プログラミング入門2

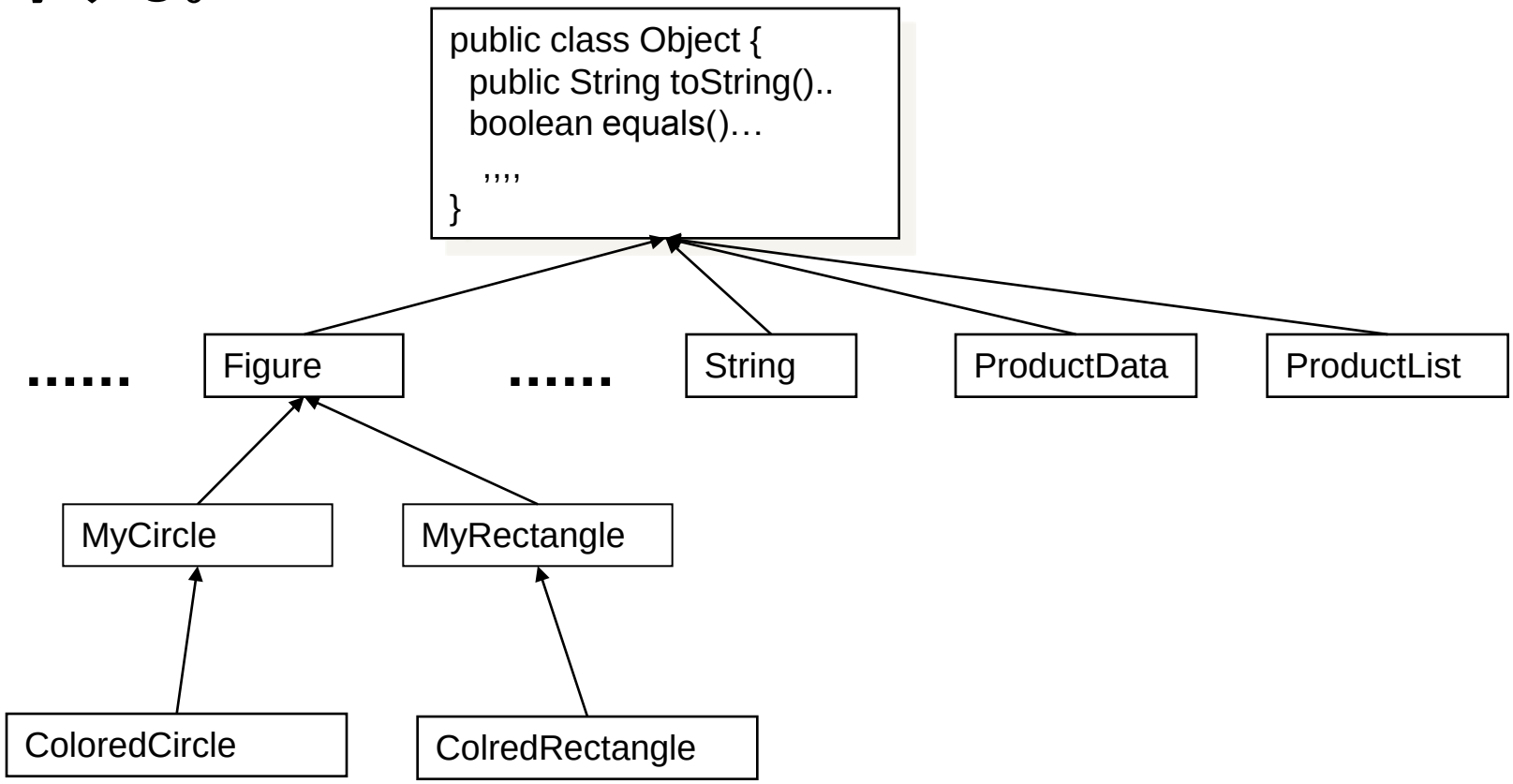
第13回　ArrayList と HashMap

テーマ : ArrayList と HashMap

- ラッパオブジェクト
  - 基本型と参照型
- ArrayList
- HashMap

前回の復習: クラスObject

Java言語では、階層構造のルート（根）に、クラスObjectがある。
スーパークラスを指定しないクラスも、暗黙にクラスObjectを継承する。

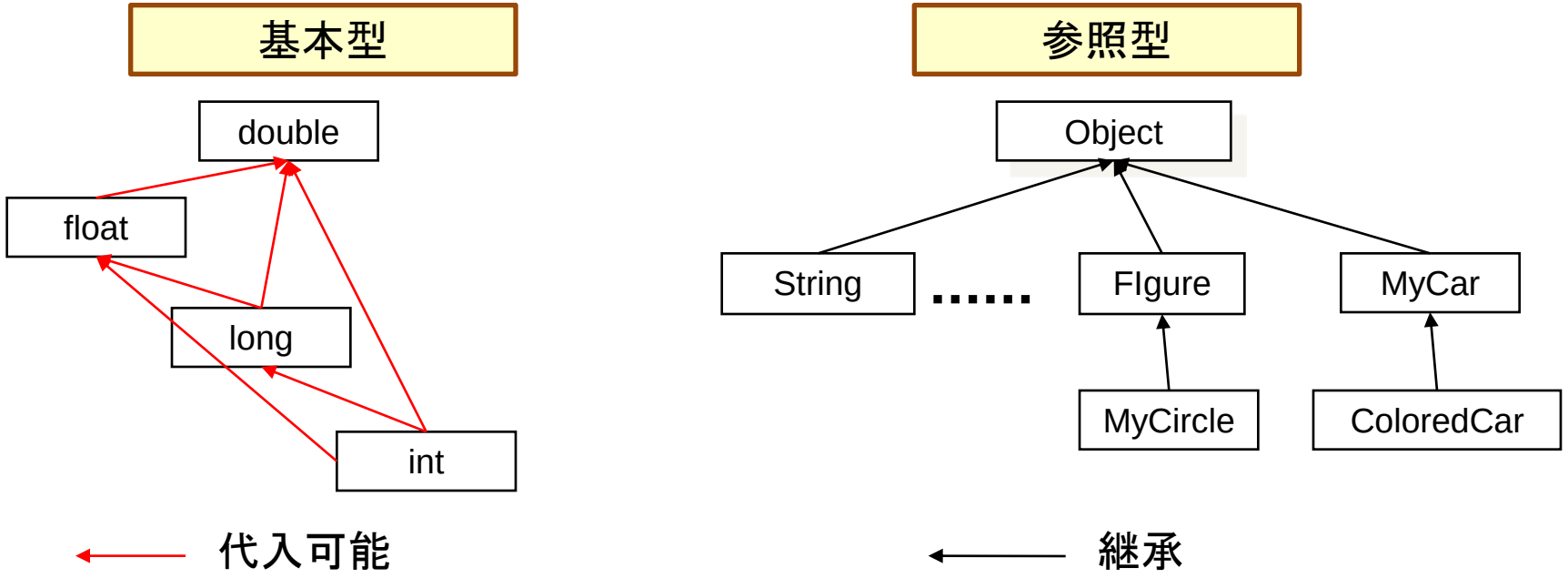


テーマ：ArrayList と HashMap

- ラッパオブジェクト
  - 基本型と参照型
- ArrayList
- HashMap

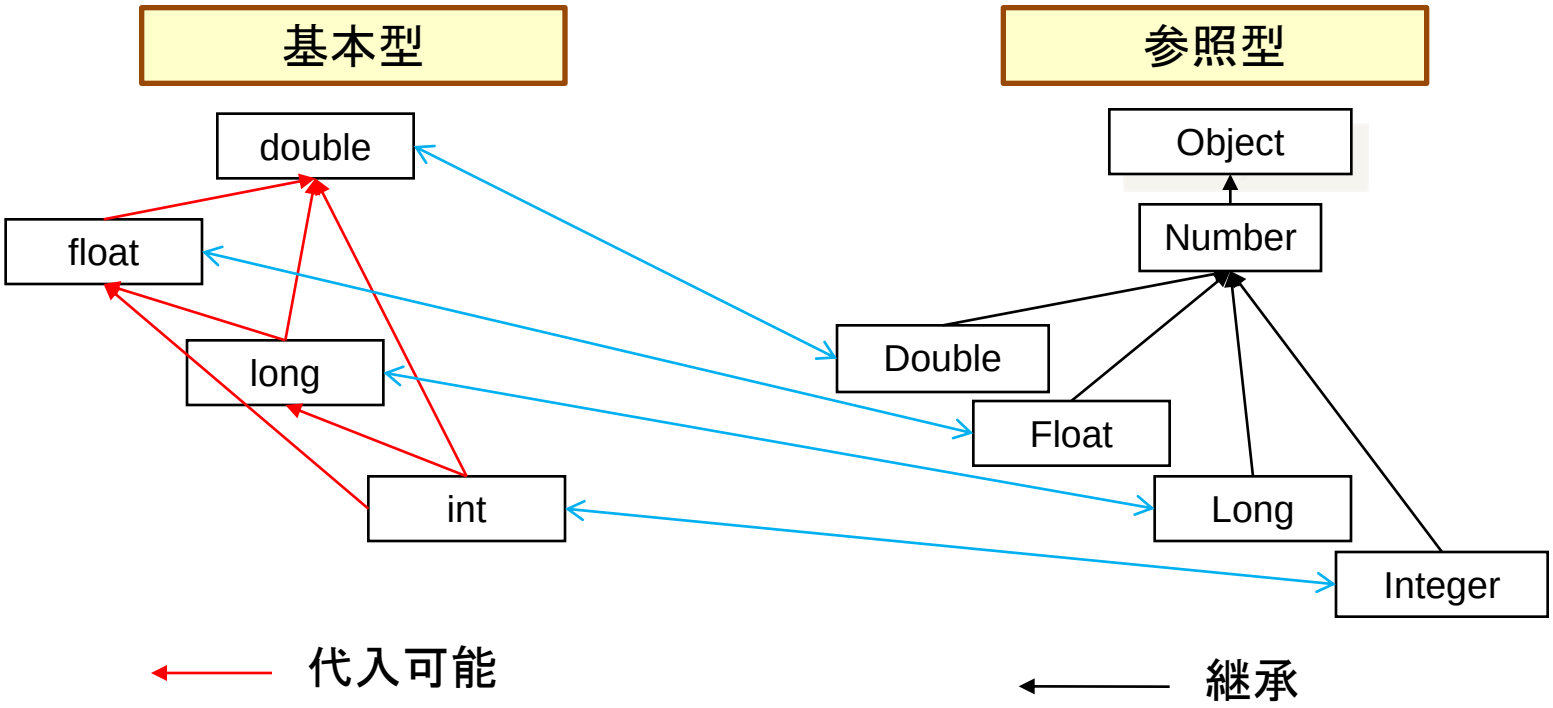
