

# プログラミング入門2

## 第7回 CSVファイルの読み込み

# テーマ:CSVファイルの読み込み

- インスタンスメソッド、コンストラクタ(復習)
- CSVファイルの読み込み(応用)
- 演習
  - 今回までに学んだ事項の復習と応用。

# ねらい

- 様々なケースを用いて、インスタンス作成とその利用法に慣れる。
- 本日の例題
  - 単語帳、島データ
- 本日の問題
  - 電話帳データ
  - 県データ
  - 車のデータ
  - ボールのアニメーション

## ■ インスタンス作成 復習(例題10)

問題：次のクラスMyFlashCardを作成せよ。

クラスMyFlashCardのインスタンスは次のインスタンス変数を持ち、フラッシュカード(単語帳)の情報を保持する。(初期値は指定しなくて良い)

インスタンス変数	初期値	説明
String word1	無し	カードの表
String word2	無し	カードの裏

MyFlashCard のコンストラクタ

コンストラクタ(引数)	機能
MyFlashCard(String word1, String word2)	MyFlashCardクラスのインスタンス変数の値がword1,word2であるMyFlashCardインスタンスを作成する。

(クラス名: MyFlashCard)

# 例題10

```
1 package j2.lesson07;  
2  
3 import gpjava.Spreadsheet;  
4 import gpjava.Canvas;  
5  
6 public class MyFlashCard {  
7     String word1;  
8     String word2;  
9  
10    MyFlashCard(String word1, String word2) {  
11        this.word1 = word1;  
12        this.word2 = word2;  
13    }  
14 }
```

クラスMyFlashCard  
初期値は無し

word1:

word2:

## ■ 例題11(1)

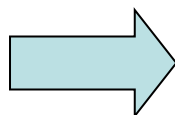
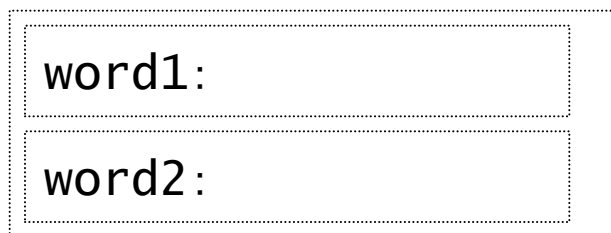
問題: クラスMyFlashCardのインスタンスを4個作成するプログラムを作成せよ。ただし、それぞれのインスタンスはMyFlashCard型の配列に格納するようにせよ。

(クラス名: Ex11FlashCard\_1)

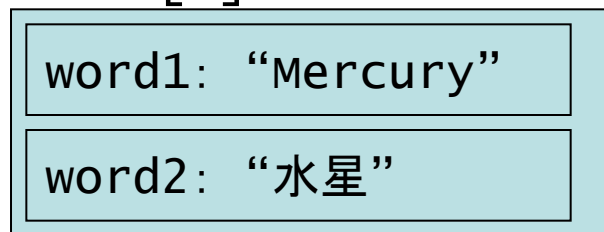
# 例題11(1)

```
1 package j2.lesson07;
2
3 public class Ex11FlashCard_1 {
4
5     public static void main(String[] args) {
6         new Ex11FlashCard_1().start();
7     }
8
9     void start() {
10         int numCards = 4;
11         MyFlashCard[] card = new MyFlashCard[numCards];
12
13         card[0] = new MyFlashCard("Mercury", "水星");
14         card[1] = new MyFlashCard("Venus", "金星");
15         card[2] = new MyFlashCard("Earth", "地球");
16         card[3] = new MyFlashCard("Mars", "火星");
17     }
18 }
```

クラスMyFlashCard  
初期値は無し



card[0]



card[1]

■  
■

## ■ 例題11(2)

問題：（例題11(1)の拡張）作成したMyFlashCardクラスのインスタンスの内容を表示させよ。

### MyFlashCard のインスタンスメソッド

返り値の型	メソッド名(引数)	機能
void	showFields(int row)	引数aCardの内容をSpreadsheetのrow行に表示する。
void	header()	Spreadsheetにヘッダ行を表示する。

### MyFlashCard のコンストラクタ

コンストラクタ(引数)	機能
MyFlashCard(String word1, String word2)	MyFlashCardクラスのインスタンス変数の値がword1,word2であるMyFlashCardインスタンスを作成する。



※既存のEx11FlashCard\_1  
を編集する



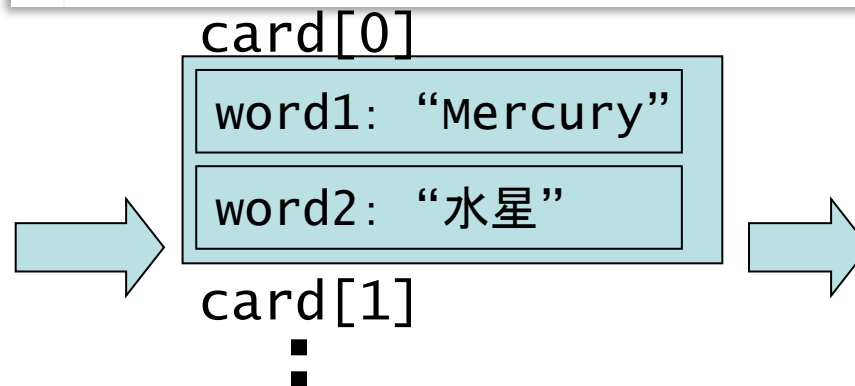
# 例題11(2)

## Ex11FlashCard\_2

```
1 package j2.lesson07;
2
3 import gpjava.Spreadsheet;
4
5 public class Ex11FlashCard_2 {
6
7     public static void main(String[] args) {
8         new Ex11FlashCard_2().start();
9     }
10
11     void start() {
12         int numCards = 4;
13         MyFlashCard[] card = new MyFlashCard[numCards];
14
15         card[0] = new MyFlashCard("Mercury", "水星");
16         card[1] = new MyFlashCard("Venus", "金星");
17         card[2] = new MyFlashCard("Earth", "地球");
18         card[3] = new MyFlashCard("Mars", "火星");
19
20
21         Spreadsheet.show();
22         card[0].header();
23         for (int i = 0; i < numCards; i++) {
24             card[i].showFields(i + 1);
25         }
26     }
27 }
```

クラスMyFlashCard  
初期値は無し

word1:
word2:



スプレッドシート		
ファイル(E) ウィンドウ(W) ヘルプ(H)		
A	B	C
単語1	単語2	
Mercury	水星	
Venus	金星	
Earth	地球	
Mars	火星	

## 例題11(2) MyFlashCard

```
1 package j2.lesson07;
2
3 import gpjava.Spreadsheet;
4 import gpjava.Canvas;
5
6 public class MyFlashCard {
7     String word1;
8     String word2;
9
10    MyFlashCard(String word1, String word2) {
11        this.word1 = word1;
12        this.word2 = word2;
13    }
14
15    //Ex11_2
16    void showFields(int row) {
17        Spreadsheet.setString(row, 0, word1);
18        Spreadsheet.setString(row, 1, word2);
19    }
20
21    //Ex11_2
22    void header() {
23        Spreadsheet.setString(0, 0, "単語1");
24        Spreadsheet.setString(0, 1, "単語2");
25    }
26 }
```

## ■ 例題11(3)

問題：（例題11(2)の拡張）作成したMyFlashCardクラスのインスタンスをもとに、カードの表と裏を連続で表示していくフラッシュカードのプログラムを作成せよ。

カード1枚目表（1秒表示）

→カード1枚目裏（1.5秒表示）

→カード2枚目表（1秒表示）

→カード2枚目裏（1.5秒表示）

...（以下続く）

のようにする。

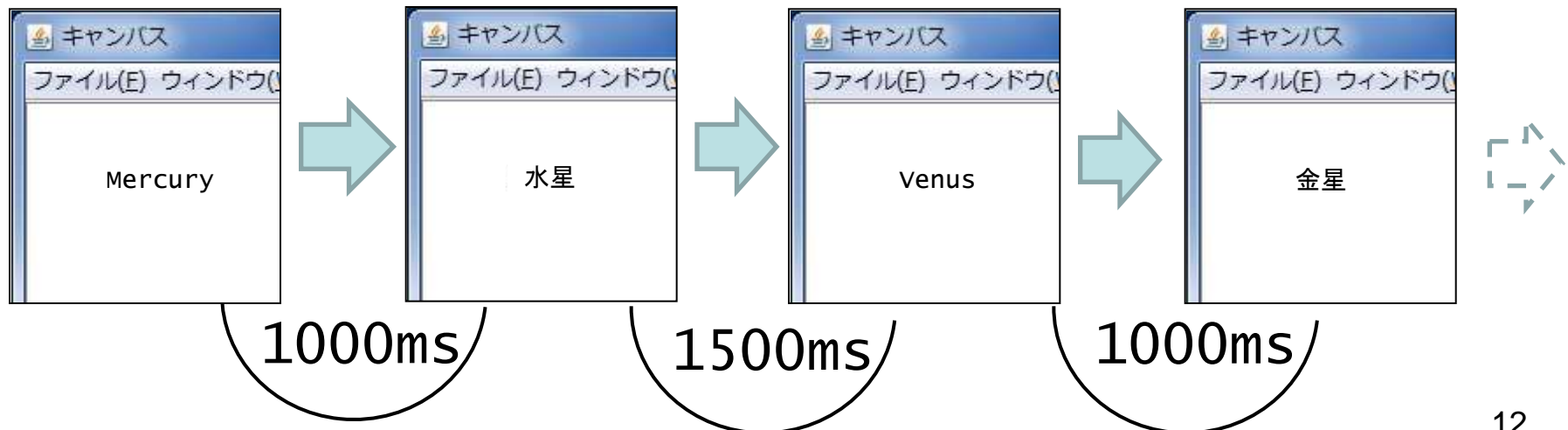
# 例題11(3)

## MyFlashCard のインスタンスメソッド

返り値の型	メソッド名(引数)	機能
void	showCard()	引数aCardのword1、word2の内容を問題文のようなタイミングで順にキャンバスに描画する。

## MyFlashCard のコンストラクタ

コンストラクタ(引数)	機能
MyFlashCard(String word1, String word2)	MyFlashCardクラスのインスタンス変数の値がword1,word2であるMyFlashCardインスタンスを作成する。



# テーマ: CSVファイルの読み込み

- インスタンスメソッド、コンストラクタ(復習)

➡ CSVファイルの読み込み(応用)

- 演習
  - 今回までに学んだ事項の復習と応用。

# CSVファイルの利用

- データをプログラムに直接記述せず、外部のファイルから読み込む。

```
1 package j2.lesson07;  
2  
3 public class Ex11FlashCard_1 {  
4  
5     public static void main(String[] args) {  
6         new Ex11FlashCard_1().start();  
7     }  
8  
9     void start() {  
10         int numCards = 4;  
11         MyFlashCard[] card = new MyFlashCard[numCards];  
12  
13         card[0] = new MyFlashCard("Mercury", "水星");  
14         card[1] = new MyFlashCard("Venus", "金星");  
15         card[2] = new MyFlashCard("Earth", "地球");  
16         card[3] = new MyFlashCard("Mars", "火星");  
17     }  
18 }
```

カードのデータが  
直接プログラムに  
記述されている。  
→違うデータを登録  
するには、プログラム  
を変更する必要がある。

外部ファイルを用いて、様々なデータに対応できるようにする。

Jupiter, 木星  
Saturn, 土星  
Uranus, 天王星  
Neptune, 海王星

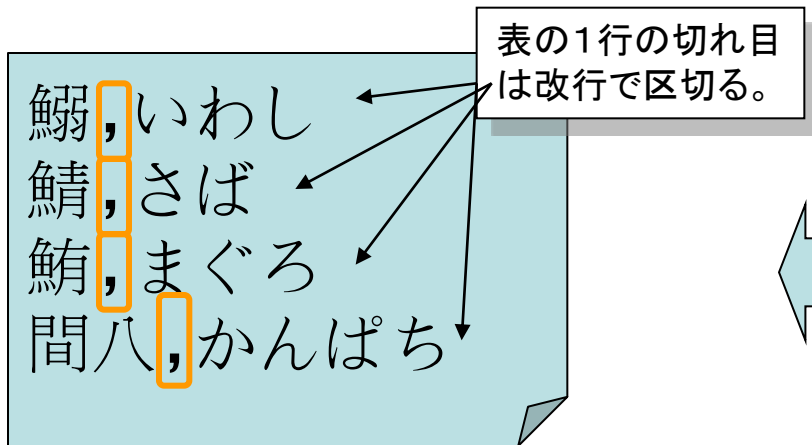
鰯, いわし  
鯖, さば  
鮪, まぐろ  
間八, かんぱち

JPN, Tokyo  
CHN, Beijing  
USA, Washington D.C.  
GBR, London

# CSVファイル

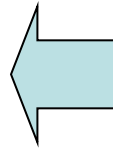
CSVファイル (comma separated values)

複数のデータを、次のような  
表の形式で記述したテキス  
トファイル。



表の1行の切れ目は改行で区切る。

鰯	いわし
鯖	さば
鯖	まぐろ
間八	かんぱち



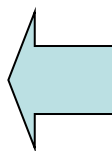
行の中のデータとデータの切れ目は、カンマ(,)で区切る。(タブなど他の区切り記号を用いる場合もある。)

カードデータ(MyFlashCard)

カード表	カード裏
鰯	いわし
鯖	さば
鯖	まぐろ
間八	かんぱち

# CSVファイルの例

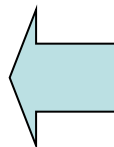
100,70,20,FF00FF  
100,20,30,00FF00  
200,50,40,FF0000  
100,40,20,000FFF



車データ(PaintedCar)

幅	高さ	直径	色
100	70	20	FF00FF
100	20	30	00FF00
200	50	40	FF0000
100	40	20	000FFF

沖縄島, 1206.49  
淡路島, 592.17  
伊豆大島, 91.06  
与那国島, 28.91



島データ(island)

