

応用プログラミングII

第2回

岩下・柴田

本日の内容

1. CSVファイルからの情報抽出

- i. CSVファイルから情報抽出してそのまま出力
- ii. 抽出した情報をクラスのインスタンスとして保持
- iii. 抽出した複数の情報をリストとして保持
- iv. (応用)Dateクラスの利用

2. 第1回演習課題のヒント

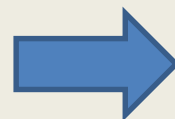
1. CSVファイルからの情報抽出

1-i: カンマ区切りファイルからデータ抽出

- CSVファイル
 - 表データをカンマ区切りのテキスト形式で記述
 - 同じ列が同じ意味を表す

例) TwitterのIDと氏名のデータ

twitterID	氏名
abc	佐藤太郎
123	鈴木花
jkl	高橋一馬

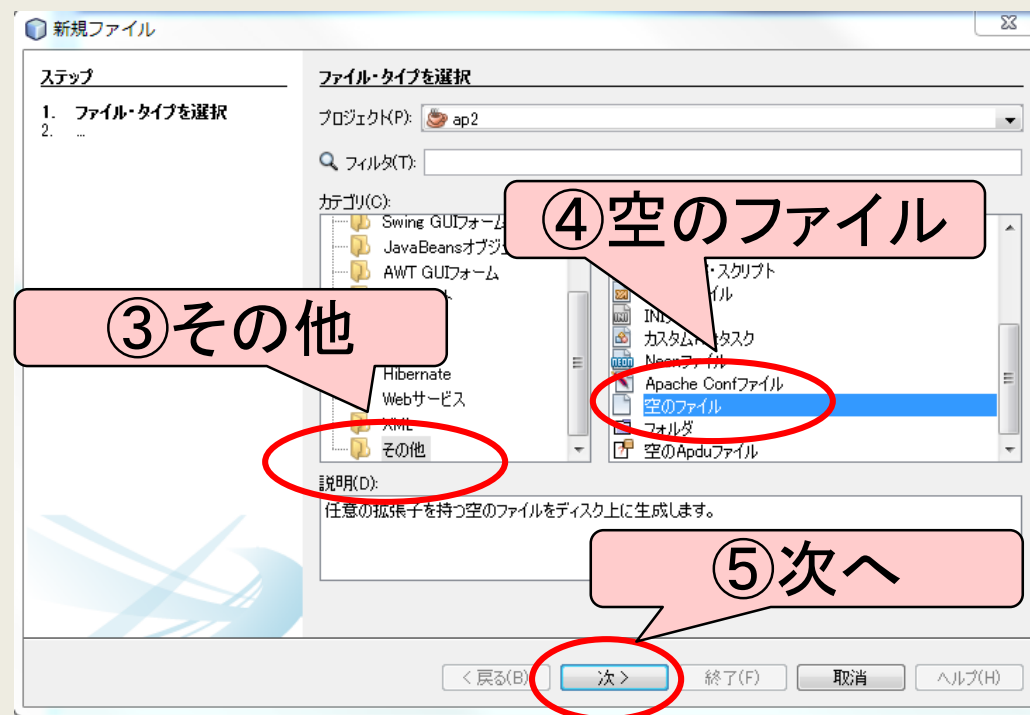
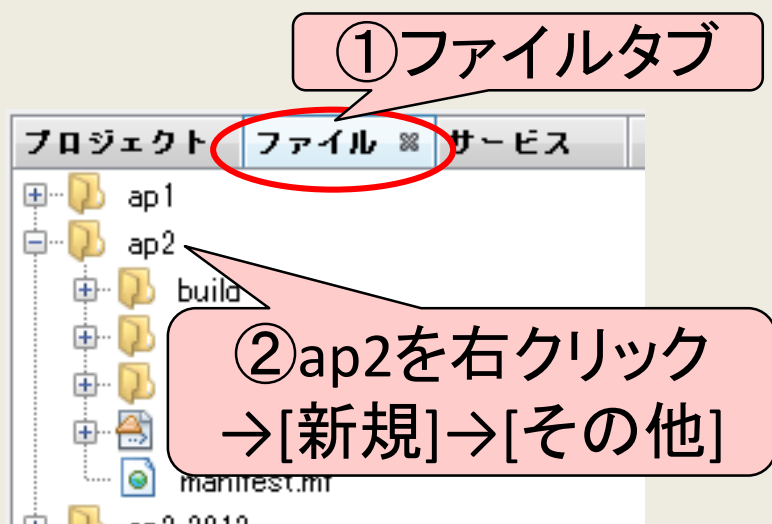