基本型と参照型

- Java には、int, long, double, char, boolean などのオブジェクトではない数値・論理値型が存在する。
- 基本型はオブジェクトの性質を持たないため、他のクラスとは相いれない。



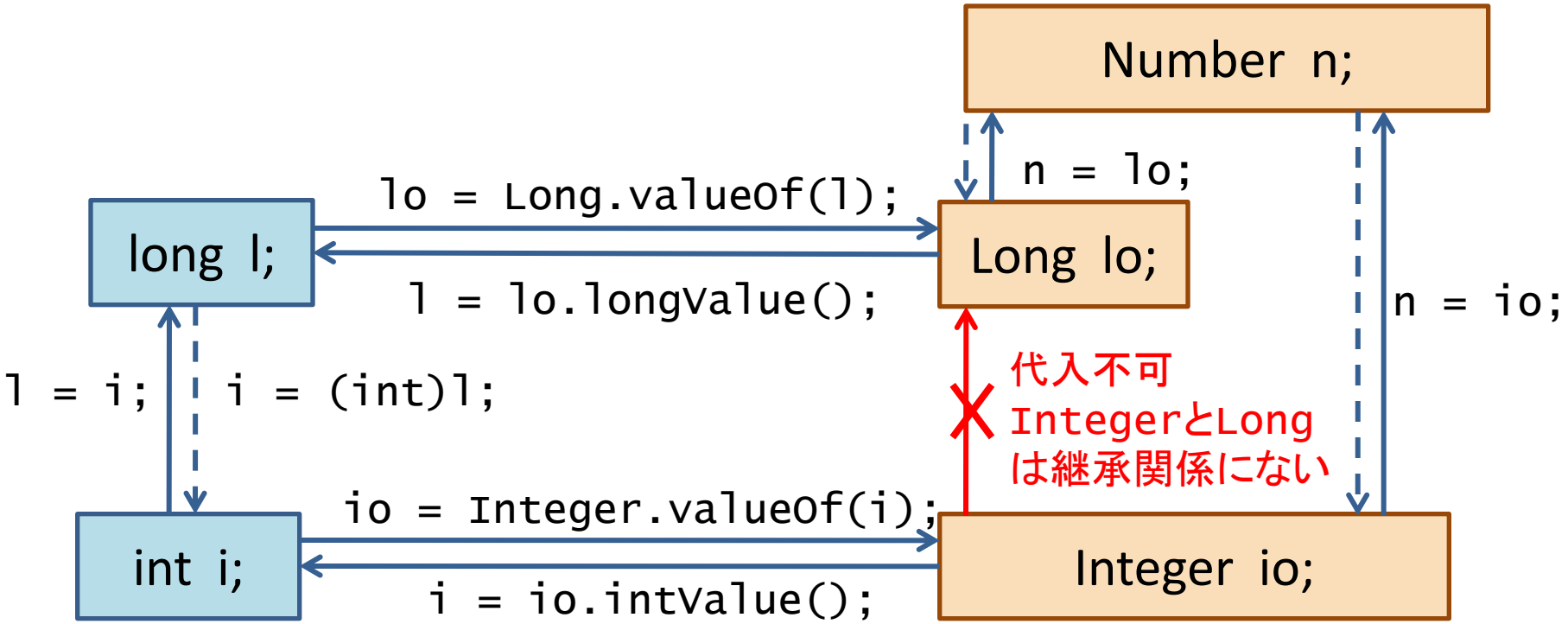
基本型とラッパクラス

- Java には、int, long, double, char, boolean などのデータを内部に保持したラッパクラスが存在する。
- 基本型に対して、1対のラッパが存在する。



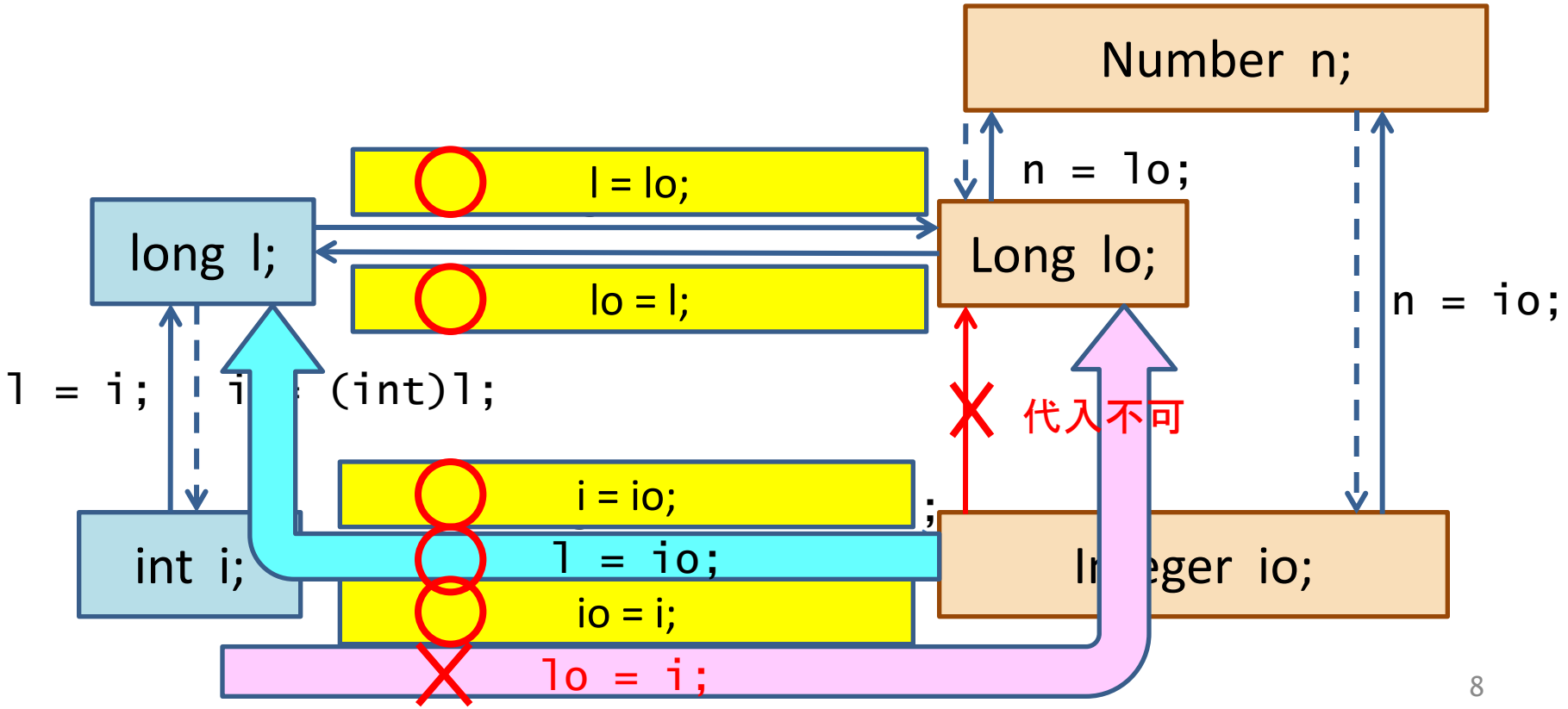
基本的な型変換

- キャストや、型変換メソッドを用いた確実で明示的な型変換の関係



暗黙の自動型変換(Auto Boxing)