名前	面積
沖縄島	1206.49
淡路島	592.17
伊豆大島	91.06
与那国島	28.91



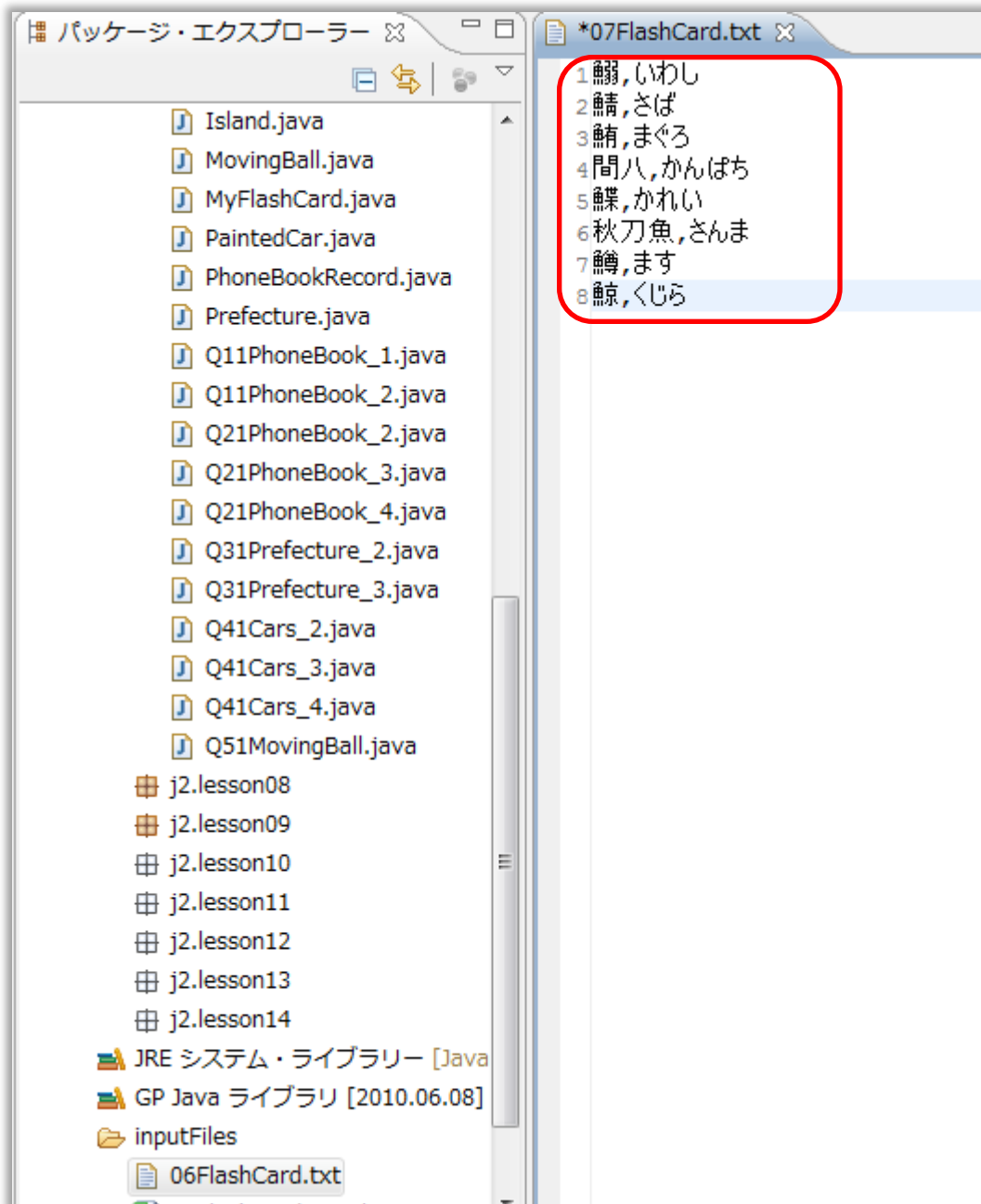
## ■ 例題21(1)より

問題：フラッシュカードのデータを記述した次のcsvファイルを作成せよ。

鰯, いわし  
鯖, さば  
鮪, まぐろ  
間八, かんぱち  
鰺, かれい  
秋刀魚, さんま  
鰯, ます  
鯨, くじら

(ファイル名 07FlashCard.txt)

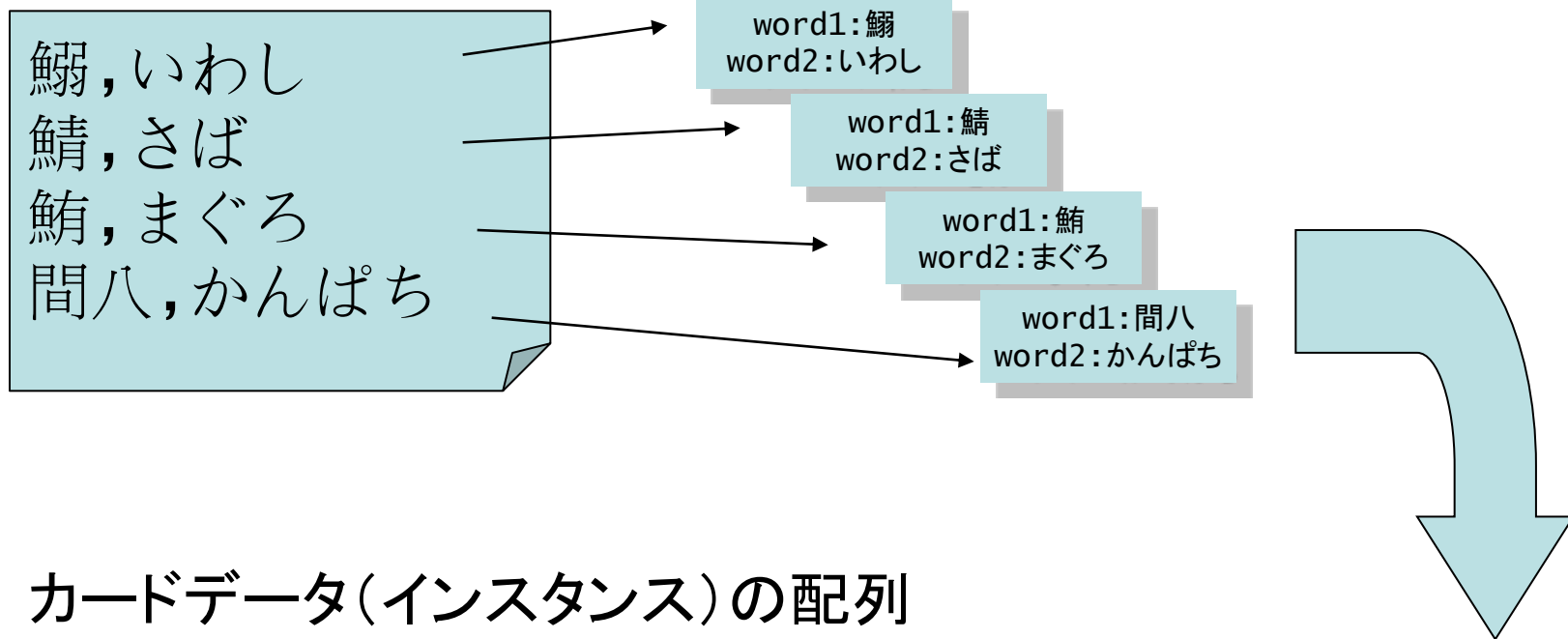
(ファイルの作り方は例題集の解説を参照)



# CSVファイルのデータの読み込みと 配列への格納(イメージ図)

CSVファイル

カードデータ(インスタンス)



カードデータ(インスタンス)の配列

[0]	[1]	[2]	[3] card
word1: 鰯 word2: いわし	word1: 鯖 word2: さば	word1: 鮪 word2: まぐろ	word1: 間八 word2: かんぱち

## ■ CSVファイルの読み込み(例題21(2) より)

問題: 例題21(1)で作成したcsvファイルを読み込み、フラッシュカードのデータを保持させよ。1レコード分(1行分=1カード分)のデータごとに、MyFlashCardクラスのインスタンスを一つ用意して、それぞれのインスタンスは作成した順に配列cardに格納せよ。

返り値の型	メソッド名(引数)	機能
<i>MyFlashCard</i>	<i>createMyFlashCard</i> ( <i>String word1</i> , <i>String word2</i> )	<i>MyFlashCard</i> クラスのインスタンス変数の値が <i>word1</i> , <i>word2</i> である <i>MyFlashCard</i> インスタンスを作成し、返す。(前と同じ)

(クラス名: Ex21FlashCard\_2)

## 例題21(2)

```
1 package j2.lesson07;
2
3 import java.io.*;
4
5 public class Ex21FlashCard_2 {
6
7     public static void main(String[] args) throws IOException {
8         new Ex21FlashCard_2().start();
9     }
10
11     void start() throws IOException {
12         int maxRecords = 1000;
13         MyFlashCard[] card = new MyFlashCard[maxRecords];
14
15         BufferedReader reader = new BufferedReader(new FileReader(
16             "inputFiles/07FlashCard.txt"));
17         int index = 0;
18         while (index < maxRecords) {
19             String line = reader.readLine();
20             if (line == null) { // null なら終了
21                 break;
22             }
23             String[] strs = line.split(",");
24             String word1 = strs[0]; // 第1列(カード表)
25             String word2 = strs[1]; // 第2列(カード裏)
26
27             card[index] = new MyFlashCard(word1, word2);
28             index++;
29         }
30     }
31 }
```

# CSVファイルの読み込み (概要)

CSVファイルから個々のデータを読み取るプログラムの流れの概観は、次のとおりである。

- (1) ファイルをオープンする。  
(読み込むファイルを指定し、読み込みの用意をする。)
- (2) 1データずつ読み込み、処理を行う。
- (3) 必要な処理が終わったら、ファイルをクローズする。

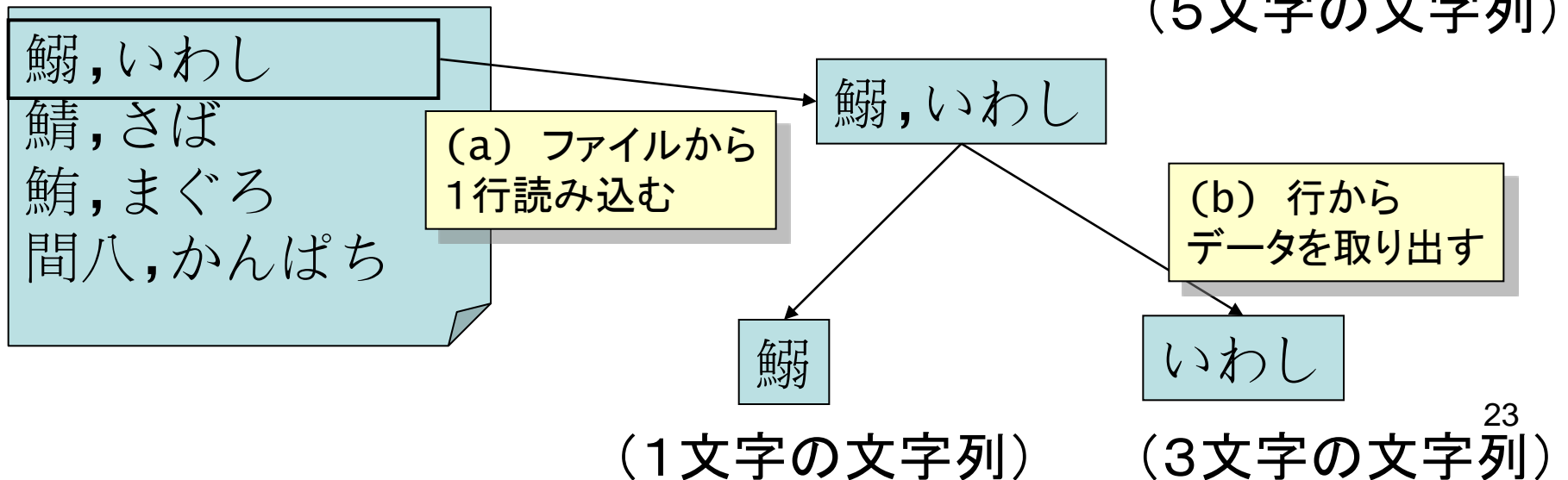
(2) の部分は、さらに具体的にすると次のようになる。(疑似コード)

```
次を必要な回数繰り返す {  
  (a) 1行分を読み込む。  
  ただし、読み込むべき新しい行がない場合は、ループを抜ける。  
  
  (b) 読み込んだ行から一つずつデータを取り出す。  
      → 取り出す際には区切り文字(カンマ)を目印にする。  
  (c) 取り出したデータからインスタンスを生成し、配列に格納  
}
```