CSV形式
tweet.csv

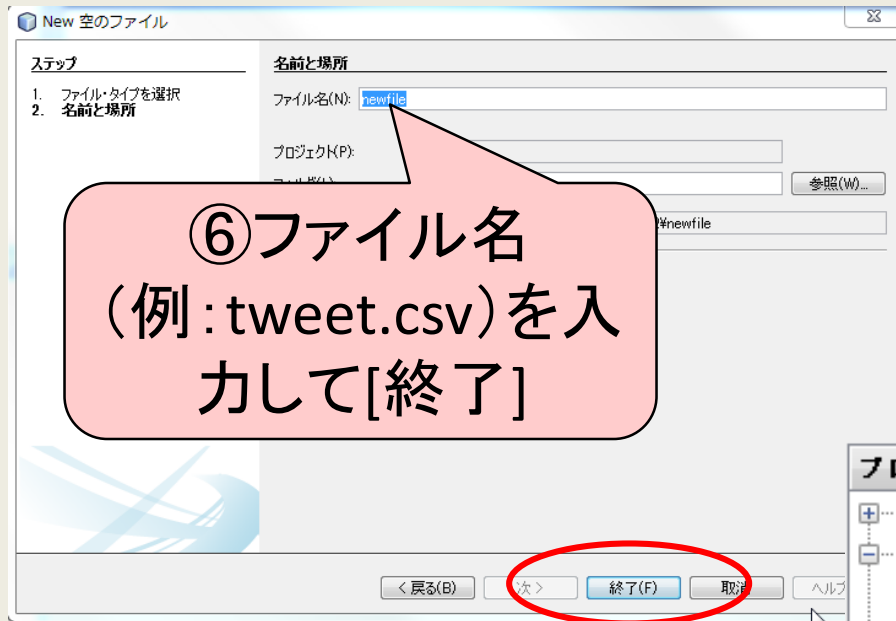
abc,佐藤太郎
123,鈴木花
jkl,高橋一馬

CSVファイルの作成(1)

- ファイルを相対パスで指定した場合, **NetBeansProjects¥ap2** の下にあるファイルを参照する
- NetBeans上でCSVファイルを作成できる

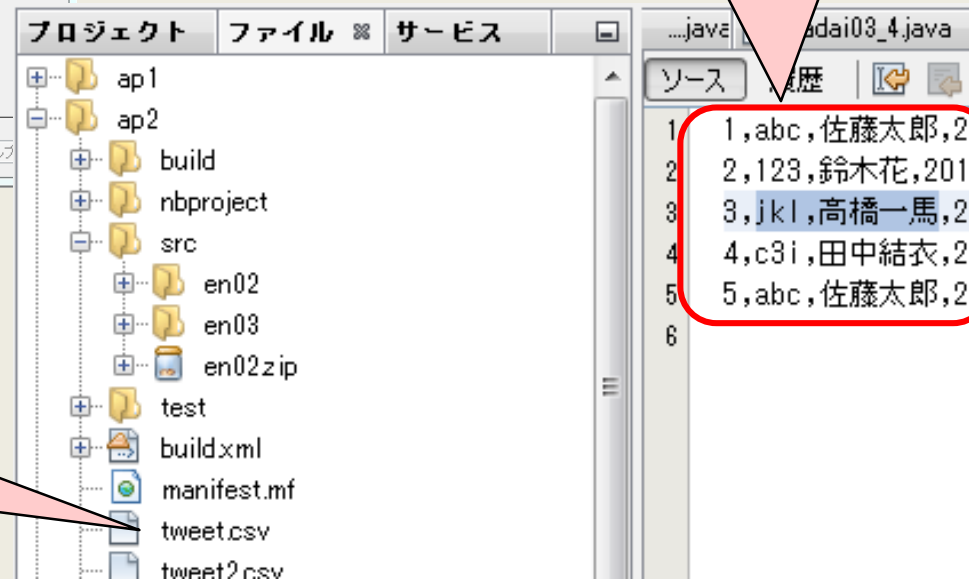


CSVファイルの作成(2)



