プログラミング入門2

第8回　表形式データ(1)

テーマ：表形式データ(1)

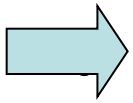
配列と複合データを用いた表形式データ

- データの登録
- データの検索
- データの更新

実際的はソフトウェアでは、表形式データの（例えば、データベースのデータ）を利用する場面が非常に多く、とても重要である。そこで、表形式を扱うプログラミングを繰り返しとりあげる。

テーマ：表形式データ(1)

配列と複合データを用いた表形式データ



データの登録

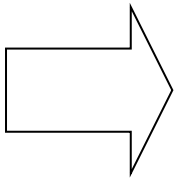
- データの検索
- データの更新

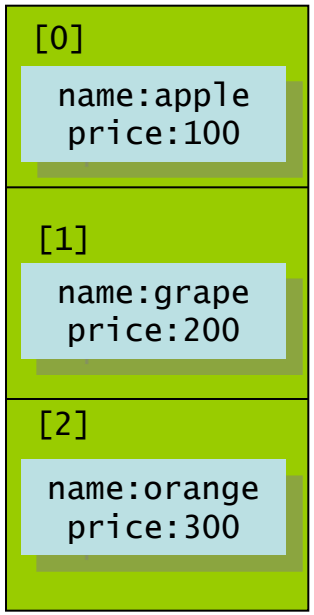
表形式のデータ

第5回、6回で取り上げたように、複数のデータ（データの集合）を処理しやすい形で蓄積するためには、表の形式を用いると良い。表形式をプログラムで実現するために、配列を用意しその要素として複合データを割り当てる。

下記では、商品の価格表を作成することを考える。
商品の名前と、単価を組にして管理する。

商品名	単価
apple	100
grape	200
orange	300





■例題40より

問題：次のクラスProductDataを作成せよ。このクラスのインスタンスは商品の情報を保持する。

インスタンス変数

変数の型と名前	初期値	説明
String name	無し	商品の名前
int price	無し	商品の単価

コンストラクタ

コンストラクタ(引数)	機能
ProductData (String name, int price)	ProductDataクラスのインスタンスを生成し、引数のname,priceを各インスタンス変数に代入する。

インスタンスメソッド

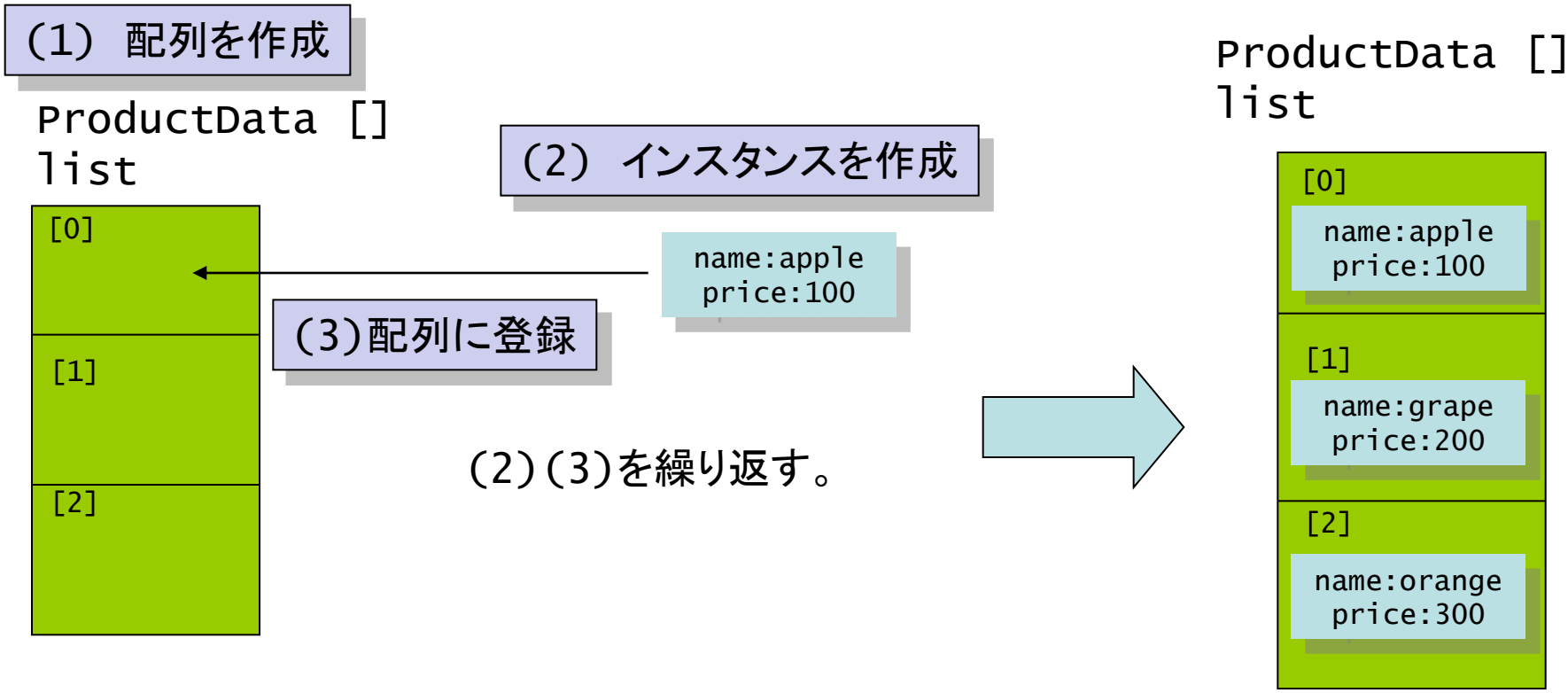
返り値の型	メソッド名(引数)	機能
void	showProductData(int row)	ProductDataインスタンスの文字列表現をSpreadsheetの row行に表示する。
void	header()	Spreadsheet のヘッダ行を表示する。