# CSVファイルからデータを切り出す

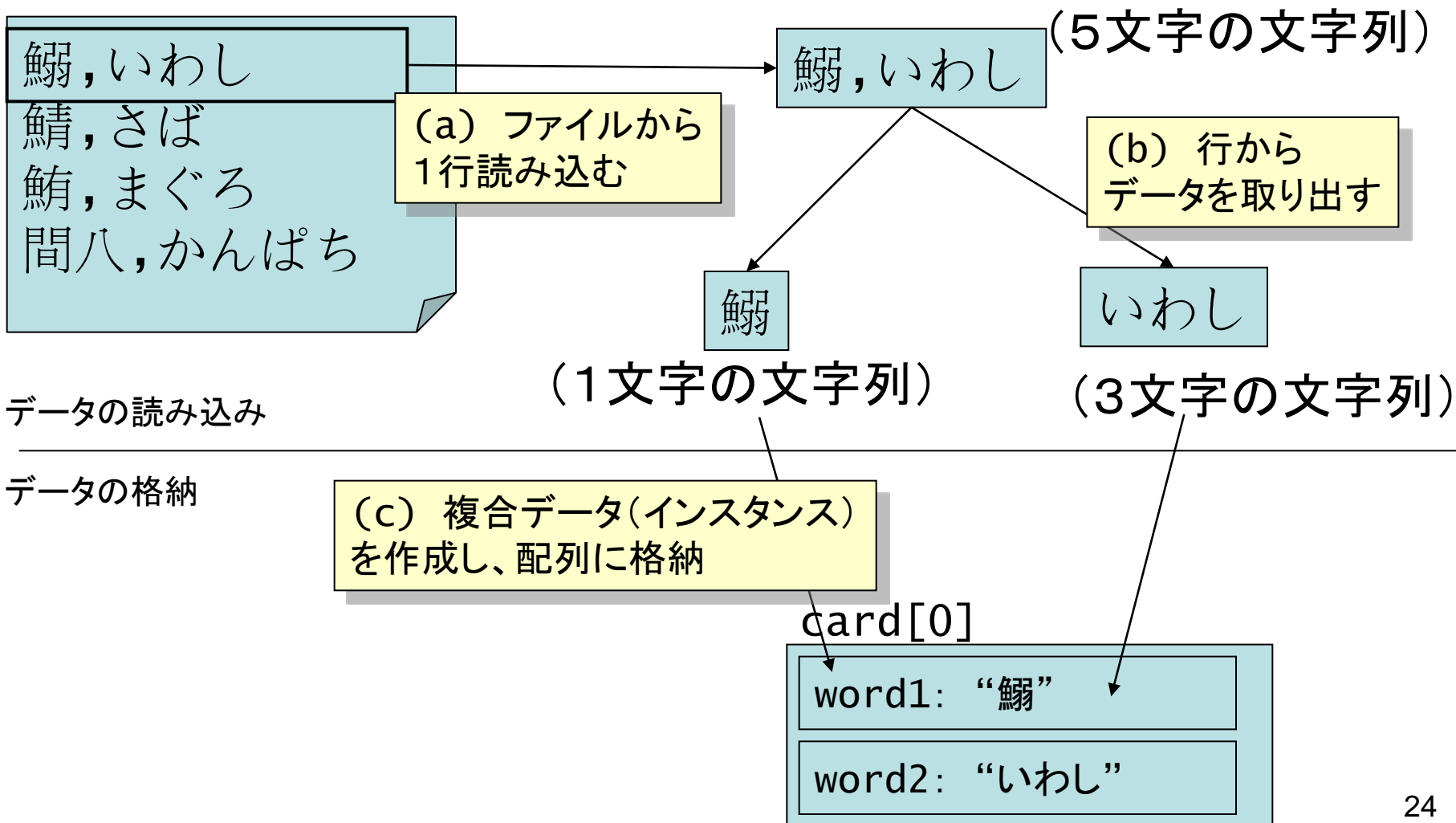
(2) の部分は、さらに具体的にすると次のようになる。(疑似コード)

```
次を必要な回数繰り返す {  
  (a) 1行分を読み込む。  
  ー ただし、読み込むべき新しい行がない場合は、ループを抜ける。  
  
  (b) 読み込んだ行から一つずつデータを取り出す。  
  ー → 取り出す際には区切り文字(カンマ)を目印にする。  
  
  (c) 取り出したデータからインスタンスを生成し、配列に格納。  
}
```



今回は、ファイルから読み込んだデータをもとに、複合データ(インスタンス)を生成し、これを配列に格納する。

すべてのデータを配列に格納した後に、そのデータを利用する。





# CSVファイル 読み取り (概要)

```
int maxRecords = 1000;
MyFlashCard[] card = new MyFlashCard[maxRecords];

BufferedReader reader = new BufferedReader(new FileReader(
    "inputFiles/07FlashCard.txt"));

int index = 0;
while (index < maxRecords) {
    String line = reader.readLine();
    if (line == null) { // null なら終了
        break;
    }
    String[] strs = line.split(",");
    String word1 = strs[0]; // 第1列(カード表)
    String word2 = strs[1]; // 第2列(カード裏)

    card[index] = new MyFlashCard(word1, word2);
    index++;
}
```

- (1) ファイルをオープンする。 ← (読み込むファイルを指定し、読み込みの用意をする。)
- (2) 1データずつ読み込み、処理を行う。 ←
- (3) 必要な処理が終わったら、ファイルをクローズする。 ←

プログラムの終了の際に、クローズは行われるので  
今回は、明示的なクローズの処理は含まれない。

## データの読み取りと配列への格納

```
int maxRecords = 1000;
MyFlashCard[] card = new MyFlashCard[maxRecords];

BufferedReader reader = new BufferedReader(new FileReader(
    "inputFiles/07FlashCard.txt"));

int index = 0;
while (index < maxRecords) {
    String line = reader.readLine();
    if (line == null) { // null なら終了
        break;
    }
    String[] strs = line.split(",");
    String word1 = strs[0]; // 第1列(カード表)
    String word2 = strs[1]; // 第2列(カード裏)

    card[index] = new MyFlashCard(word1, word2);
    index++;
}
```

必要な回数繰り返す {

(a) 1行分を読み込む。

ただし、読み込むべき新しい行がない場合は、ループを抜ける。

(b) 読み込んだ行から一つずつデータを取り出す。

→ 取り出す際には区切り文字(カンマ)を目印にする。

(c) 取り出したデータからインスタンスを生成し、配列に格納

}

```

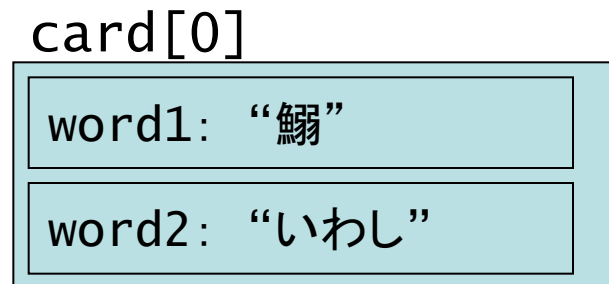
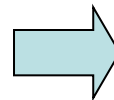
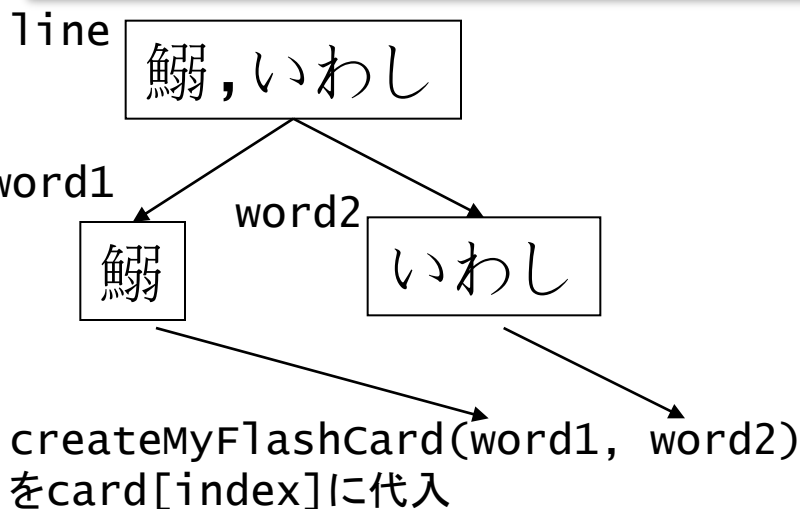
int maxRecords = 1000;
MyFlashCard[] card = new MyFlashCard[maxRecords];

BufferedReader reader = new BufferedReader(new FileReader(
    "inputFiles/07FlashCard.txt"));
int index = 0;
while (index < maxRecords) {
    String line = reader.readLine();
    if (line == null) { // null なら終了
        break;
    }
    String[] strs = line.split(",");
    String word1 = strs[0]; // 第1列(カード表)
    String word2 = strs[1]; // 第2列(カード裏)

    card[index] = new MyFlashCard(word1, word2);
    index++;
}

```

インスタンス  
生成と  
配列への格納  
(例)



# ■読み込んだデータの表示(例題21(3)より)

問題: 例題21(2)で読み込んで保持させたカードの内容をコンソールに表示させよ。

MyFlashCard のインスタンスメソッド

返り値の型	メソッド名(引数)	機能
<i>void</i>	<i>showFields(int row)</i>	インスタンスの内容をSpreadsheet の <i>row</i> 行に表示する。(既に存在する)



A	B	C
単語1	単語2	
弱	いわし	
鯖	さば	
鮪	まぐろ	
間八	かんぱち	
鰈	かれい	
秋刀魚	さんま	
鰯	ます	
鯨	くじら	

※既存のEx21FlashCard\_2を編集する

# 例題21(3)

## Ex21FlashCard\_3

```
1 package j2.lesson07;
2
3 import gpjava.Spreadsheet;
4
5 import java.io.*;
6
7 public class Ex21FlashCard_3 {
8
9     public static void main(String[] args) throws IOException {
10         new Ex21FlashCard_3().start();
11     }
12
13     void start() throws IOException {
14         int maxRecords = 1000;
15         MyFlashCard[] card = new MyFlashCard[maxRecords];
16
17         BufferedReader reader = new BufferedReader(new FileReader(
18             "inputFiles/07FlashCard.txt"));
19         int index = 0;
20         while (index < maxRecords) {
21             String line = reader.readLine();
22             if (line == null) { // null なら終了
23                 break;
24             }
25             String[] strs = line.split(",");
26             String word1 = strs[0]; // 第1列(カード表)
27             String word2 = strs[1]; // 第2列(カード裏)
28
29             card[index] = new MyFlashCard(word1, word2);
30             index++;
31         }
32         int numRecords = index;
33
34         Spreadsheet.show();
35
36         card[0].header();
37         for (int i = 0; i < numRecords; i++) {
38             card[i].showFields(i + 1);
39         }
40     }
41 }
```

配列に格納した  
データを閲覧する

# CSVファイルの読み込み(解説)

## (各自で読むこと)

- ファイルの読み込みの詳細は、後の講義で扱う予定である。
- 今回は、例題21, 例題41などを参考に、大まかな流れが分かればよい。

```

int maxRecords = 1000;
MyFlashCard[] card = new MyFlashCard[maxRecords];

BufferedReader reader = new BufferedReader(new FileReader(
    "inputFiles/07FlashCard.txt"));

int index = 0;
while (index < maxRecords) {
    String line = reader.readLine();
    if (line == null) { // null なら終了
        break;
    }
    String[] strs = line.split(",");
    String word1 = strs[0]; // 第1列(カード表)
    String word2 = strs[1]; // 第2列(カード裏)

    card[index] = new MyFlashCard(word1, word2);
    index++;
}
int numRecords = index;

```

読み取りのための外部メソッドを用いて、  
ファイル読み取りの準備(ファイルのオープン)を行う。

```

3 import java.io.*;
4
5 public class Ex21FlashCard_3 {
6
7     public static void main(String[] args) throws IOException {

```

外部メソッドを利用するために  
必要。

```
int maxRecords = 1000;
MyFlashCard[] card = new MyFlashCard[maxRecords];

BufferedReader reader = new BufferedReader(new FileReader(
    "inputFiles/07FlashCard.txt"));
int index = 0;
while (index < maxRecords) {
    String line = reader.readLine();
    if (line == null) { // null なら終了
        break;
    }
    String[] strs = line.split(",");
    String word1 = strs[0]; // 第1列(カード表)
    String word2 = strs[1]; // 第2列(カード裏)

    card[index] = new MyFlashCard(word1, word2);
    index++;
}
int numRecords = index;
```

1行を読み出し(外部メソッドreadLine())、結果を文字列としてlineに代入する。

読み出す行がない場合(ファイルの終わり(end of file=EOF)まで達した時) nullという定数値が代入される。

→ 従って、line == nullとなるときにループを脱出させる。



```
int maxRecords = 1000;
MyFlashCard[] card = new MyFlashCard[maxRecords];

BufferedReader reader = new BufferedReader(new FileReader(
    "inputFiles/07FlashCard.txt"));
int index = 0;
while (index < maxRecords) {
    String line = reader.readLine();
    if (line == null) { // null なら終了
        break;
    }
    String[] strs = line.split(",");
    String word1 = strs[0]; // 第1列(カード表)
    String word2 = strs[1]; // 第2列(カード裏)

    card[index] = new MyFlashCard(word1, word2);
    index++;
}
int numRecords = index;
```

文字列を“,”で区切って、切り出す。(splitメソッド)

結果は文字列の配列となる。例えば、lineが"沖縄島,1206.49"であれば、配列tokensは、"沖縄島"と"1206.49"の2つの文字列をこの順に保持する。

line = "東京都,2187.42,1257"であれば、配列tokensは、"東京都"と"2187.42"と"1257"の3つの文字列をこの順に保持する。

```
int maxRecords = 1000;
MyFlashCard[] card = new MyFlashCard[maxRecords];

BufferedReader reader = new BufferedReader(new FileReader(
    "inputFiles/07FlashCard.txt"));
int index = 0;
while (index < maxRecords) {
    String line = reader.readLine();
    if (line == null) { // null なら終了
        break;
    }
    String[] strs = line.split(",");
    String word1 = strs[0]; // 第1列(カード表)
    String word2 = strs[1]; // 第2列(カード裏)

    card[index] = new MyFlashCard(word1, word2);
    index++;
}
int numRecords = index;
```