- Java 5.0 以降に認められた基本型とラッパクラス間の自動型変換



AutoBoxing の落とし穴

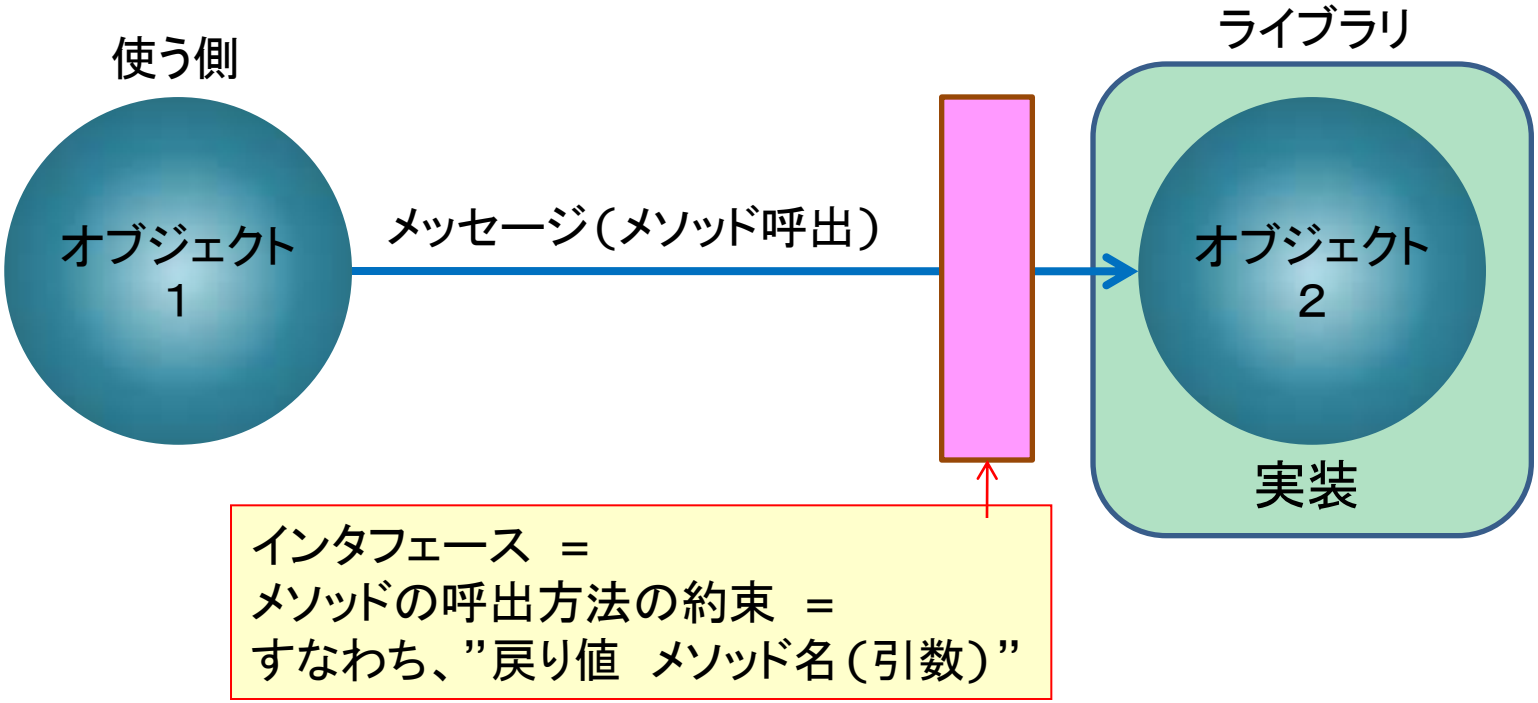
- AutoBoxingは、プログラムを書く上で便利ではあるが、課題も多く残す。
  - ラッパクラスでは、== による等号が常には成り立たない。equals を使う方が安全。
  - Long 変数に、int の値は代入できない。
    - int →Integer - ✕ → Long
    - 理由: Integer と Long は、継承関係のないクラスであり、互いに代入不可。

テーマ：ArrayList と HashMap

- ラッパオブジェクト
  - 基本型と参照型
- ArrayList
- HashMap

オブジェクト指向言語の利点
カプセル化

- オブジェクト指向言語は、「クラス」や「パッケージ」によって、内部の実装を隠ぺいしたライブラリ(役立つツール群)を提供しやすい。



Java クラスライブラリ (Java API)

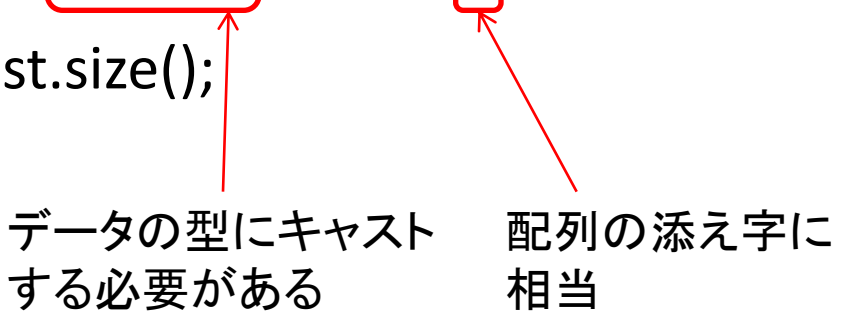
- Javaは標準的な機能として、いくつもの機能をクラスライブラリとして提供している。Java API (Application Programming Interface) と呼ばれることもある。
  - http://java.sun.com/javase/ja/6/docs/ja/api/index.html
- すでに講義で扱ったクラスライブラリの例
  - コンソールへ文字を出力する際には System クラスを用いて System.out.println メソッドを使用している。
    - System クラスのクラス変数 out に対して、println というインスタンスメソッドを呼び出している
  - sqrt や sin などの数学関数を使用するには Math クラスの Math.sqrt や Math.sin というクラスメソッドを使用している。
  - ダイアログの表示では、JOptionPaneクラスのshowMessageDialog,showInputDialogを使用してきた。

Javaクラスライブラリのパッケージの例

パッケージ名	このパッケージに含まれるクラス	例
java.lang	プログラムを作成する際に使用される、基本的なクラス	System, String, Math
java.io	入出力 (I/O) に関するクラス	File, FileReader, FileWriter
java.net	ネットワークに関するクラス	URL, HttpURLConnection
java.util	ユーティリティクラス	Calendar, Random, List
java.awt	Abstract Window Toolkit。GUIを作る際に使われるクラス	Window, Dialog, Graphics

可変長配列　ArrayList

- java.util.ArrayList として提供される。
- 配列の長さを最初に決めないで、途中で変えたい場合に便利。
- (初期化)　　ArrayList list = new ArrayList();
- (要素追加)　　list.add(“ほげほげ”);
- (参照)　　String text = (String) list.get(0);
- (サイズ計算)　　int size = list.size();