⑧編集して保存

⑦作成されたファイルをダブルクリック



練習問題【Ex21.java】

CSV形式のファイルから情報抽出

```
public static void main(String[] args) {  
    try {  
        BufferedReader br = new BufferedReader(  
            new FileReader(new File("tweet.csv")));  
  
        String line;  
        //1行ずつ抽出してカンマで区切る  
        while ((line = br.readLine()) != null) {  
            String[] word = line.split(",");  
            System.out.println("twitterID:" + word[0]);  
            System.out.println("氏名:" + word[1]);  
        }  
        br.close();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

Stringクラスのsplitメソッド

```
while ((line=br.readLine()) != null)
```

abc,佐藤太郎
123,鈴木花
jkl,高橋一馬

 line

最後の行まで1行ずつ抽出して
文字列lineに代入

```
String[] word = line.split(",");
```

文字列 line を "," で区切って配列に順番に入れる

line="abc,佐藤太郎"

```
word[0] = "abc"  
word[1] = "佐藤太郎"
```


1-ii: 情報をクラスのインスタンスとして保持

- もっと多くの情報がCSV形式で書かれている場合に, 情報をクラスとして記述しよう

例) Twitterのツイート情報をデータとした場合

No	twitterID	氏名	投稿日時	投稿内容
1	abc	佐藤太郎	2017/09/09 10:56	つかれた
2	123	鈴木花	2017/09/18 21:18	やったね!
3	jkl	高橋一馬	2017/10/01 22:43	何してる?
4	c3i	田中結衣	2017/10/02 17:31	(^^)/ ♪

tweet2.csv

- tweet2.csvに表のデータが書かれている
- カンマ区切りのCSVファイル
- 1行＝「1つのツイート」の情報

tweet2.csv

```
1,abc,佐藤太郎,2017/09/02 10:52,つかれた  
2,123,鈴木花,2017/09/19 15:35,やったね!  
3,jkl,高橋一馬,2017/10/01 23:07,何してる?  
4,c3i,田中結衣,2017/10/02 04:38,(^^)/♪
```

Tweetクラス(1)

- ツイート番号, twitterID, 氏名, 投稿日時, 投稿内容をフィールドとして持つクラス
- Tweetクラスのインスタンスは, 1つのツイート情報を保持する
 - 各情報をセットするメソッドを作成
- Tweetクラスには, あるツイートの情報を出力するメソッド
`void printTweet()`を用意

Tweetクラス プログラム(1)

```
public class Tweet {  
  
    //フィールド  
    private int tweetNo;           //ツイートの番号  
    private String twitterID;      //twitterID  
    private String name;           //氏名  
    private String tweetTime;      //投稿日時  
    private String text;           //投稿内容  
}
```

つづく

Tweetクラス プログラム(2)

```
//ツイート番号をセットするメソッド  
public void setNo(int tn) {  
    this.tweetNo = tn;  
}  
  
//twitterIDをセットするメソッド  
public void setTID(String tid) {  
    this.twitterID = tid;  
}
```

氏名, 投稿日時, 投稿内容をセットするメソッド
setName, setTime, setTextを自分で書く

Tweetクラス プログラム(3)

```
//ツイート情報出力するメソッド  
public void printTweet() {  
    System.out.println(tweetNo + ":"  
        + twitterID + "(" + name + ")%n"  
        + tweetTime + "%n" + text);  
}
```

【出力結果の形式】

tweetNo:twitterID(name)

tweetTime

text

出力例)

1:abc(佐藤太郎)

2017/09/02 10:52

つかれた

メインクラス【Ex22】からTweetクラスを利用(1)

- メインクラス Ex22.java
- tweet2.csvファイルからツイートの情報を取得して表示

```
public static void main(String[] args) {  
    // TODO code application logic here  
  
    //あるつぶやきの情報を保持するインスタンス  
    Tweet tweet = new Tweet();  
  
    try {  
        BufferedReader br = new BufferedReader(  
            new FileReader(new File("tweet2.csv")));  
        String line = "";
```

つづく

メインクラスからTweetクラスを利用(2)

```
//1行ずつファイルから読み込む
while ((line = br.readLine()) != null) {
    //カンマで区切り、配列wordへ順番に代入する
    String[] word = line.split(",");
    tweet.setNo(Integer.parseInt(word[0]));
    tweet.setTID(word[1]);
    tweet.setName(word[2]);
    tweet.setTime(word[3]);
    tweet.setText(word[4]);