CSVファイルに書かれたデータを、複合データにして、配列に読み込む。

```
int maxRecords = 1000;
MyFlashCard[] card = new MyFlashCard[maxRecords];

BufferedReader reader = new BufferedReader(new FileReader(
    "inputFiles/07FlashCard.txt"));
int index = 0;
while (index < maxRecords) {
    String line = reader.readLine();
    if (line == null) { // null なら終了
        break;
    }
    String[] strs = line.split(",");
    String word1 = strs[0]; // 第1列(カード表)
    String word2 = strs[1]; // 第2列(カード裏)

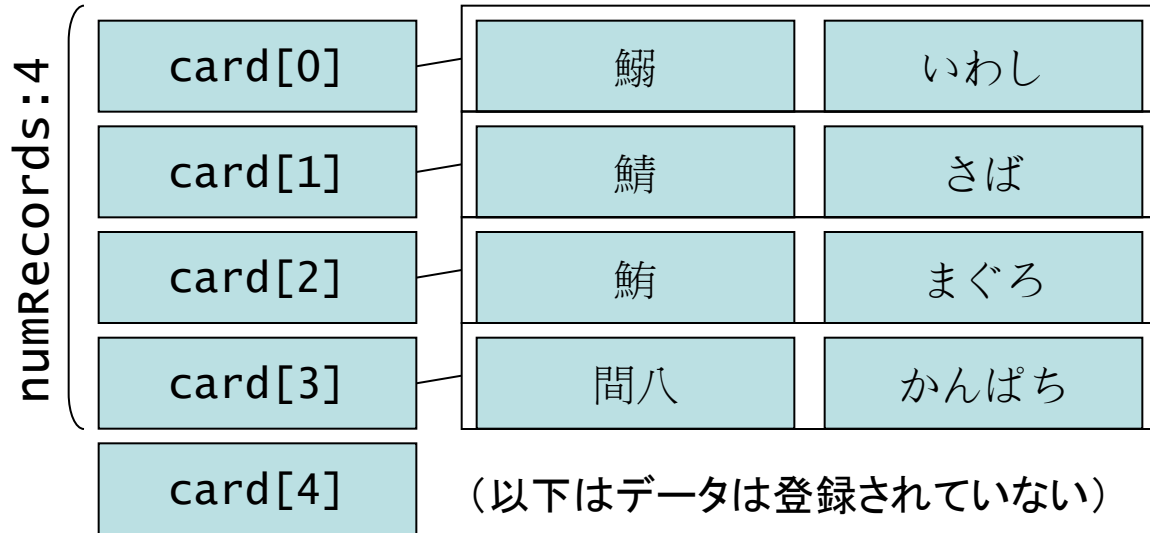
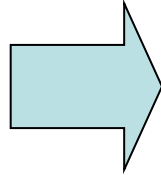
    card[index] = new MyFlashCard(word1, word2);
    index++;
}
int numRecords = index;
```

CSVファイルの表の大きさはプログラムからは決められないので、あらかじめ大きめのサイズで配列を作成しておく。実際に登録したレコード数も記憶しておき、データを処理する際に用いる。(次スライド)

# 配列に格納されるデータのイメージ図

## CSVファイル

鰯,いわし  
鯖,さば  
鮪,まぐろ  
間八,かんぱち



上のファイルの場合、numRecordsは4となる

(プログラムの説明)

- 最後のカード(card[3])を登録する部分ではindexの値は3。
- その後でindex++を実行し、whileループを抜けるのでindexは4となる。よって、numRecordsの値は4である。

```
int index = 0;
while (index < maxRecords) {
    ...
    card[index]
    = createMyFlashCard(word1, word2)
    index++;
}
int numRecords = index;
```

maxRecords - 1  
= 1000 - 1

card[999]

カードを入れる場所はmaxRecords(1000)個ははじめから確保されるが、実際には、読み込んだレコード数(この例では4)だけ蓄積される。

# CSVファイルの読み込み（解説） （ここまで）

- ここで扱った外部メソッドなどの詳細は、後の講義で扱う予定である。
- 今回は、例題21, 例題41などを参考に、大まかな流れが分かればよい。
- この時点で詳細を知りたい場合は、昨年度までの講義資料も参考にせよ。

# テーマ:まとめ(複合データ)

- インスタンスメソッド、コンストラクタ(復習)
- CSVファイルの読み込み(応用)



## 演習

– 今回までに学んだ事項の復習と応用。

# 本日の例題と問題

- FlashCard を用いた例題
  - Ex10, Ex11(1), Ex11(2), Ex11(3), Ex21(1), Ex21(2), Ex21(3), Ex21(4)
- Island の復習
  - Ex30, Ex31(1), Ex31(2), Ex31(3), Ex41(1), Ex41(2), Ex41(3), Ex41(4), (Ex41(5))
- PhoneBookを用いた課題
  - Q10, Q11(1), Q11(2), Q21(1), Q21(2), Q21(3), (Q21(4))
- Prefectureを用いた課題
  - Q30, Q31(1), Q31(2), Q31(3), (Q31(4))
- PaintedCarを用いた課題
  - Q40, Q41(1), Q41(2), Q41(3), (Q41(4))
- MovingBallを用いた課題
  - (Q51\*)

(Ex:例題, Q:問題, \*は少し手間のかかる問題)

各自に適した順番で解けばよいが、上記の順番が自然な流れとなるよう構成されている。

# 例題集



# パッケージ「j2.lesson07」を作成する。

パッケージやクラスの作成, 実行の仕方の説明は省略する。

作り方を忘れた場合は過去のスライドや

<http://java2010.cis.k.hosei.ac.jp/01/material-01/>

を参考にせよ

## ■ 例題10

問題：次のクラスMyFlashCardを作成せよ。

クラスMyFlashCardのインスタンスは次のインスタンス変数を持ち、フラッシュカード(単語帳)の情報を保持する。(初期値は指定しなくて良い)

インスタンス変数	初期値	説明
String word1	無し	カードの表
String word2	無し	カードの裏

MyFlashCard のコンストラクタ

コンストラクタ(引数)	機能
MyFlashCard(String word1, String word2)	MyFlashCardクラスのインスタンス変数の値がword1,word2であるMyFlashCardインスタンスを作成する。

(クラス名: MyFlashCard)

# 例題10

```
1 package j2.lesson07;
2
3 import gpjava.Spreadsheet;
4 import gpjava.Canvas;
5
6 public class MyFlashCard {
7     String word1;
8     String word2;
9
10    MyFlashCard(String word1, String word2) {
11        this.word1 = word1;
12        this.word2 = word2;
13    }
14 }
```

クラスMyFlashCard  
初期値は無し

word1:

word2:

## ■ 例題11(1)

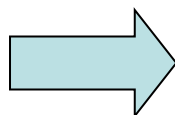
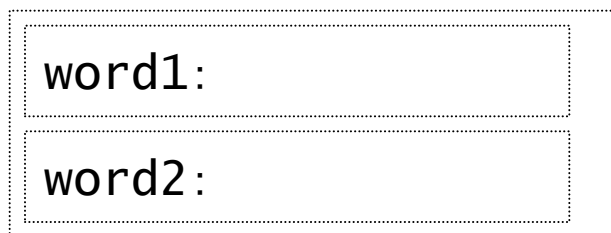
問題：クラスMyFlashCardのインスタンスを4個作成するプログラムを作成せよ。ただし、それぞれのインスタンスはMyFlashCard型の配列に格納するようにせよ。

(クラス名： Ex11FlashCard\_1)

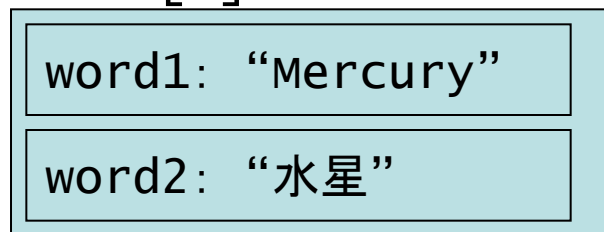
# 例題11(1)

```
1 package j2.lesson07;  
2  
3 public class Ex11FlashCard_1 {  
4  
5     public static void main(String[] args) {  
6         new Ex11FlashCard_1().start();  
7     }  
8  
9     void start() {  
10        int numCards = 4;  
11        MyFlashCard[] card = new MyFlashCard[numCards];  
12  
13        card[0] = new MyFlashCard("Mercury", "水星");  
14        card[1] = new MyFlashCard("Venus", "金星");  
15        card[2] = new MyFlashCard("Earth", "地球");  
16        card[3] = new MyFlashCard("Mars", "火星");  
17    }  
18 }
```

クラスMyFlashCard  
初期値は無し



card[0]



card[1]

■  
■

## ■ 例題11(2)

問題：（例題11(1)の拡張）作成したMyFlashCardクラスのインスタンスの内容を表示させよ。

### MyFlashCard のインスタンスメソッド

返り値の型	メソッド名(引数)	機能
void	showFields(int row)	引数aCardの内容をSpreadsheetのrow行に表示する。
void	header()	Spreadsheetにヘッダ行を表示する。

### MyFlashCard のコンストラクタ

コンストラクタ(引数)	機能
MyFlashCard(String word1, String word2)	MyFlashCardクラスのインスタンス変数の値がword1,word2であるMyFlashCardインスタンスを作成する。



※既存のEx11FlashCard\_1  
を編集する

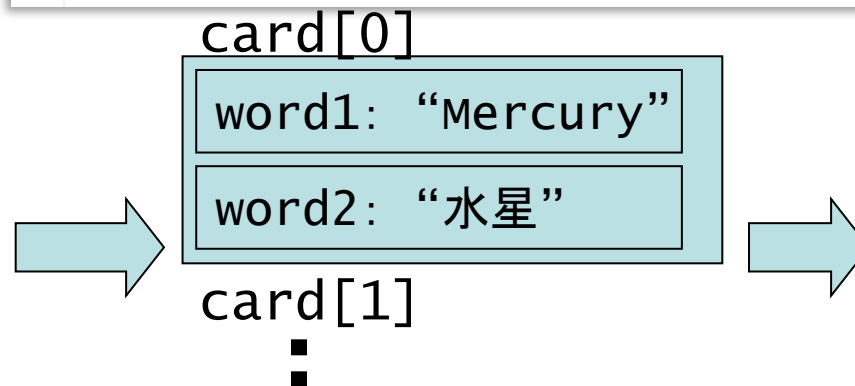
# 例題11(2)

## Ex11FlashCard\_2

```
1 package j2.lesson07;
2
3 import gpjava.Spreadsheet;
4
5 public class Ex11FlashCard_2 {
6
7     public static void main(String[] args) {
8         new Ex11FlashCard_2().start();
9     }
10
11     void start() {
12         int numCards = 4;
13         MyFlashCard[] card = new MyFlashCard[numCards];
14
15         card[0] = new MyFlashCard("Mercury", "水星");
16         card[1] = new MyFlashCard("Venus", "金星");
17         card[2] = new MyFlashCard("Earth", "地球");
18         card[3] = new MyFlashCard("Mars", "火星");
19
20
21         Spreadsheet.show();
22         card[0].header();
23         for (int i = 0; i < numCards; i++) {
24             card[i].showFields(i + 1);
25         }
26     }
27 }
```

クラスMyFlashCard  
初期値は無し

word1:
word2:



スプレッドシート		
ファイル(E) ウィンドウ(W) ヘルプ(H)		
A	B	C
単語1	単語2	
Mercury	水星	
Venus	金星	
Earth	地球	
Mars	火星	

## 例題11(2) MyFlashCard

```
1 package j2.lesson07;
2
3 import gpjava.Spreadsheet;
4 import gpjava.Canvas;
5
6 public class MyFlashCard {
7     String word1;
8     String word2;
9
10    MyFlashCard(String word1, String word2) {
11        this.word1 = word1;
12        this.word2 = word2;
13    }
14
15    //Ex11_2
16    void showFields(int row) {
17        Spreadsheet.setString(row, 0, word1);
18        Spreadsheet.setString(row, 1, word2);
19    }
20
21    //Ex11_2
22    void header() {
23        Spreadsheet.setString(0, 0, "単語1");
24        Spreadsheet.setString(0, 1, "単語2");
25    }
26 }
```



## ■ 例題11(3)

問題：（例題11(2)の拡張）作成したMyFlashCardクラスのインスタンスをもとに、カードの表と裏を連続で表示していくフラッシュカードのプログラムを作成せよ。

カード1枚目表（1秒表示）

→カード1枚目裏（1.5秒表示）

→カード2枚目表（1秒表示）

→カード2枚目裏（1.5秒表示）

...（以下続く）

のようにする。

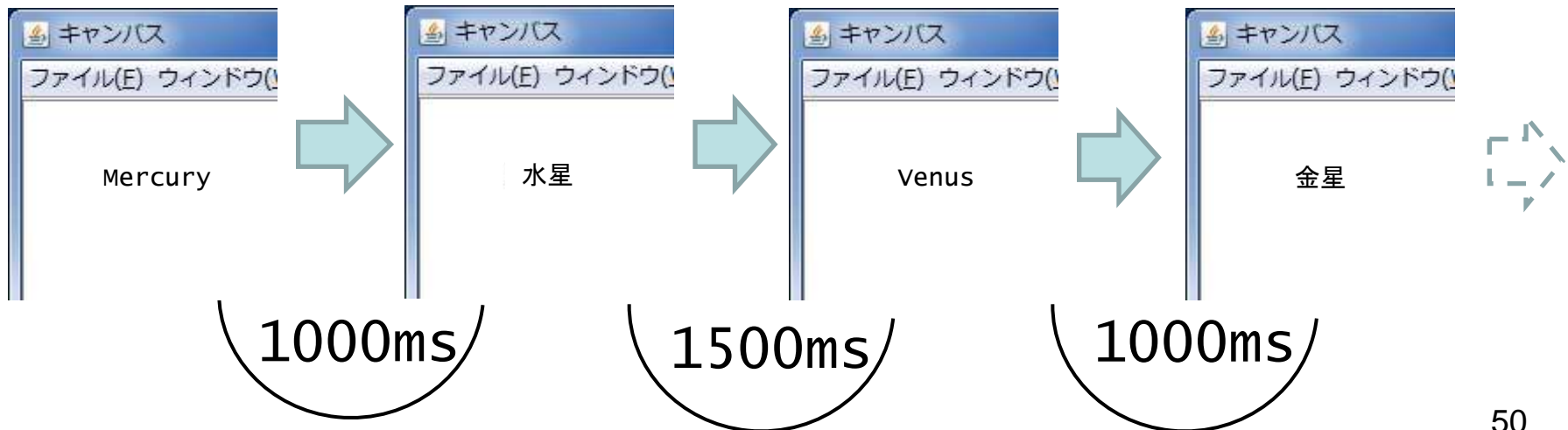
# 例題11(3)