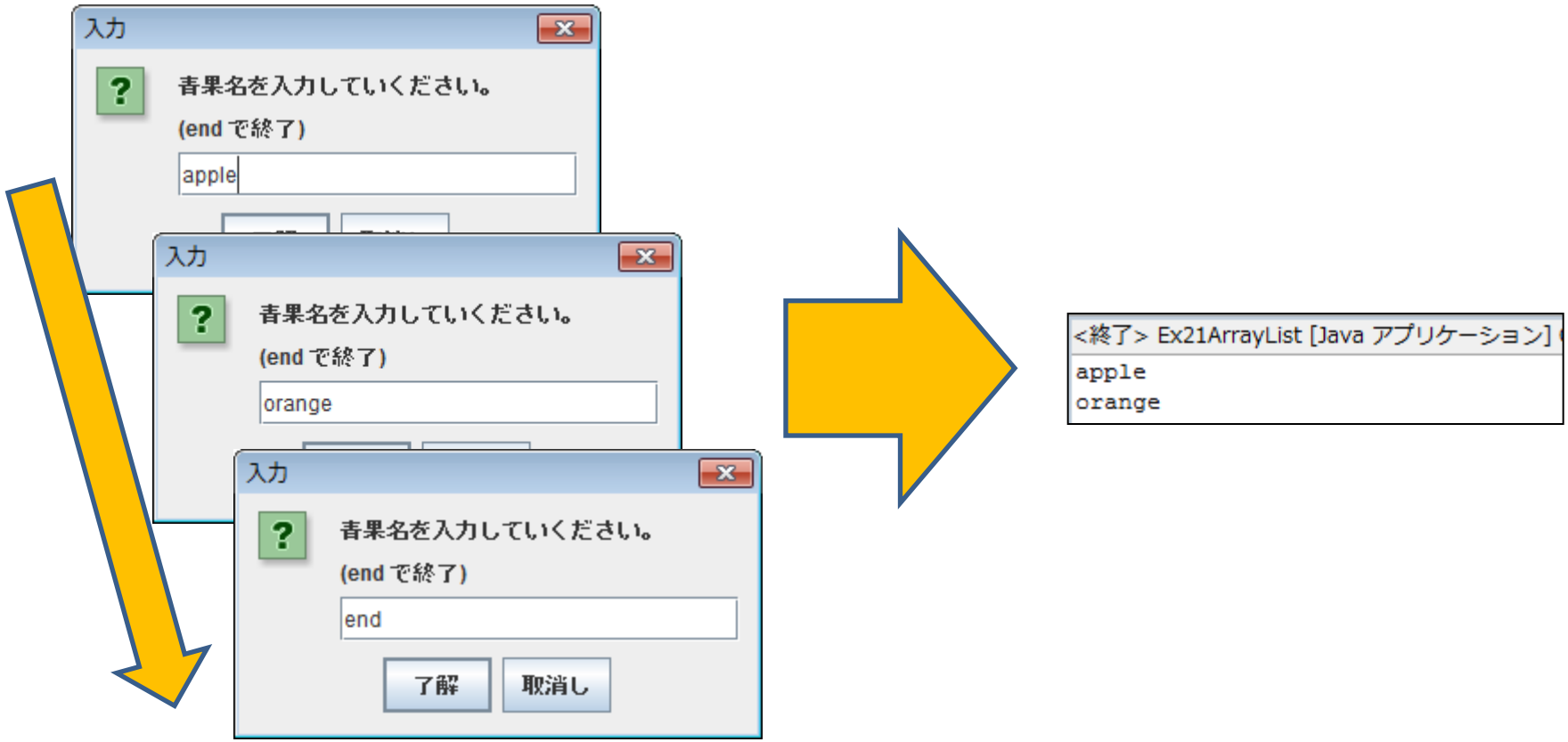
配列とArrayList の比較

- ArrayList は、内部の配列の長さが足りない場合には「配列の長さを自動的に拡張する処理」を自動的に行ってくれる。
- 配列は、最初に大きさの宣言が必要で、途中で変えられない。

内容	**配列**	**ArrayList**
値を代入	array[i] = obj	list.set(i, obj)
値を参照	obj = array[i]	obj = list.get(i)

■例題21

問題:ArrayList に、ダイアログから入力された文字列を任意個格納した後に、まとめて一覧をコンソールに表示せよ。



動作確認クラス： Ex21ArrayList

例題21 Ex21ArrayList

```java
package j2.lesson13;

import javax.swing.JOptionPane;
import java.util.ArrayList;

public class Ex21ArrayList {
    public static void main(String[] args) {
        ArrayList list = new ArrayList();

        while(true) {
            String message = "書果名を入力していください。\n(end で終了)";
            String input = JOptionPane.showInputDialog(message);
            if(input.equals("end")) {
                break;
            } else {
                list.add(input);
            }
        }

        for(int i = 0; i < list.size(); i++) {
            System.out.println(list.get(i));
        }
    }
}
```

警告が出ている

```
6  public class Ex21ArrayList {
7      public static void main(String[] args) {
8  この行に複数マーカーがあります
9    - ArrayList は raw 型です。総称型 ArrayList<E> への参照は、パラメーター化する必要があります
10   - ArrayList は raw 型です。総称型 ArrayList<E> への参照は、パラメーター化する必要があります
11                                                                    d で終了)");
12             if(input.equals("end")) {
```

上級コラム

Eclipse で警告が出る理由

- ArrayList には、どんな型のインスタンスでも登録できる。
  - 逆に、内容を参照する時に、どの型のインスタンスが取り出されるかわからないので、明示的なキャストが必要。
- 上記のキャストの問題を回避するために、あらかじめ、どの型のインスタンスを登録するかを宣言する「型パラメータ」をつけることができるようになった
  - ArrayList<String> list = new ArrayList<String>();
  - 昔からの使い方に従って、型パラメータを使わないで ArrayList を使うと、Eclipse が警告を出すようになっている
  - 警告は表示されるが、実行には支障がない。

例題21 Ex21ArrayList（警告なし版）

```java
package j2.lesson13;

import javax.swing.JOptionPane;

public class Ex21ArrayList_0 {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<String>();

        while(true) {
            String message = "青果名を入力していください。\n(end で終了)";
            String input = JOptionPane.showInputDialog(message);
            if(input.equals("end")) {
                break;
            } else {
                list.add(input);
            }
        }

        for(int i = 0; i < list.size(); i++) {
            System.out.println(list.get(i));
        }
    }
}
```

【コラム】
ArrayList は、Genericなクラス（総称型クラス）と呼ばれ、クラスに、「型パラメータ」を指定できる。

ArrayList を使う時の注意

- ArrayList に格納できるのは、インスタンスだけである。
  - 基本型の int や double は格納できない。
  - 替わりに、Integer や Double に変換して格納する。
- 値を取り出す時には、必ず登録したデータの型にキャストする。
  - 登録時と違う型の変数に取り出そうとすると例外が発生する。
- 配列の途中の順番に挿入したい時には、add(index, データ)のように、挿入する場所を指定する
  - set(index, データ)は、その場所のデータを置き換える