    //つぶやきの情報を出力
    tweet.printTweet();
}
br.close();
```

setNoの引数
はint型

tweetのメンバ変数は
1行読み込むごとに
毎回上書きされるので
各ツイートの情報を
後で使えない



次のListを使う！

1-iii: 情報をリストとして保持

- Ex22.javaでは、毎回tweetの内容が上書きされるので、後からその各ツイートの情報を使うことができない
- Listを使い、各ツイートの情報を格納する
→ Listから情報を取得できるようになる

List

- インタフェース java.util.List
- 順番を持つ複数データを操作できる
- 要素を番号で管理するクラス: ArrayList
- 今回は, Tweetクラスのインスタンスを追加するArrayListを作成するため, ジェネリクスを使用する

```
List<Tweet> tlist = new ArrayList<>();
```

tlistはTweetクラスのインスタンスを追加するArrayList

Listのメソッド

- void add(E o)
- E get(int index)
- int size()
- E remove(int index) など

例)

```
List<Tweet> tlist=new ArrayList<>();  
Tweet t = new Tweet();  
tlist.add(t);
```

【Ex23】Listによる複数情報の保持(1)

- メインクラス: Ex23.java
- ArrayList<Tweet> を用いて複数の情報を保持

```
//複数のつぶやき情報を保持するList
List<Tweet> tlist = new ArrayList<>();

try {
    BufferedReader br = new BufferedReader(
        new FileReader(new File("tweet2.csv")));
    String line = "";
```

つづく

Listによる複数情報の保持(2)

```
while ((line = br.readLine()) != null) {  
    //毎回, Tweet クラスの新しいインスタンスを作成  
    Tweet tweet = new Tweet();  
    String[] word = line.split(",");  
    tweet.setNo(Integer.parseInt(word[0]));  
    tweet.setTID(word[1]);  
    tweet.setName(word[2]);  
    tweet.setTime(word[3]);  
    tweet.setText(word[4]);  
  
    //リストtlistに1つのtweet インスタンスを追加  
    tlist.add(tweet);  
}  
br.close();
```

毎回Tweetクラスの新しいインスタンス
を作成し, リストに保持する



全ての情報を参照可能

Listによる複数情報の保持(3)

- 出力(for文で全ての情報を繰り返し出力)

```
for (int i = 0; i < tlist.size(); i++) {  
    //i番目のツイート情報を取り出す  
    Tweet tweet = tlist.get(i);  
  
    //取り出した情報の内容を入力  
    tweet.printTweet();  
}
```



```
//拡張for文  
for(Tweet tweet: tlist){  
    tweet.printTweet();  
}
```

拡張for文で
書く

Listによる情報保持方法のまとめ

CSVファイル

1,abc,佐藤太郎,2017/09/02 10:52,つかれた
2,123,鈴木花,2017/09/19 15:35,やったね!
...



`br.readLine()`

`line="1,abc,佐藤太郎,2017/09/02
10:52,つかれた"`



`line.split(",")`

`word[0]="1"
word[1]="abc"
word[2]="佐藤太郎"
...`



`tweet.setNo(Integer.parseInt(word[0]));
...`

Tweetクラスのインスタンス tweet

```
int tweetNo = 1;  
String twitterID = "abc";  
String name = "佐藤太郎";  
String tweetTime = "2017/09/02 10:56";  
String text = "つかれた";
```



`tlist.add(tweet)`

ArrayList<Tweet>tlist

ツイート1の情報

ツイート2の情報

... ※下に追加

1-iv: Dateクラスを利用した応用

- Tweetクラスの投稿日時をStringからDateに変更したクラス Tweet2クラスを作成
- 日時による検索が便利になる
(10日前～現在までのつぶやきのみを出力する, など)

```
public class Tweet2 {  
  
    //フィールド  
    private int tweetNo;           //ツイートの番号  
    private String twitterID;      //twitterID  
    private String name;           //氏名  
    private Date tweetTime;        //投稿日時  
    private String text;           //投稿内容  
}
```

Tweetクラスをコピーして
Tweet2クラスを作成

Tweet2クラスの修正

//投稿日時をセットするメソッド

```
public void setTime(String tm) {
```

```
    try {
```

```
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy/MM/dd hh:mm");
```

```
        Date date = sdf.parse(tm);
```

```
        this.tweetTime = date;
```

```
    } catch (ParseException e) {
```

```
        e.printStackTrace();
```

```
    }
```

```
}
```

修正

追加

//投稿日時を取得するメソッド

```
public Date getTime(){
```

```
    return tweetTime;
```

```
}
```

“2017/09/02 10:52”の形式の文字列から
Dateクラスのインスタンスを作成し、
フィールドtweetTimeの値として保持

【Ex24】メインクラスの作成

- Ex23.javaをコピーしてEx24.javaを作成
- Tweetクラス→Tweet2クラスに修正
- リストに追加する部分はほぼ修正なし