## MyFlashCard のインスタンスメソッド

返り値の型	メソッド名(引数)	機能
void	showCard()	引数aCardのword1、word2の内容を問題文のようなタイミングで順にキャンバスに描画する。

## MyFlashCard のコンストラクタ

コンストラクタ(引数)	機能
MyFlashCard(String word1, String word2)	MyFlashCardクラスのインスタンス変数の値がword1,word2であるMyFlashCardインスタンスを作成する。



## 例題11(3)

### Ex11FlashCard\_3

```
1 package j2.lesson07;
2
3 import gpjava.Canvas;
4
5 public class Ex11FlashCard_3 {
6
7     public static void main(String[] args) {
8         new Ex11FlashCard_3().start();
9     }
10
11     void start() {
12         int numCards = 4;
13         MyFlashCard[] card = new MyFlashCard[numCards];
14
15         card[0] = new MyFlashCard("Mercury", "水星");
16         card[1] = new MyFlashCard("Venus", "金星");
17         card[2] = new MyFlashCard("Earth", "地球");
18         card[3] = new MyFlashCard("Mars", "火星");
19
20         Canvas.show();
21         for (int i = 0; i < numCards; i++) {
22             card[i].showCard();
23         }
24     }
25 }
```

# 例題11(3) MyFlashCard

Ex11(1)  
Ex11(2)  
と同じ

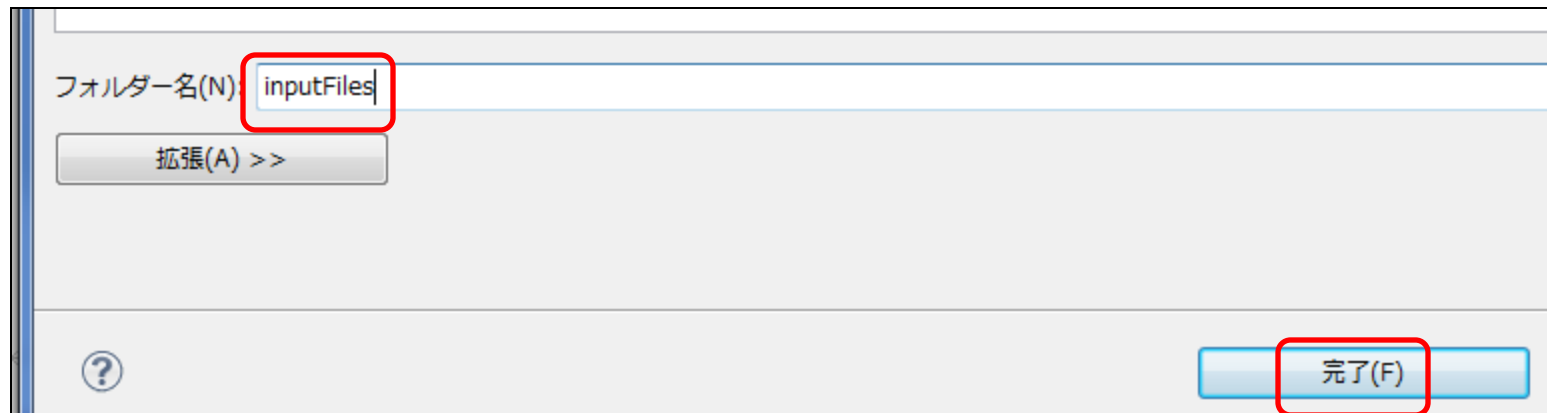
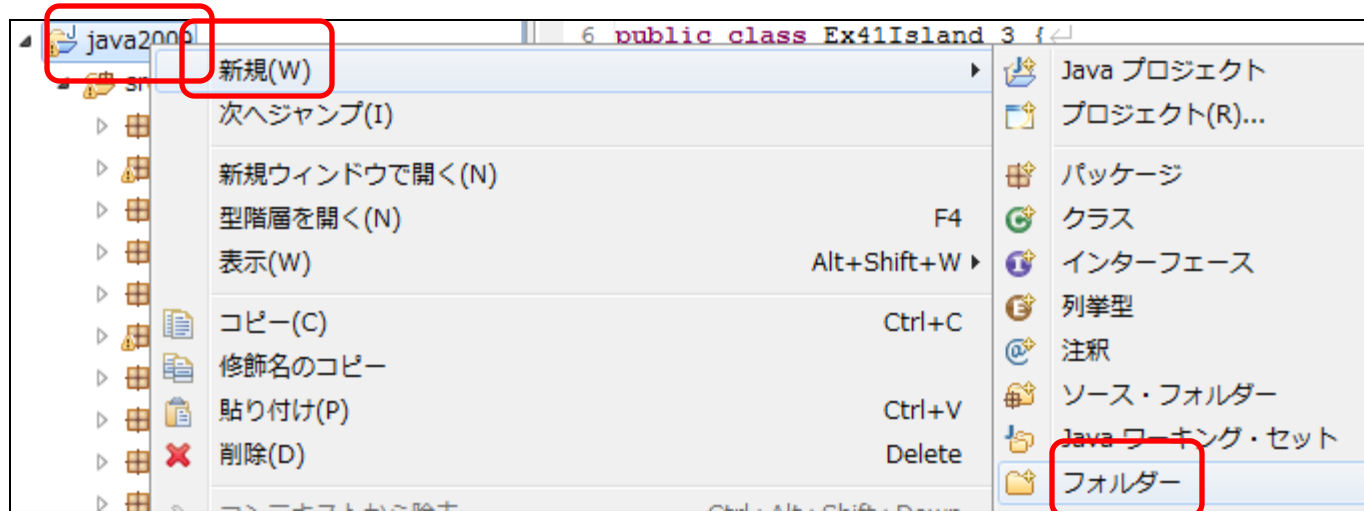
```
1 package j2.lesson07;
2
3 import gpjava.Spreadsheet;
4 import gpjava.Canvas;
5
6 public class MyFlashCard {
7     String word1;
8     String word2;
9
10    MyFlashCard(String word1, String word2) {
11        this.word1 = word1;
12        this.word2 = word2;
13    }
14
15    //Ex11_2
16    void showFields(int row) {}
17
18    //Ex11_2
19    void header() {}
20
21    //Ex11_3
22    void showCard() {
23        Canvas.drawString(50, 50, word1);
24        Canvas.waitForCountdown(1000);
25        Canvas.clear();
26
27        Canvas.drawString(50, 50, word2);
28        Canvas.waitForCountdown(1500);
29        Canvas.clear();
30    }
31 }
32 }
```

## ■ 例題21(1)

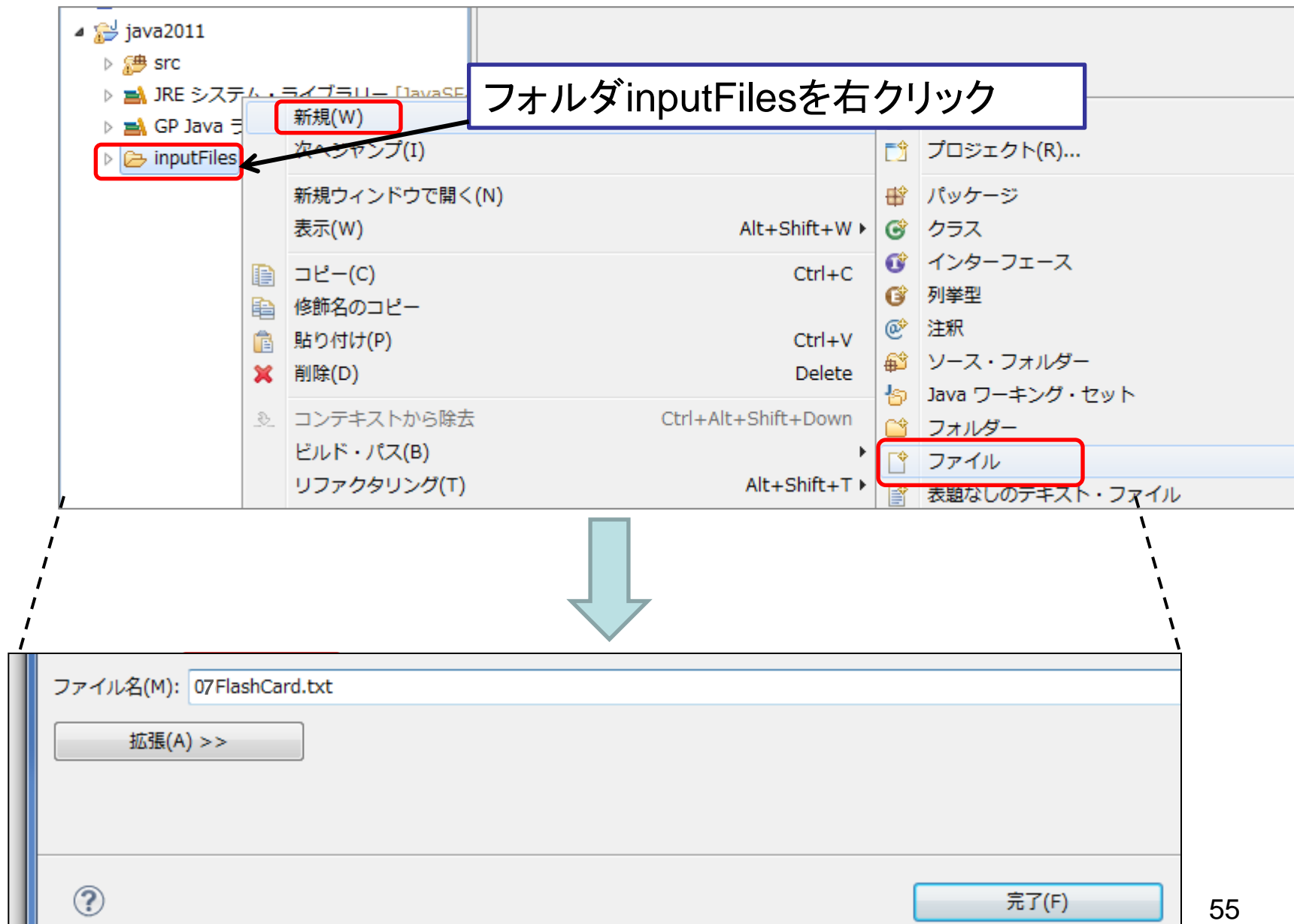
問題：フラッシュカードのデータを記述した次のcsvファイルを作成せよ。

鰯, いわし  
鯖, さば  
鮪, まぐろ  
間八, かんぱち  
鰺, かれい  
秋刀魚, さんま  
鰯, ます  
鯨, くじら

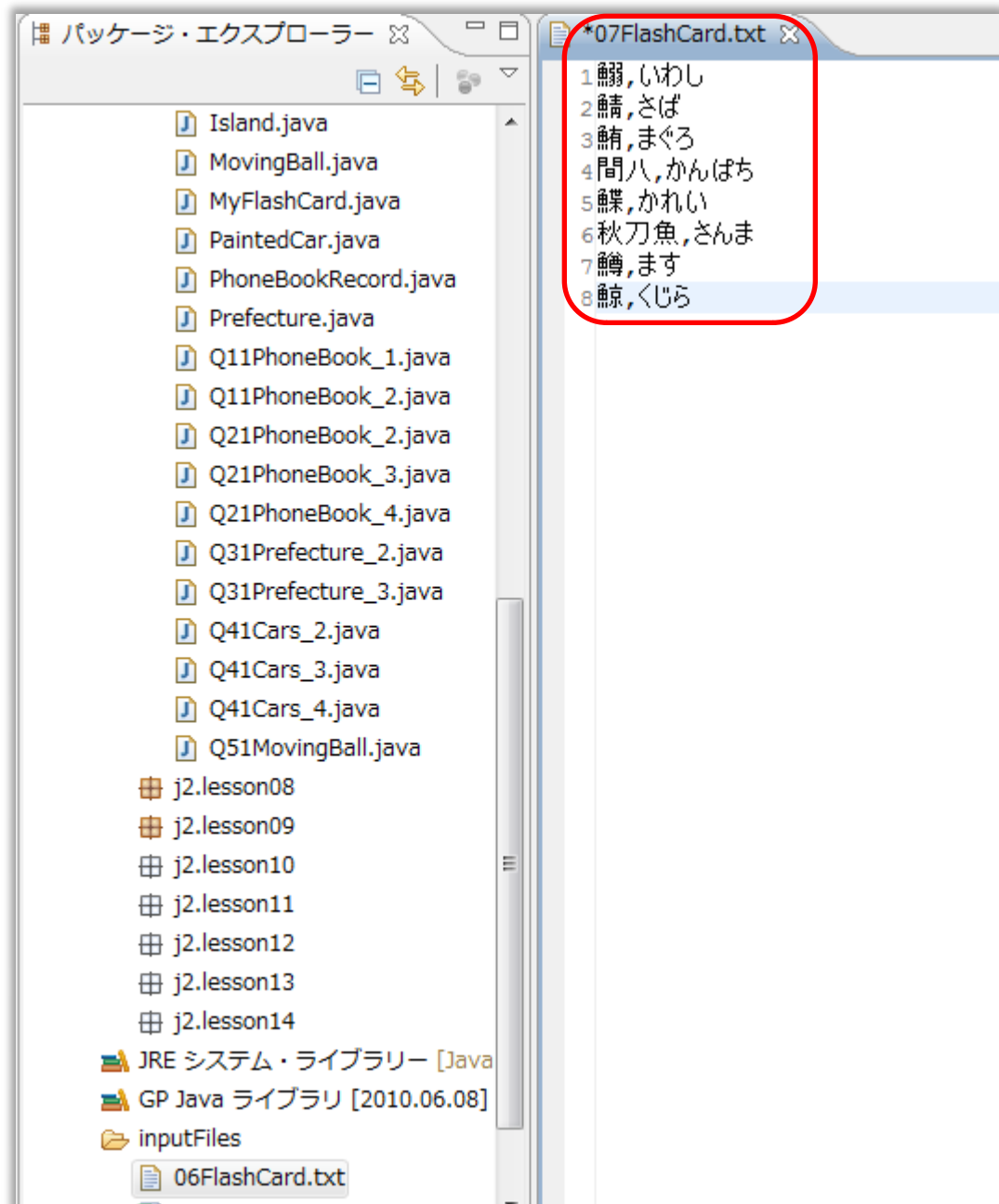
(1) 作業中のプロジェクト(Java2000とか)の  
直下にinputFilesというフォルダを作成する。