テーマ : ArrayList と HashMap

- ラッパオブジェクト
  - 基本型と参照型
- ArrayList
- HashMap

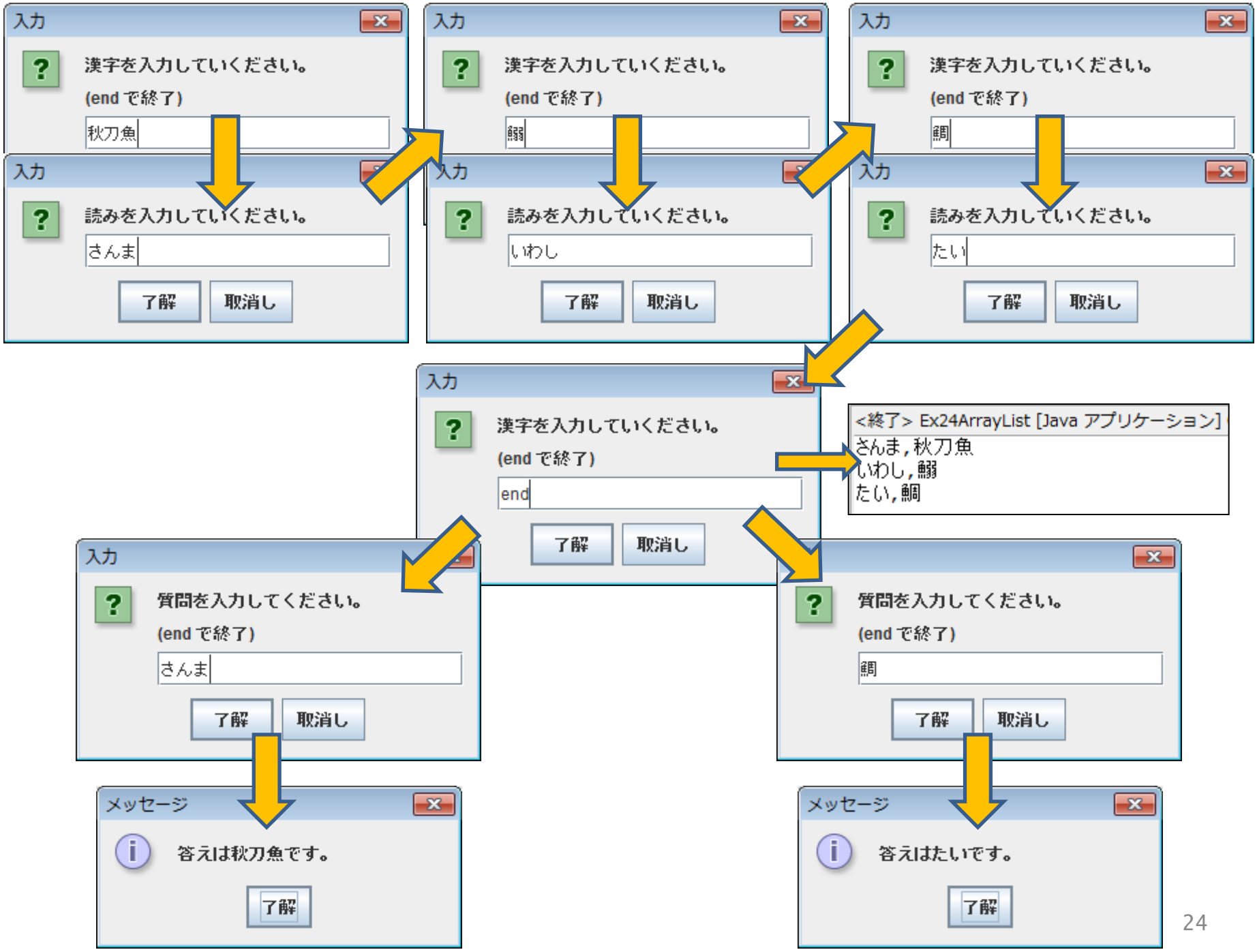
連想記憶 HashMap

- java.util.HashMap として提供される
- キーを指定して、キーと関連付けれられたデータを検索できるライブラリ
- (初期化)　　HashMap map = new HashMap();
- (登録)　　map.put(“キー”, “値”);
- (検索)　　String value
  = (String)map.get(“キー”);
- (キー一覧)　　Set keys = map.keySet();

■例題31

問題: 例題31と同じ動作をするプログラムを、
MyFlashCard を利用しないで、HashMapを使って作成せよ。

動作確認クラス： Ex31HashMapList


入力
漢字を入力していください。
(end で終了)
秋刀魚
入力
読みを入力していください。
さんま
了解
取消し
入力
漢字を入力していください。
(end で終了)
鰯
入力
読みを入力していください。
いわし
了解
取消し
入力
漢字を入力していください。
(end で終了)
鯛
入力
読みを入力していください。
たい
了解
取消し
入力
漢字を入力していください。
(end で終了)
end
了解
取消し
<終了> Ex24ArrayList [Java アプリケーション]
さんま,秋刀魚
いわし,鰯
たい,鯛
入力
質問を入力してください。
(end で終了)
さんま
了解
取消し
メッセージ
答えは秋刀魚です。
了解
入力
質問を入力してください。
(end で終了)
鯛
了解
取消し
メッセージ
答えはたいです。
了解

例題31 Ex31HashMap

```
1  package j2.lesson13;
2
3  import javax.swing.JOptionPane;
4  import java.util.*;
5
6  public class Ex31HashMap {
7      public static void main(String[] args) {
8          HashMap cardMap = new HashMap();
9
10         while(true) {
11             String kanji = JOptionPane.showInputDialog("漢字を入力していください。\n(end で終了)");
12             if(kanji.equals("end")) {
13                 break;
14             } else {
15                 String yomi = JOptionPane.showInputDialog("読みを入力していください。");
16                 cardMap.put(kanji, yomi);
17                 cardMap.put(yomi, kanji);
18             }
19         }
20
21         Set keys = cardMap.keySet();
22         Iterator iterator = keys.iterator();
23
24         while(iterator.hasNext()) {
25             String key = (String)iterator.next();
26             String value = (String)cardMap.get(key);
27             System.out.println(key + "," + value);
28         }
29
```

例題31 Ex31HashMap 続き

```
30          while(true) {
31              String question = JOptionPane.showInputDialog("質問を入力してください。\n(end で終了)");
32              if(question.equals("end")) {
33                  break;
34              } else {
35                  String result = (String)cardMap.get(question);
36                  String message;
37                  if(result == null) {
38                      message = "登録されていません。";
39                  } else {
40                      message = "答えは" + result + "です。";
41                  }
42                  JOptionPane.showMessageDialog(null, message);
43              }
44          }
45      }
46  }
```

HashMap の使う時の注意

- キーも、値も、インスタンスだけが利用できる。
  - 基本型は利用できない。
  - キーは文字列にすることが多いが、等価性を判定できるのであれば、他のインスタンスでも良い。
- 登録したデーター覧を取りだしたい時には工夫が必要
  - キーの集合を取りだす　　Set keys = map.keySet();
  - キーを順に取り出す準備　　Iterator i = keys.iterator();
  - 次のキー　　String key = (String)i.next();
  - キーとペアの値を取りだす　　String val = (String)map.get(key);

まとめ : ArrayList と HashMap

- ラッパオブジェクト
  - 基本型と参照型
- ArrayList
- HashMap

本日の例題と問題

- ラッパクラスの演習
  - Ex11, Ex12, Ex13
- ArrayList の演習
  - Ex21, Ex22, Ex23, Ex24*, Q11, Q12, (Q13)
- HashMap の演習
  - Ex31, (Ex32), Q21, (Q22), Q31, Q32, (Q33)
- OrderingSystem
  - (Ex41*)

(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.lesson13」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11

問題:次のページの Ex11Integer クラスを入力し、// 正確には　となっている行と次の行を入れ替えて、結果が同じことを確認せよ。

また、// エラー　となっている行のコメントをはずしてみて、なぜ、エラーになるのか答えよ。

動作確認クラス: `Ex11Integer`

例題11 Ex11Integer

```java
package j2.lesson13;

public class Ex11Integer {
    public static void main(String[] args) {
        int x = 100;
        Integer xx = Integer.valueOf(x);
        x++;
        // 正確には、xx = Integer.valueOf(xx.intValue() + 1);
        xx++;

        System.out.println(x + ": "+ xx);
        // エラー: System.out.println(x.toString());
        System.out.println(xx.toString());

        double y = 200;
        // エラー: Double yy = 200;
        Double yy = y;
        y++;
        // 正確には、yy = Long.valueOf(yy.longValue() + 1);
        yy++;

        System.out.println(y + ": "+ yy);
        // エラー: System.out.println(y.toString());
        System.out.println(yy.toString());

        y = x;
        // エラー: yy = xx;
    }
}
```

例題11　解説

```
// エラー: System.out.println(x.toString());
```

x は、int 型でありプリミティブ型。よって、Object クラスを継承しないので、toString() 等のインスタンスメソッドは呼べない。

```
// エラー: Double yy = 200;
```

200は、int 型。オブジェクトに自動変換すると、Integer 型。
Integer型オブジェクトは、Double型を継承しないので、代入できない。

```
// エラー: yy = xx;
```

xx は、Integer 型。yy は Double 型。
Integer型オブジェクトは、Double型を継承しないので、代入できない。

■例題12

問題:次のEx12Integer クラスを実行すると、100と100は == が成り立つが、1000と1000では == が成り立たない。成り立つ/成り立たないの境界線はどこにあるのか調べよ。(ヒント: 2進数で考える)

```
1  package j2.lesson13;
2
3  public class Ex12Integer {
4      public static void main(String[] args) {
5          System.out.println("100と100");
6          compare(100, 100);
7          System.out.println("1000と1000");
8          compare(1000, 1000);
9      }
10
11     static void compare(Integer x, Integer y) {
12         if(x == y) {
13             System.out.println("x == y が成立");
14         }
15         if(x.equals(y)) {
16             System.out.println("x.equals(y) が成立");
17         }
18     }
19 }
```

