラと同様です。

<補足>

jTextArea コンポーネントはそのままでは、行端で自動的に折り返してくれません。つまり意図的に改行しない限り次行へ移りません。これでは不便なので、自動的に行端で折り返すようにするためには次のようにします。

プログラムの中ほどに、「コンポーネントの初期化」とコメントされている、jbInit() というメソッドを定義している部分があります。その中の最下部(実はメソッド内であればどこでも構いません)に次の1文を加えて下さい。これは、自動折り返しを有効にする命令です。

//コンポーネントの初期化

```
private void jbInit() throws Exception {
    . . .
    jTextArea1.setLineWrap(true);
}
```