## (2) 新規ファイルで、07FlashCard.txtを指定。



(4) 入力が終わったら、  
保存。(Ctrl+S)





## ■ 例題21(2)

問題：例題21(1)で作成したcsvファイルを読み込み、フラッシュカードのデータを保持させよ。1レコード分（1行分＝1カード分）のデータごとに、MyFlashCardクラスのインスタンスを一つ用意して、それぞれのインスタンスは作成した順に配列cardに格納せよ。

## 例題21(2) (Ex21FlashCard\_2)

```
1 package j2.lesson07;
2
3 import java.io.*;
4
5 public class Ex21FlashCard_2 {
6
7     public static void main(String[] args) throws IOException {
8         new Ex21FlashCard_2().start();
9     }
10
11     void start() throws IOException {
12         int maxRecords = 1000;
13         MyFlashCard[] card = new MyFlashCard[maxRecords];
14
15         BufferedReader reader = new BufferedReader(new FileReader(
16             "inputFiles/07FlashCard.txt"));
17         int index = 0;
18         while (index < maxRecords) {
19             String line = reader.readLine();
20             if (line == null) { // null なら終了
21                 break;
22             }
23             String[] strs = line.split(",");
24             String word1 = strs[0]; // 第1列(カード表)
25             String word2 = strs[1]; // 第2列(カード裏)
26
27             card[index] = new MyFlashCard(word1, word2);
28             index++;
29         }
30     }
31 }
```

忘れない

```
Exception in thread "main" java.io.FileNotFoundException: inputFiles#06MovingBallData.txt (  
    at java.io.FileInputStream.open(Native Method)  
    at java.io.FileInputStream.<init>(FileInputStream.java:106)  
    at java.io.FileInputStream.<init>(FileInputStream.java:66)  
    at java.io.FileReader.<init>(FileReader.java:41)  
    at j2.lesson06.Q51MovingBall.main(Q51MovingBall.java:13)
```

コンソールに次のようなメッセージが出て実行がとまってしまった場合、

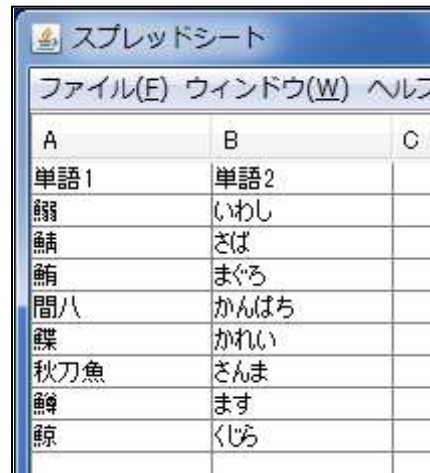
- (1) 作成したCSVファイルの名前、作成した場所が合っているかどうかを確認する。
- (2) プログラム中のオープンするファイル名が間違っていないかどうかを確認する。

## ■ 例題21(3)

問題：例題21(2)で読み込んで保持させたカードの内容をコンソールに表示させよ。

MyFlashCard のインスタンスメソッド

返り値の型	メソッド名(引数)	機能
<i>void</i>	<i>showFields(int row)</i>	インスタンスの内容をSpreadsheet の <i>row</i> 行に表示する。(既に存在する)



A	B	C
単語1	単語2	
鰯	いわし	
鯖	さば	
鮪	まぐろ	
間八	かんぱち	
鰺	かれい	
秋刀魚	さんま	
鰯	ます	
鯨	くじら	

※既存のEx21FlashCard\_2を編集する

# 例題21(3)

## Ex21FlashCard\_3

```
1 package j2.lesson07;
2
3 import gpjava.Spreadsheet;
4
5 import java.io.*;
6
7 public class Ex21FlashCard_3 {
8
9     public static void main(String[] args) throws IOException {
10         new Ex21FlashCard_3().start();
11     }
12
13     void start() throws IOException {
14         int maxRecords = 1000;
15         MyFlashCard[] card = new MyFlashCard[maxRecords];
16
17         BufferedReader reader = new BufferedReader(new FileReader(
18             "inputFiles/07FlashCard.txt"));
19         int index = 0;
20         while (index < maxRecords) {
21             String line = reader.readLine();
22             if (line == null) { // null なら終了
23                 break;
24             }
25             String[] strs = line.split(",");
26             String word1 = strs[0]; // 第1列(カード表)
27             String word2 = strs[1]; // 第2列(カード裏)
28
29             card[index] = new MyFlashCard(word1, word2);
30             index++;
31         }
32         int numRecords = index;
33
34         Spreadsheet.show();
35
36         card[0].header();
37         for (int i = 0; i < numRecords; i++) {
38             card[i].showFields(i + 1);
39         }
40     }
41 }
```

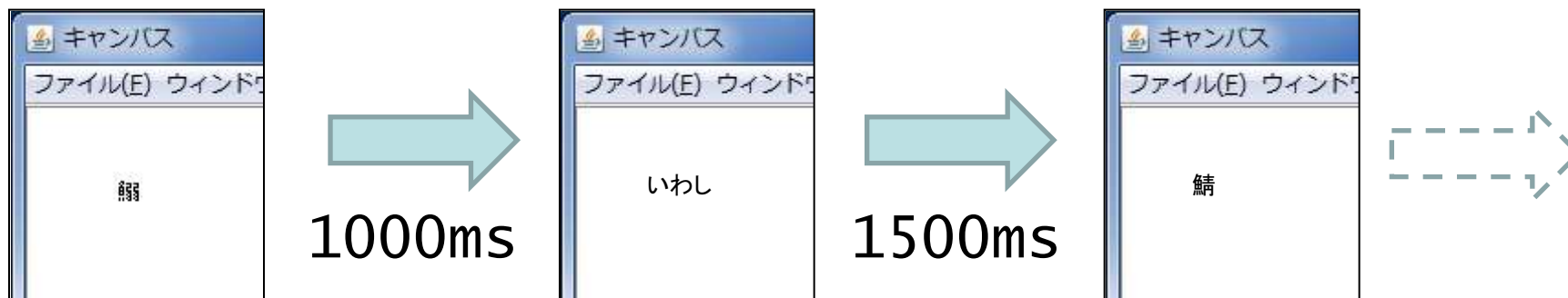
配列に格納した  
データを閲覧する

## ■ 例題21(4)

問題：（例題21(3)の拡張）保持させたカードをもとに、漢字と対応する読みを連続で表示していくプログラムを作成せよ。（表示間隔は例題11(3)と同じ）

### MyFlashCard のインスタンスメソッド

返り値の型	メソッド名(引数)	機能
void	showCard()	引数aCardのword1、word2の内容を例題11(3)のようにキャンバスに描画する。(既に存在する)



※既存のEx21FlashCard\_3を編集する

# 例題21(4)

Ex21FlashCard\_4

```
1 package j2.lesson07;
2
3 import java.io.*;
4 import gpjava.Canvas;
5
6 public class Ex21FlashCard_4 {
7
8     public static void main(String[] args) throws IOException {
9         new Ex21FlashCard_4().start();
10    }
11
12    void start() throws IOException {
13        int maxRecords = 1000;
14        MyFlashCard[] card = new MyFlashCard[maxRecords];
15
16        BufferedReader reader = new BufferedReader(new FileReader(
17            "inputFiles/07FlashCard.txt"));
18        int index = 0;
19        while (index < maxRecords) {
20            String line = reader.readLine();
21            if (line == null) { // null なら終了
22                break;
23            }
24            String[] strs = line.split(",");
25            String word1 = strs[0]; // 第1列(カード表)
26            String word2 = strs[1]; // 第2列(カード裏)
27
28            card[index] = new MyFlashCard(word1, word2);
29            index++;
30        }
31        int numRecords = index;
32
33        Canvas.show();
34
35        for (int i = 0; i < numRecords; i++) {
36            card[i].showCard();
37        }
38    }
39 }
```

## ■ 例題30

問題：次のクラスIslandを作成せよ。(第4回例題10参照)

クラスIslandのインスタンスは次のインスタンス変数を持ち、「島」の情報を保持する。(初期値は指定しなくて良い)

インスタンス変数	初期値	説明
String name	無し	島の名前
double area	無し	島の面積( $km^2$ )

(クラス名: Island)



## 例題30

```
1 package j2.lesson07;
2
3 import gpjava.Spreadsheet;
4 import gpjava.Canvas;
5
6 public class Island {
7     String name; // 島の名前
8     double area; // 面積
9
10    Island(String name, double area) {
11        this.name = name;
12        this.area = area;
13    }
14 }
```

クラスIsland  
初期値は無し

name:
area:

インスタンス変数	初期値	説明
String name	無し	島の名前
double area	無し	島の面積( $km^2$ )

## ■ 例題31(1)

問題: クラスIslandのインスタンスを4個作成するプログラムを作成せよ。ただし、それぞれのインスタンスはIsland型の配列に格納するようにせよ。

### Island のコンストラクタ

メソッド名(引数)	機能
Island(String name, double area)	各インスタンス変数の値がそれぞれnameとareaであるIslandのインスタンスを作成する。

[0]	[1]	[2]	[3] island
名前: 沖縄島 面積: 1206.49	名前: 淡路島 面積: 592.17	名前: 伊豆大島 面積: 91.06	名前: 与那国島 面積: 28.91

(クラス名: Ex31Island\_1)

## 例題31(1)

```
1 package j2.lesson07;
2
3 public class Ex31Island_1 {
4     public static void main(String[] args) {
5         new Ex31Island_1().start();
6     }
7
8     void start() {
9         int numIsland = 4;
10        Island [] island = new Island [numIsland];
11        island[0] = new Island("沖縄島", 1206.49);
12        island[1] = new Island("淡路島", 592.17);
13        island[2] = new Island("伊豆大島", 91.06);
14        island[3] = new Island("与那国島", 28.91);
15    }
16 }
```

## ■ 例題31(2)

問題：（例題31(1)の拡張）作成したIslandクラスのインスタンスの内容を表示せよ。

Island のインスタンスメソッド

返り値の型	メソッド名(引数)	機能
void	showIslandData(int row)	引数iの各インスタンス変数の値を Spreadsheet の row行に表示する。
void	header()	Spreadsheetのヘッダ行を表示する。



スプレッドシート				
ファイル(E) ウィンドウ(W) ヘルプ(H)				
A	B	C	D	E
名前	面積			
沖縄島	1206.49km2			
淡路島	592.17km2			
伊豆大島	91.06km2			
与那国島	28.91km2			

※既存のEx31Island\_1を編集する  
(クラス名: Ex31Island\_2)

## 例題31(2)

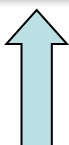
```
1 package j2.lesson07;
2
3 import gpjava.Spreadsheet;
4
5 public class Ex31Island_2 {
6     public static void main(String[] args) {
7         new Ex31Island_2().start();
8     }
9
10    void start() {
11        int numIsland = 4;
12        Island [] island = new Island [numIsland];
13        island[0] = new Island("沖繩島", 1206.49);
14        island[1] = new Island("淡路島", 592.17);
15        island[2] = new Island("伊豆大島", 91.06);
16        island[3] = new Island("与那国島", 28.91);
17
18        Spreadsheet.show();
19        island[0].header();
20        for (int i = 0; i < numIsland; i++) {
21            island[i].showIslandData(i + 1);
22        }
23    }
24 }
```

## 例題31(2) Island

```
1  package j2.lesson07;
2
3  import gpjava.Spreadsheet;
4
5  public class Island {
6      String name; // 島の名前
7      double area; // 面積
8
9      Island(String name, double area){
10         this.name = name;
11         this.area = area;
12     }
13
14     // Ex31_2
15     void showIslandData(int row) {
16         Spreadsheet.setString(row, 0, name);
17         Spreadsheet.setString(row, 1, area + "km2");
18     }
19     // Ex31_2
20     void header() {
21         Spreadsheet.setString(0, 0, "名前");
22         Spreadsheet.setString(0, 1, "面積");
23     }
24 }
```

## 例題31(2)

```
10 void start() {  
11     int numIsland = 4;  
12     Island [] island = new Island [numIsland];  
13     island[0] = new Island("沖縄島", 1206.49);  
14     island[1] = new Island("淡路島", 592.17);  
15     island[2] = new Island("伊豆大島", 91.06);  
16     island[3] = new Island("与那国島", 28.91);  
17  
18     Spreadsheet.show();  
19     island[0].header();  
20     for (int i = 0; i < numIsland; i++) {  
21         island[i].showIslandData(i + 1);  
22     }  
23 }
```



```
Island[0].showIslandData(1);  
Island[0].showIslandData(1);  
Island[0].showIslandData(1);  
Island[0].showIslandData(1);
```