動作確認クラス： Ex12Integer

■例題13

問題:次のページのEx13Integer クラスで、コメントアウトされている // エラー: 行は、なぜエラーになるのか考えよ。

(型変換の明示)

plusInteger に対する plusInteger2 を参考に、plusLong に対するplusLong2 を作成して、動作を確認せよ。

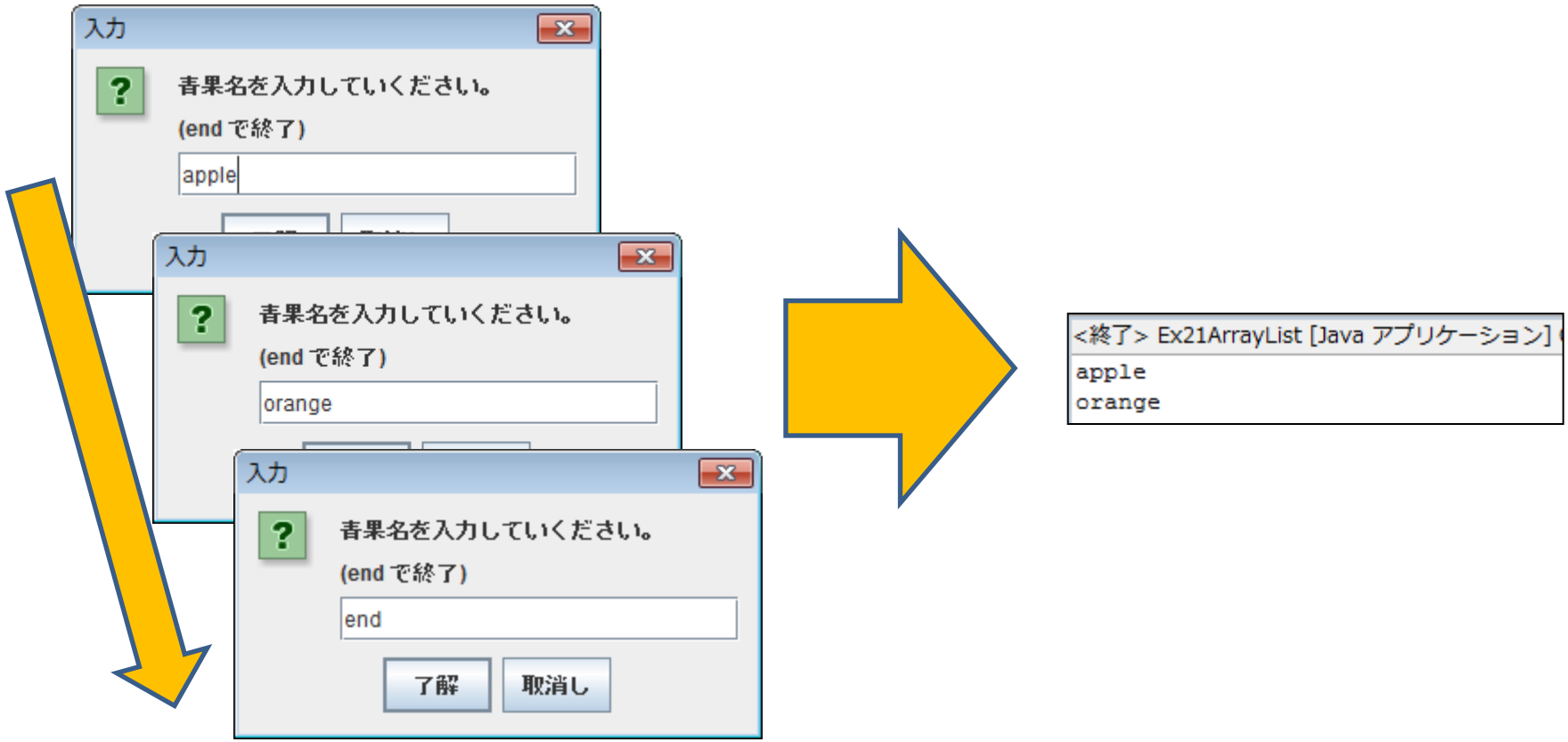
動作確認クラス： Ex13Integer

例題13
Ex13Integer

```java
package j2.lesson13;

public class Ex13Integer {
    public static void main(String[] args) {
        int i0 = plusint(100, 200);
        long l0 = pluslong(100, 200);
        System.out.println(i0 + ": "+ l0);

        Integer iObject = plusInteger(100, 200);
        // エラー: Long lObject = plusLong(100, 200);
        Long lObject = pluslong(100L, 200L);
        System.out.println(iObject + ": "+ lObject);
    }

    static int plusint(int x, int y) {
        return x + y;
    }

    static long pluslong(long x, long y) {
        return x + y;
    }

    static Integer plusInteger(Integer x, Integer y) {
        return x + y;
    }

    static Integer plusInteger2(Integer x, Integer y) {
        int xx = x.intValue();
        int yy = y.intValue();

        return Integer.valueOf(xx + yy);
    }

    static Long plusLong(Long x, Long y) {
        return x + y;
    }
}
```

■例題21

問題:ArrayList に、ダイアログから入力された文字列を任意個格納した後に、まとめて一覧をコンソールに表示せよ。



動作確認クラス： Ex21ArrayList

例題21 Ex21ArrayList

```java
package j2.lesson13;

import javax.swing.JOptionPane;
import java.util.ArrayList;

public class Ex21ArrayList {
    public static void main(String[] args) {
        ArrayList list = new ArrayList();

        while(true) {
            String message = "青果名を入力していください。\n(end で終了)";
            String input = JOptionPane.showInputDialog(message);
            if(input.equals("end")) {
                break;
            } else {
                list.add(input);
            }
        }

        for(int i = 0; i < list.size(); i++) {
            System.out.println(list.get(i));
        }
    }
}
```

警告が出ている

```
6  public class Ex21ArrayList {
7      public static void main(String[] args) {
8  この行に複数マーカーがあります
9    - ArrayList は raw 型です。総称型 ArrayList<E> への参照は、パラメーター化する必要があります
10   - ArrayList は raw 型です。総称型 ArrayList<E> への参照は、パラメーター化する必要があります
11                                                                    d で終了)");
12             if(input.equals("end")) {
```

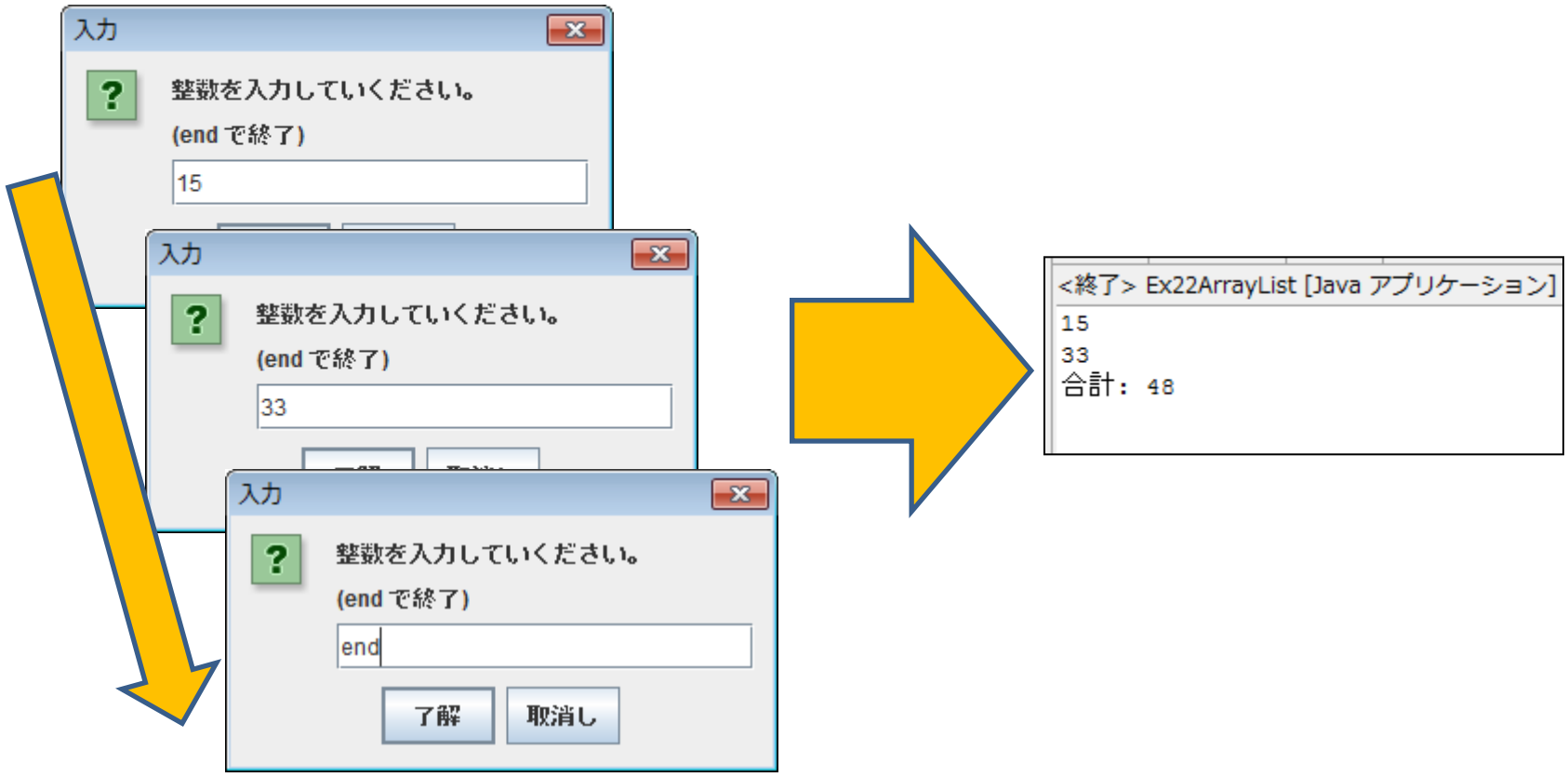
例題21 Ex21ArrayList（警告なし版）

```java
package j2.lesson13;

import javax.swing.JOptionPane;

public class Ex21ArrayList_0 {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<String>();

        while(true) {
            String message = "青果名を入力していください。\n(end で終了)";
            String input = JOptionPane.showInputDialog(message);
            if(input.equals("end")) {
                break;
            } else {
                list.add(input);
            }
        }

        for(int i = 0; i < list.size(); i++) {
            System.out.println(list.get(i));
        }
    }
}
```