```
public static void main(String[] args) {  
    // TODO code application logic here  
  
    //複数のつぶやき情報を保持するList  
    List<Tweet2> tlist = new ArrayList<>();
```

⋮

```
while ((line = br.readLine()) != null) {  
    //毎回, Tweetクラスの新しいインスタンスを作成  
    Tweet2 tweet = new Tweet2();
```

修正

10日前から現在までのつぶやきのみを出力

```
//10日前から現在までのつぶやきのみを出力
for(Tweet2 tweet2: tlist){
    //現在から10日前の日時
    Calendar cal = Calendar.getInstance();
    cal.add(Calendar.DATE, -10);

    //投稿日時
    Calendar cal2 = Calendar.getInstance();
    cal2.setTime(tweet2.getTime());

    //10日前よりも後の日時かどうか
    boolean after10 = cal2.after(cal);

    //10日前よりも後に投稿されたもののみを出力
    if(after10){
        tweet2.printTweet();
    }
}
```

現在時刻から10日前の日時を
Calendarクラスのインスタンス
calにセット

投稿日時を取得して
Calendarクラスのインスタンス
cal2にセット

Calendarクラスのメソッド
「after」で、投稿日時が10日前
の日時より後かどうかを計算

10日前より後の場合のみ出力

2. 第1回演習課題の解説

課題1-1 (1)

- jTextField1とjTextField2からgetText()メソッドを使って、それぞれ名前と点数の文字列を取得
- 点数はInteger.parseInt()を用いて文字列を整数に変換
- 合計値を保持する変数double sumをKadai01_1クラスのフィールドとして記述しておき、追加ボタンが押されたらjTextField2から取得した値をsumに加える
 - double型にint型を足すとdouble型で計算される

課題1-1 (2)

```
public class Kadai01_1 extends javax.swing.JFrame {  
    //合計点数sumと人数num  
    double sum=0.0;  
    int num = 0;
```

変数sumとnumはKadai01_1クラスの
フィールドとして設定しておく
初期値0

課題1-1 (3)

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    //テキストフィールドから文字列を取得→文字列をテキスト領域に書き出す  
    String text1 = jTextField1.getText() + "%n"  
        + jTextField2.getText() + "%n";  
    jTextArea1.append(text1);  
  
    //人数を1増やす  
    num++;  
    //合計値を求める  
    sum += Integer.parseInt(jTextField2.getText());  
    //平均値を求め、出力  
    jTextField3.setText(Double.toString(sum/num));  
  
    jTextField1.setText("");  
    jTextField2.setText("");  
}
```

点数の文字列を整数に変換して合計値に足す

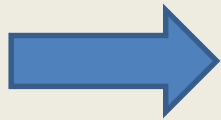
平均値を文字列に変換

課題1-2(1)

- 第1回講義 練習問題Ex12を参照
- out11.txtは奇数行が名前, 偶数行が点数になっている
→ 行数を変数に代入して, 奇数の時と偶数の時で処理を分ける
- 平均点を求めるとき, 人数=行数/2 を使用する

out11.txt

1	うさこ
2	87
3	かばお
4	56
5	ちびぞう
6	79



- 奇数行の処理:
うさこ/かばお/ちびぞう/
- 偶数行の処理
 $(87+56+79)/3$

課題1-2(2)

- 行数を数えて偶数・奇数で処理を分ける

```
String line;  
int linenum=0;  
//1行ずつ読み込む  
while ((line = br.readLine()) != null) {  
    linenum++;  
    //名前の行ならば  
    if (linenum%2==1){  
        namelist += line + "/";  
    }  
    else{  
        sum += Integer.parseInt(line);  
    }  
}
```

1行読むたびにlinenumを1増やす

linenumが奇数の時は名前の処理

偶数の時は合計値を求める

課題1-4のヒント

- in14.txtから1行ずつ読み込む
- 読み込んだ行数がおみくじの数 n になる
- $0 \sim n-1$ の乱数を発生させる
Random rnd = new Random();
int num = rnd.nextInt(n);
- (0から数えて) num番目のおみくじを表示