配列を用いることで、for文のインデックス*i*を用いて配列islandに格納したインスタンスを参照している。

なお、配列の要素数に変更があったとしても、(例えば、登録したいデータが増えたとしても)、この部分を変更する必要は無い。

## ■ 例題31(3)

問題：(例題31(2)の拡張)作成したIslandクラスのインスタンスの情報をもとにして、「島」の名前と、面積に比例する大きさの正方形で表示させよ。

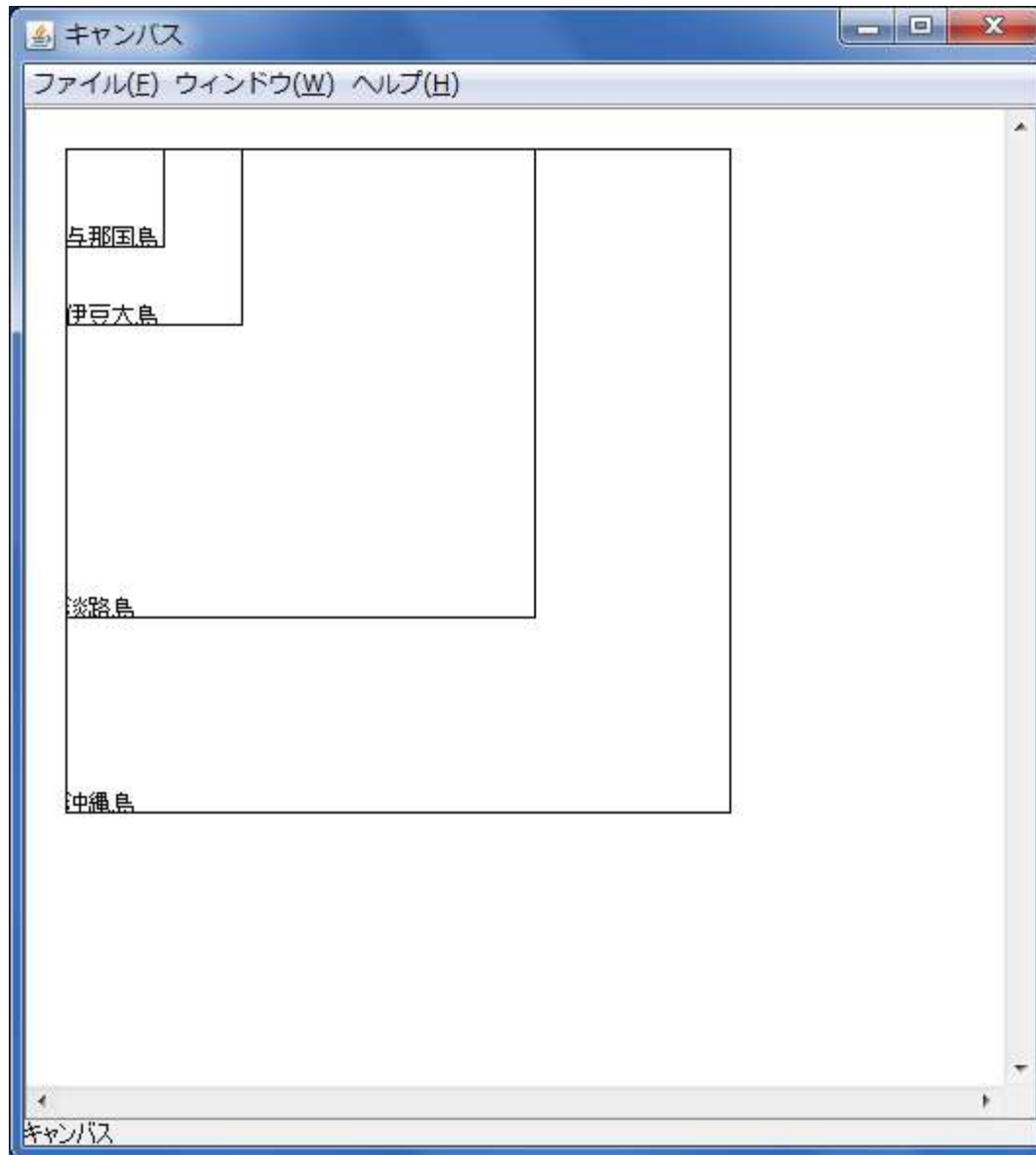
Island のインスタンスメソッド

返り値の型	メソッド名(引数)	機能
void	drawIslandSquare()	キャンバスに「島」の名前と、面積に比例する大きさの正方形を描画する。(前回例題11(3)を参照)

※既存のEx31Islandを編集する  
(クラス名: Ex31Island\_3)



# 例題31(3) 実行例



## 例題31(3)

```
1 package j2.lesson07;
2
3 import gpjava.Canvas;
4
5 public class Ex31Island_3 {
6     public static void main(String[] args) {
7         new Ex31Island_3().start();
8     }
9
10    void start() {
11        int numIsland = 4;
12        Island [] island = new Island [numIsland];
13        island[0] = new Island("沖繩島", 1206.49);
14        island[1] = new Island("淡路島", 592.17);
15        island[2] = new Island("伊豆大島", 91.06);
16        island[3] = new Island("与那国島", 28.91);
17
18        Canvas.show();
19        for (int i = 0; i < numIsland; i++) {
20            island[i].drawIslandSquare();
21        }
22    }
23 }
```

## 例題31(3) Island

```
1 package j2.lesson07;
2
3 import gpjava.Spreadsheet;
4 import gpjava.Canvas;
5
6 public class Island {
7     String name; // 島の名前
8     double area; // 面積
9
10    Island(String name, double area){
11        this.name = name;
12        this.area = area;
13    }
14
15    // Ex31_2
16    void showIslandData(int row) {
17        Spreadsheet.setString(row, 0, name);
18        Spreadsheet.setString(row, 1, area + "km2");
19    }
20    // Ex31_2
21    void header() {
22        Spreadsheet.setString(0, 0, "名前");
23        Spreadsheet.setString(0, 1, "面積");
24    }
25
26    // Ex31_3
27    void drawIslandSquare() {
28        double length = Math.sqrt(area);
29        int pixel = (int)length * 10;
30        Canvas.drawRect(20, 20, pixel, pixel);
31        Canvas.drawString(20, 20+pixel, name);
32    }
33 }
```

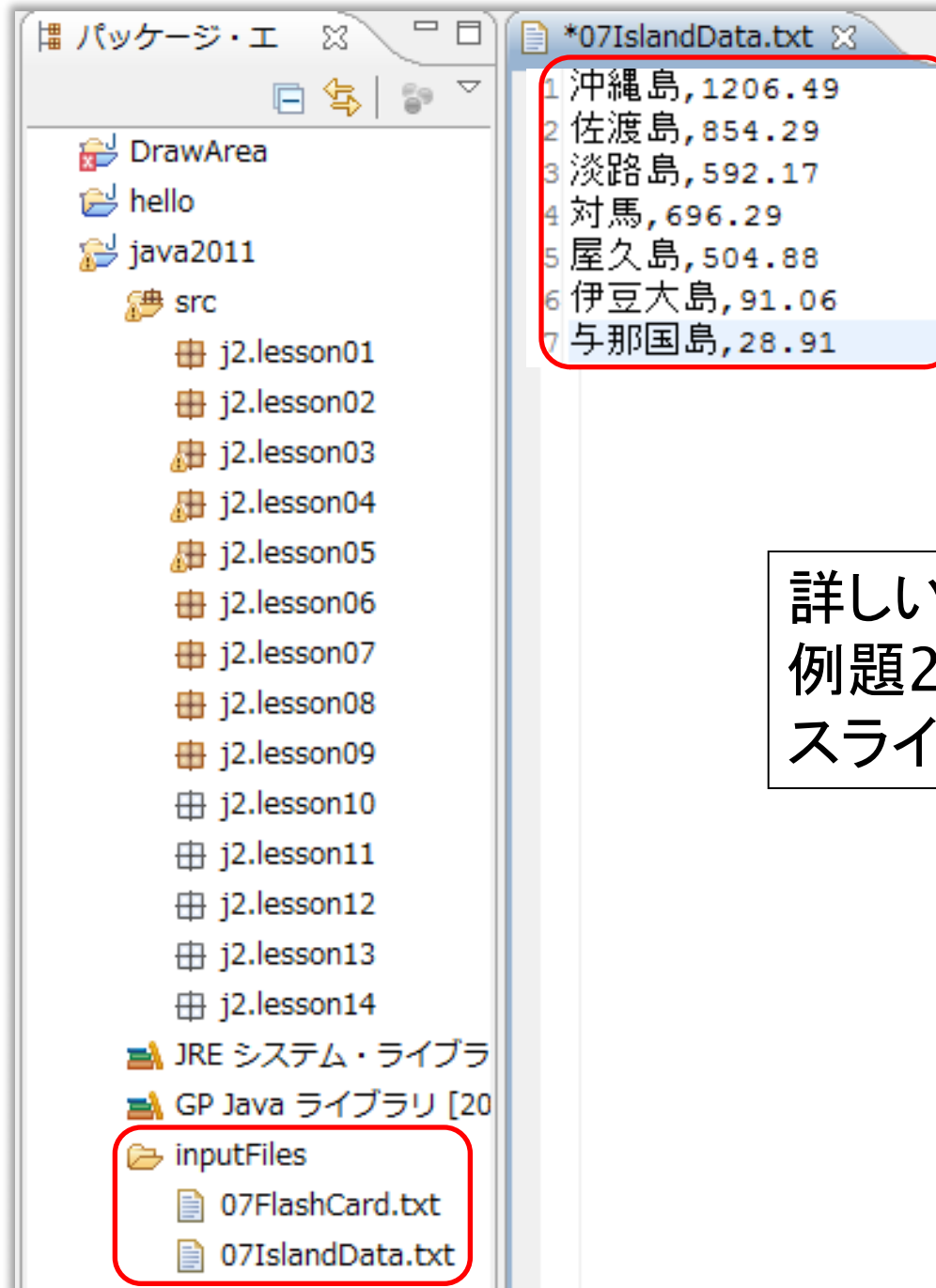
## ■ 例題41(1)

問題：島のデータを記述した次のcsvファイルを作成せよ。(フォルダinputFilesの直下に作成する。)

```
沖縄島,1206.49  
佐渡島,854.29  
淡路島,592.17  
対馬,696.29  
屋久島,504.88  
伊豆大島,91.06  
与那国島,28.91
```

(ファイル名:07IslandData.txt)

# 例題41(1)



詳しい作成方法は  
例題21(1)の  
スライドに有り

## ■ 例題41(2)

問題：例題41(1)で作成したcsvファイルを読み込み、島のデータをIsland型の配列islandに格納するようにせよ。1レコード分(1行分)のデータごとに、Islandクラスのインスタンスを一つ用意して、作成した順に配列に格納せよ。

### Island のコンストラクタ

メソッド名(引数)	機能
Island(String name, double area)	各インスタンス変数の値がそれぞれnameとareaであるIslandのインスタンスを作成する。

(クラス名: Ex41Island\_2)

## 例題41(2)

忘れない

```
1 package j2.lesson07;
2
3 import java.io.*;
4
5 public class Ex41Island_2 {
6     public static void main(String[] args) throws IOException {
7         new Ex41Island_2().start();
8     }
9
10    void start() throws IOException {
11        int maxRecords = 1000; // ファイルから読み込める最大レコード数
12        int index = 0;
13        Island[] island = new Island[maxRecords];
14
15        // データの読み込み
16        BufferedReader reader = new BufferedReader(new FileReader(
17            "inputFiles/07IslandData.txt"));
18        while (index < maxRecords) {
19            String line = reader.readLine();
20            if (line == null) { // null なら終了
21                break;
22            }
23            String[] strs = line.split(",");
24
25            String name = strs[0]; // 第1列(島の名前)
26            double area = Double.parseDouble(strs[1]); // 第2列(面積)
27
28            island[index] = new Island(name, area);
29            index++;
30        }
31    }
32 }
```

## ■ 例題41(3)

問題：例題41(2)で読み込んで保持させたデータを、実行例を参考に表示させよ。

### Island のインスタンスメソッド

返り値の型	メソッド名(引数)	機能
void	showIslandData(int row)	引数iの各インスタンス変数の値を Spreadsheet の row行に表示する。 (既に存在する)



A	B	C	D	E
名前	面積			
沖縄島	1206.49km <sup>2</sup>			
佐渡島	854.29km <sup>2</sup>			
淡路島	592.17km <sup>2</sup>			
対馬	696.29km <sup>2</sup>			
屋久島	504.88km <sup>2</sup>			
伊豆大島	91.06km <sup>2</sup>			
与那国島	28.91km <sup>2</sup>			