【コラム】
ArrayList は、Genericなクラス（総称型クラス）と呼ばれ、クラスに、「型パラメータ」を指定できる。

■例題22

問題:ArrayList に、ダイアログから入力された整数を任意個格納した後に、まとめて一覧をコンソールに表示しながら、合計も計算して表示せよ。



動作確認クラス: Ex22ArrayList

例題22 Ex22ArrayList

```
1  package j2.lesson13;
2
3  import javax.swing.JOptionPane;
4  import java.util.ArrayList;
5
6  public class Ex22ArrayList {
7      public static void main(String[] args) {
8          ArrayList list = new ArrayList();
9
10         while(true) {
11             String message = "整数を入力していください。\n(end で終了)";
12             String input = JOptionPane.showInputDialog(message);
13             if(input.equals("end")) {
14                 break;
15             } else {
16                 Integer number = Integer.valueOf(input);
17                 list.add(number);
18             }
19         }
20
21         int sum = 0;
22         for(int i = 0; i < list.size(); i++) {
23             System.out.println(list.get(i));
24             sum += ((Integer)list.get(i)).intValue();
25         }
26         System.out.println("合計: " + sum);
27     }
28 }
```

■例題23

問題: MyFlashCardに、ダイアログから読みと漢字のペアを入力し、CSV形式の一覧をコンソールに表示せよ。

MyFlashCard

クラス変数	初期値	説明
ArrayList contents	new ArrayList();	生成したインスタンスを格納する配列

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

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

(クラス名： MyFlashCard)
動作確認クラス： Ex23ArrayList

■例題23(続き)

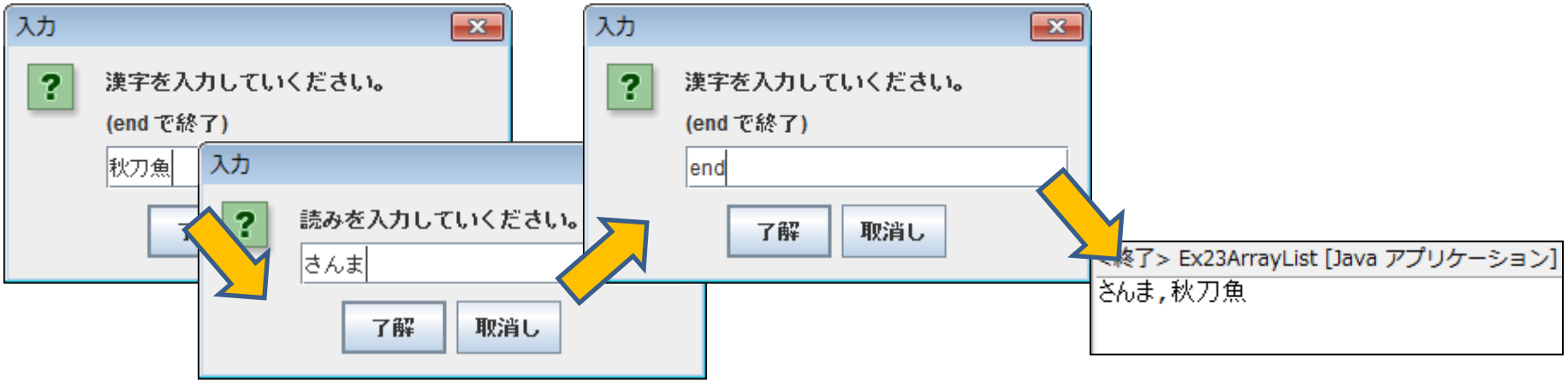
MyFlashCard 続き

クラスメソッド

返り値の型	メソッド名(引数)	機能
MyFlashCard	createFlashCard(String word1, String word2)	インスタンス変数の値がword1,word2であるMyFlashCardクラスのインスタンスを作成しクラス変数のcontentsに格納する。

インスタンスメソッド

返り値の型	メソッド名(引数)	機能
MyFlashCard	get(int i)	i番目のMyFlashCard を返却
int	size()	MyFlashCard のインスタンスの総数



(クラス名： MyFlashCard)

例題23 MyFlashCard

```
 1  package j2.lesson13;
 2
 3  import java.util.ArrayList;
 4
 5  public class MyFlashCard {
 6      static ArrayList contents = new ArrayList();
 7
 8      String word1;
 9      String word2;
10
11      private MyFlashCard(String word1, String word2) {
12          this.word1 = word1;
13          this.word2 = word2;
14      }
15
16      public static MyFlashCard createFlashCard(String word1, String word2) {
17          MyFlashCard card = new MyFlashCard(word1, word2);
18          contents.add(card);
19          return card;
20      }
21
22      public static MyFlashCard get(int i) {
23          return (MyFlashCard)contents.get(i);
24      }
25
26      public static int size() {
27          return contents.size();
28      }
29  }
```

例題23 Ex23ArrayList

```
1  package j2.lesson13;
2
3  import javax.swing.JOptionPane;
4
5  public class Ex23ArrayList {
6      public static void main(String[] args) {
7          while(true) {
8              String kanji = JOptionPane.showInputDialog("漢字を入力していください。\n(end で終了)");
9              if(kanji.equals("end")) {
10                 break;
11             } else {
12                 String yomi = JOptionPane.showInputDialog("読みを入力していください。");
13                 MyFlashCard.createFlashCard(yomi, kanji);
14             }
15         }
16
17         for(int i = 0; i < MyFlashCard.size(); i++) {
18             MyFlashCard card = MyFlashCard.get(i);
19             System.out.println(card.word1 + "," + card.word2);
20         }
21     }
22 }
```

■例題24

問題: 例題23に加えて、登録したMyFlashCardに対して、質問を出し、対になる言葉をダイアログ表示するプログラムを作成せよ。

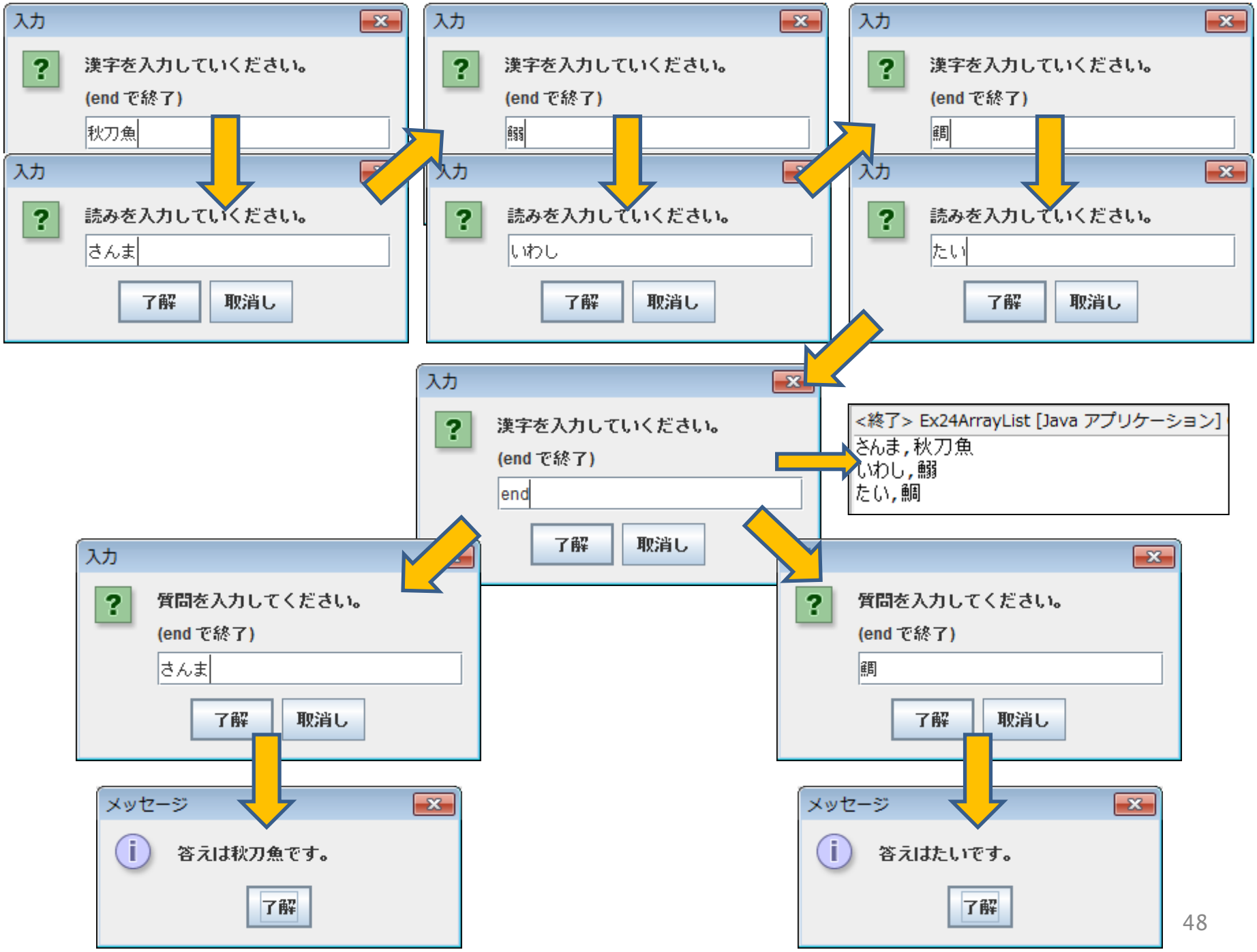
MyFlashCardに追加するインスタンスメソッド

インスタンスメソッド

返り値の型	メソッド名(引数)	機能
String	get(String key)	key を基に、対になる言葉を検索する。漢字→かな、かな→漢字の双方の対を検索できること。

(クラス名： MyFlashCard)

動作確認クラス： Ex24ArrayList


入力
漢字を入力していください。
(end で終了)
秋刀魚
入力
読みを入力していください。
さんま
了解
取消し
入力
漢字を入力していください。
(end で終了)
鰯
入力
読みを入力していください。
いわし
了解
取消し
入力
漢字を入力していください。
(end で終了)
鯛
入力
読みを入力していください。
たい
了解
取消し
入力
漢字を入力していください。
(end で終了)
end
了解
取消し
<終了> Ex24ArrayList [Java アプリケーション]
さんま,秋刀魚
いわし,鰯
たい,鯛
入力
質問を入力してください。
(end で終了)
さんま
了解
取消し
メッセージ
答えは秋刀魚です。
了解
入力
質問を入力してください。
(end で終了)
鯛
了解
取消し
メッセージ
答えはたいです。
了解

例題24 MyFlashCard

```
1  package j2.lesson13;
2
3  import java.util.ArrayList;
4
5  public class MyFlashCard {
6      static ArrayList contents = new ArrayList();
7
8      String word1;
9      String word2;
10
11     private MyFlashCard(String word1, String word2) {..
15
16     public static MyFlashCard createFlashCard(String word1, String word2) {..
21
22     public static MyFlashCard get(int i) {..
25
26     public static int size() {..
29
30     public static String get(String key) {
31         int end = contents.size();
32         for(int i = 0; i < end; i++) {
33             MyFlashCard card = (MyFlashCard)contents.get(i);
34             if(card.word1.equals(key)) {
35                 return card.word2;
36             } else if(card.word2.equals(key)) {
37                 return card.word1;
38             }
39         }
40         return null;
41     }
42 }
```

例題24 Ex24ArrayList

```
1  package j2.lesson13;
2
3  import javax.swing.JOptionPane;
4
5  public class Ex23ArrayList {
6      public static void main(String[] args) {
7          while(true) {
8              String kanji = JOptionPane.showInputDialog("漢字を入力していください。\n(end で終了)");
9              if(kanji.equals("end")) {
10                 break;
11             } else {
12                 String yomi = JOptionPane.showInputDialog("読みを入力していください。");
13                 MyFlashCard.createFlashCard(yomi, kanji);
14             }
15         }
16
17         for(int i = 0; i < MyFlashCard.size(); i++) {
18             MyFlashCard card = MyFlashCard.get(i);
19             System.out.println(card.word1 + "," + card.word2);
20         }
21
22         while(true) {
23             String question = JOptionPane.showInputDialog("質問を入力してください。\n(end で終了)");
24             if(question.equals("end")) {
25                 break;
26             } else {
27                 String result = MyFlashCard.get(question);
28                 String message;
29                 if(result == null) {
30                     message = "登録されていません。";
31                 } else {
32                     message = "答えは" + result + "です。";
33                 }
34                 JOptionPane.showMessageDialog(null, message);
35             }
36         }
37     }
38 }
```

■例題31

問題: 例題24と同じ動作をするプログラムを、
MyFlashCard を利用しないで、HashMapを使って作成せよ。

動作確認クラス： Ex31HashMapList

例題31 Ex31HashMap

```
1  package j2.lesson13;
2
3  import javax.swing.JOptionPane;
4  import java.util.*;
5
6  public class Ex31HashMap {
7      public static void main(String[] args) {
8          HashMap cardMap = new HashMap();
9
10         while(true) {
11             String kanji = JOptionPane.showInputDialog("漢字を入力していください。\n(end で終了)");
12             if(kanji.equals("end")) {
13                 break;
14             } else {
15                 String yomi = JOptionPane.showInputDialog("読みを入力していください。");
16                 cardMap.put(kanji, yomi);
17                 cardMap.put(yomi, kanji);
18             }
19         }
20
21         Set keys = cardMap.keySet();
22         Iterator iterator = keys.iterator();
23
24         while(iterator.hasNext()) {
25             String key = (String)iterator.next();
26             String value = (String)cardMap.get(key);
27             System.out.println(key + "," + value);
28         }
29
```

例題31 Ex31HashMap 続き

```
30          while(true) {
31              String question = JOptionPane.showInputDialog("質問を入力してください。\n(end で終了)");
32              if(question.equals("end")) {
33                  break;
34              } else {
35                  String result = (String)cardMap.get(question);
36                  String message;
37                  if(result == null) {
38                      message = "登録されていません。";
39                  } else {
40                      message = "答えは" + result + "です。";
41                  }
42                  JOptionPane.showMessageDialog(null, message);
43              }
44          }
45      }
46  }
```

■例題32(opt)

問題: 例題31のEx31HashMap の警告マークを全てなくすように、Generics の設定を行え。

動作確認クラス： Ex32HashMapList

例題32　Ex32HashMap

3行以外に変更無し

```
1  package j2.lesson13;
2
3  import javax.swing.JOptionPane;
4  import java.util.*;
5
6  public class Ex32HashMap {
7      public static void main(String[] args) {
8          HashMap<String, String> cardMap = new HashMap<String, String>();
9
10         while(true) {
11             String kanji = JOptionPane.showInputDialog("漢字を入力していください。\n(end で終了)");
12             if(kanji.equals("end")) {
13                 break;
14             } else {
15                 String yomi = JOptionPane.showInputDialog("読みを入力していください。");
16                 cardMap.put(kanji, yomi);
17                 cardMap.put(yomi, kanji);
18             }
19         }
20
21         Set<String> keys = cardMap.keySet();
22         Iterator<String> iterator = keys.iterator();
23
24         while(iterator.hasNext()) {
25             String key = (String)iterator.next();
26             String value = (String)cardMap.get(key);
27             System.out.println(key + "," + value);
28         }
29
```