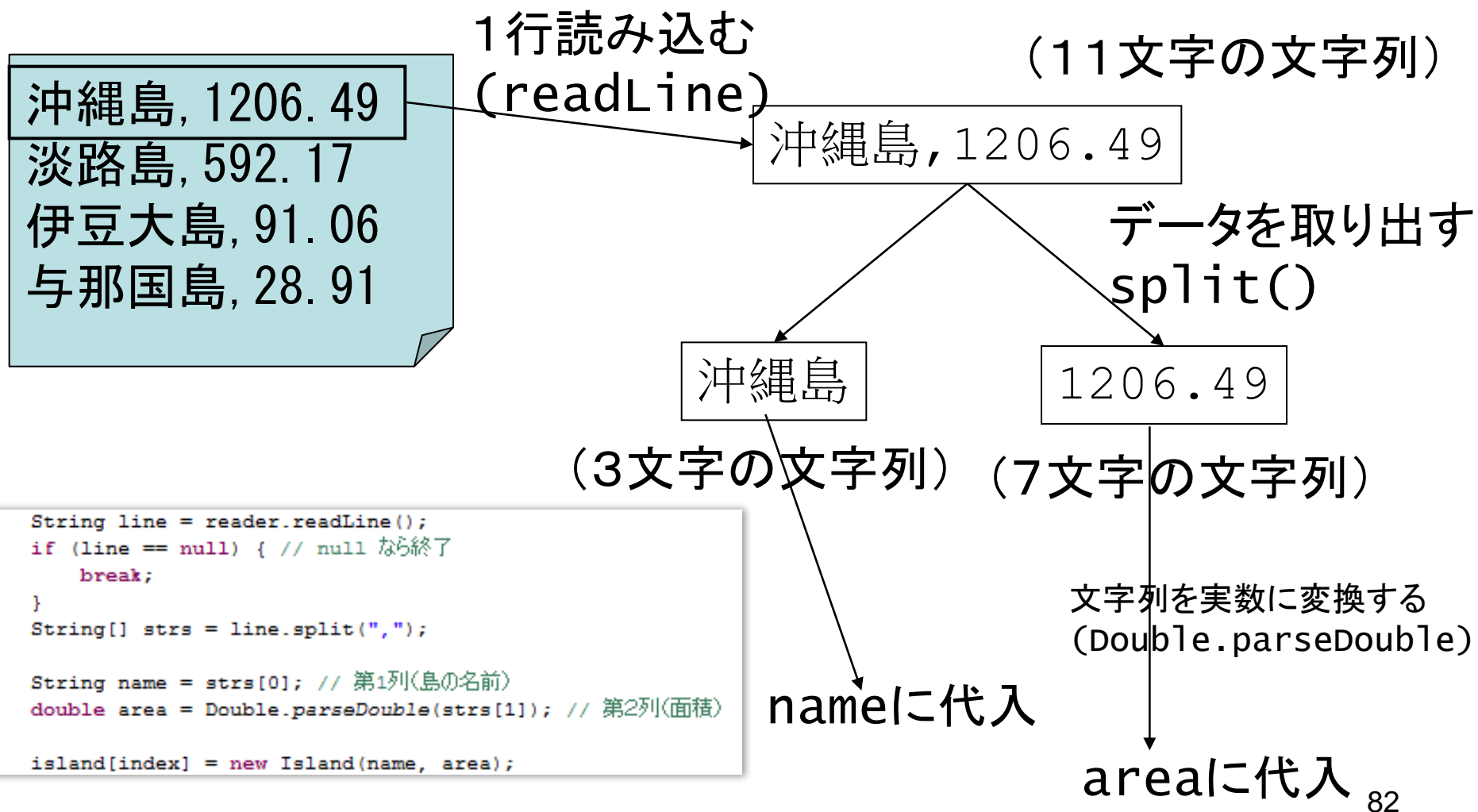
※既存のEx41Island\_2を編集する



## 例題41(3)

```
1 package j2.lesson07;
2
3 import java.io.*;
4 import gpjava.Spreadsheet;
5
6 public class Ex41Island_3 {
7     public static void main(String[] args) throws IOException {
8         new Ex41Island_3().start();
9     }
10
11     void start() throws IOException {
12         int maxRecords = 1000; // ファイルから読み込める最大レコード数
13         int index = 0;
14         Island[] island = new Island[maxRecords];
15
16         // データの読み込み
17         BufferedReader reader = new BufferedReader(new FileReader(
18             "inputFiles/07IslandData.txt"));
19         while (index < maxRecords) {
20             String line = reader.readLine();
21             if (line == null) { // null なら終了
22                 break;
23             }
24             String[] strs = line.split(",");
25
26             String name = strs[0]; // 第1列(島の名前)
27             double area = Double.parseDouble(strs[1]); // 第2列(面積)
28
29             island[index] = new Island(name, area);
30             index++;
31         }
32         int numRecords = index; // 実際に読み込んだレコード数
33
34         Spreadsheet.show();
35         island[0].header();
36         // 読み込んだデータの利用
37         for(int i = 0; i < numRecords; i++) {
38             island[i].showIslandData(i + 1);
39         }
40     }
41 }
```

# 行から島の名前と面積のデータを切り出す



## ■ 例題41(4)

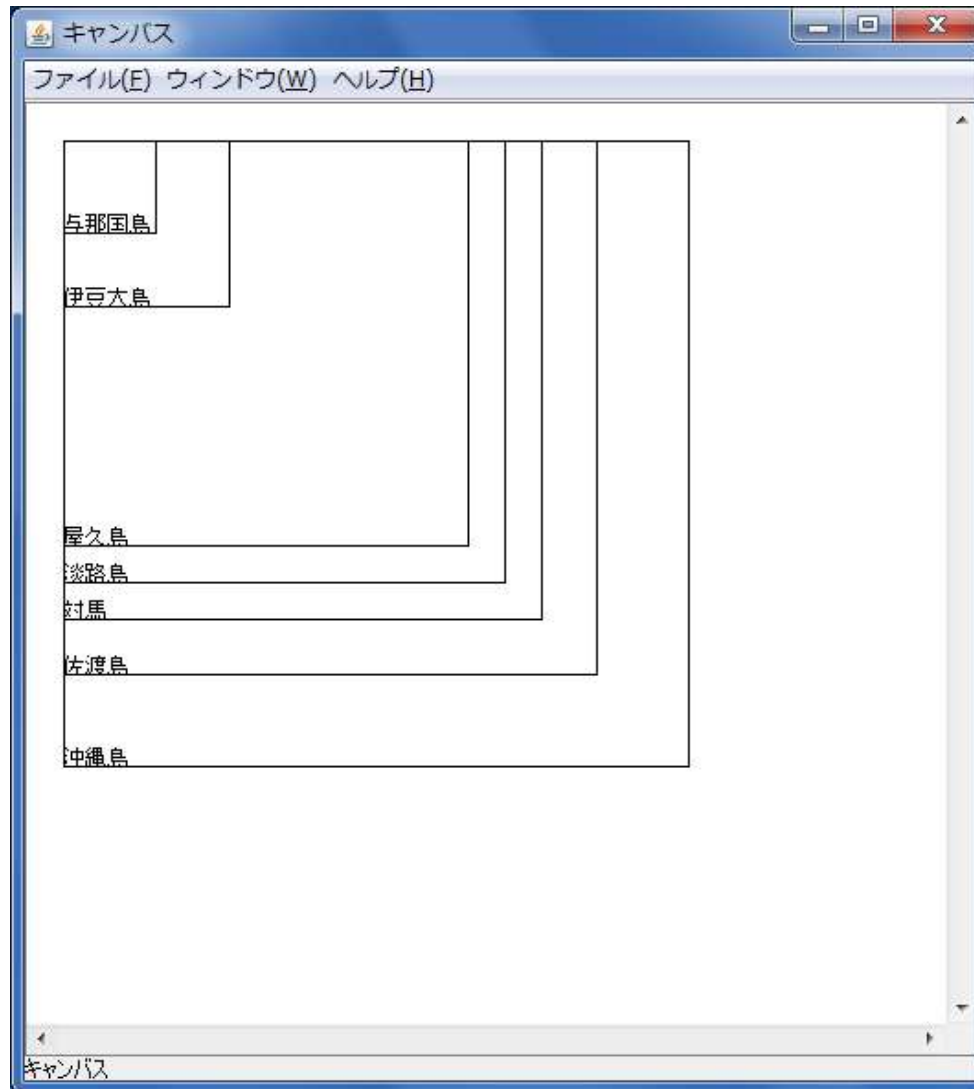
問題: 例題41(3)に加え、保持させた島のデータをもとに、島の名前と、島の面積に比例する大きさの正方形を表示させよ。

Island のインスタンスメソッド

返り値の型	メソッド名(引数)	機能
void	drawIslandSquare()	キャンバスに「島」の名前と、面積に比例する大きさの正方形を描画する。(例題31(3)を参照)

※既存のEx41Island\_3を編集する

# 例題41(4) 実行例



## 例題41(4)

```
1 package j2.lesson07;
2
3 import java.io.*;
4 import gpjava.Canvas;
5
6 public class Ex41Island_4 {
7     public static void main(String[] args) throws IOException {
8         new Ex41Island_4().start();
9     }
10
11     void start() throws IOException {
12         int maxRecords = 1000; // ファイルから読み込める最大レコード数
13         int index = 0;
14         Island[] island = new Island[maxRecords];
15
16         // データの読み込み
17         BufferedReader reader = new BufferedReader(new FileReader(
18             "inputFiles/07IslandData.txt"));
19         while (index < maxRecords) {
20             String line = reader.readLine();
21             if (line == null) { // null なら終了
22                 break;
23             }
24             String[] strs = line.split(",");
25
26             String name = strs[0]; // 第1列(島の名前)
27             double area = Double.parseDouble(strs[1]); // 第2列(面積)
28
29             island[index] = new Island(name, area);
30             index++;
31         }
32         int numRecords = index; // 実際に読み込んだレコード数
33
34         // 読み込んだデータの利用
35         Canvas.show();
36         for (int i = 0; i < numRecords; i++) {
37             island[i].drawIslandSquare();
38         }
39     }
40 }
```

## ■ 例題41(5) (opt)

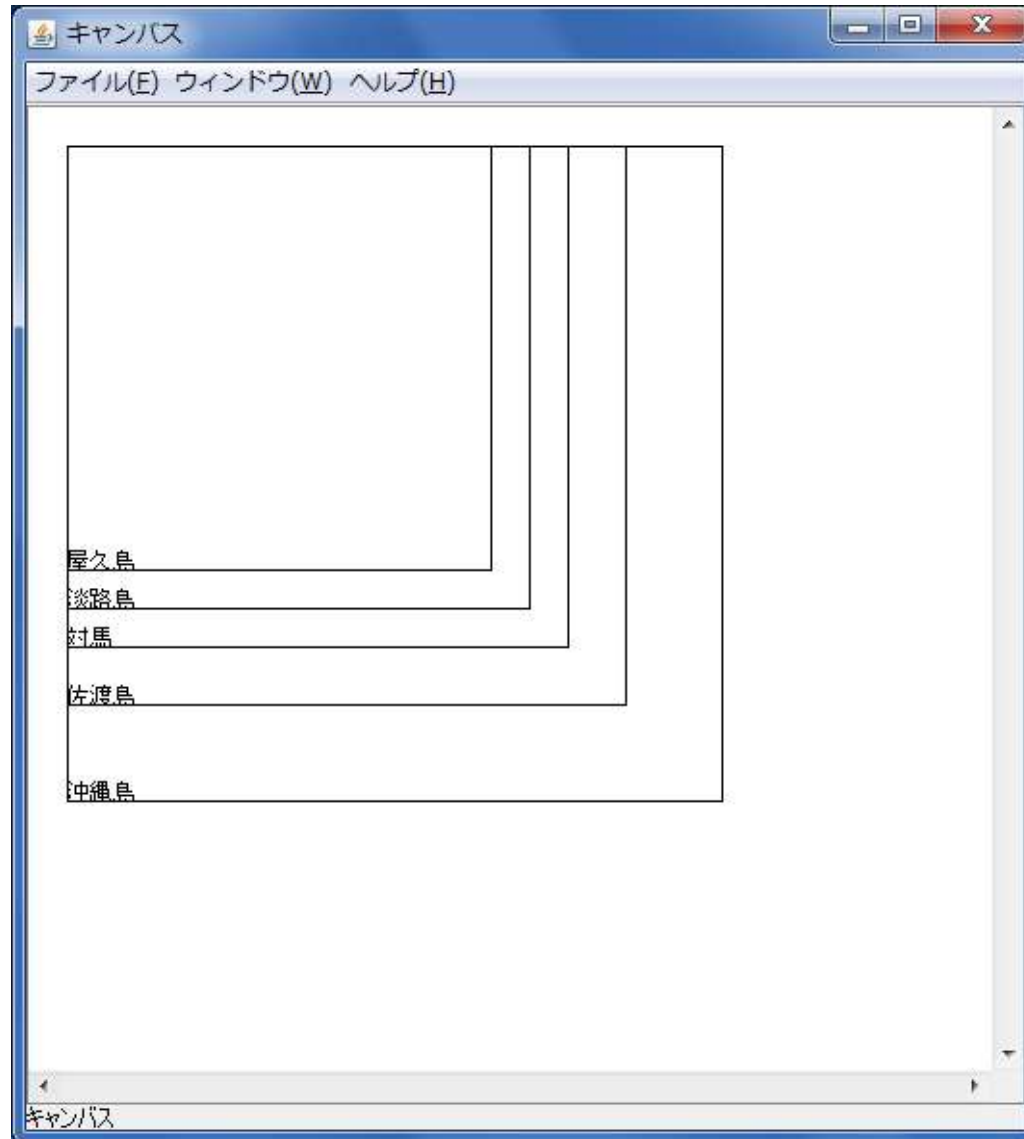
問題: 例題41(4)で、島の面積が500平方km以上のもののデータを島の名前と正方形として表示させよ。

### Island のインスタンスメソッド

返り値の型	メソッド名(引数)	機能
void	drawIslandSquare()	キャンバスに「島」の名前と、面積に比例する大きさの正方形を描画する。(例題31(3)を参照)

※既存のEx41Island\_4を編集する

# 例題41(5) 実行例



## 例題41(5)

指定した条件  
(面積 $\geq 500.0$ )  
を満たす島データ  
のみ情報を表示

```
1 package j2.lesson07;
2
3 import java.io.*;
4 import gpjava.Canvas;
5
6 public class Ex41Island_5 {
7     public static void main(String[] args) throws IOException {
8         new Ex41Island_5().start();
9     }
10
11     void start() throws IOException {
12         int maxRecords = 1000; // ファイルから読み込める最大レコード数
13         int index = 0;
14         Island[] island = new Island[maxRecords];
15
16         // データの読み込み
17         BufferedReader reader = new BufferedReader(new FileReader(
18             "inputFiles/07IslandData.txt"));
19         while (index < maxRecords) {
20             String line = reader.readLine();
21             if (line == null) { // null なら終了
22                 break;
23             }
24             String[] strs = line.split(",");
25
26             String name = strs[0]; // 第1列(島の名前)
27             double area = Double.parseDouble(strs[1]); // 第2列(面積)
28
29             island[index] = new Island(name, area);
30             index++;
31         }
32         int numRecords = index; // 実際に読み込んだレコード数
33
34         // 読み込んだデータの利用
35         Canvas.show();
36         for (int i = 0; i < numRecords; i++) {
37             if (island[i].area >= 500.0) {
38                 island[i].drawIslandSquare();
39             }
40         }
41     }
42 }
```