（クラス名： ProductData）

例題40

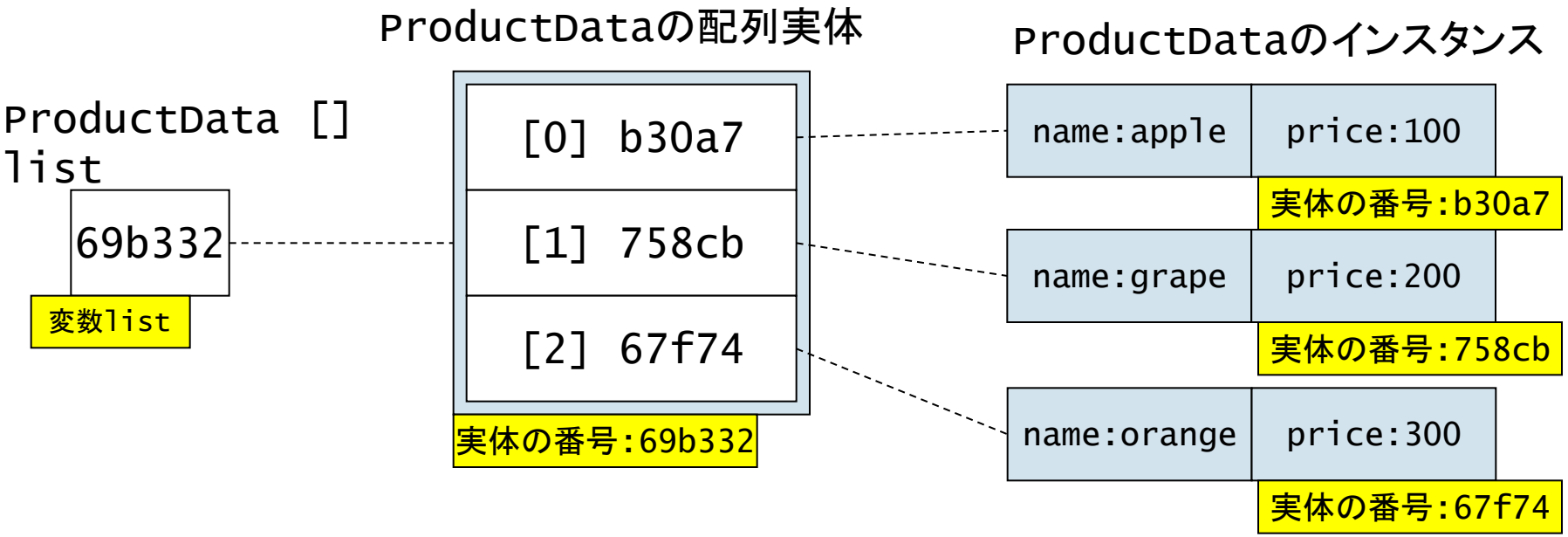
```
 1  package j2.lesson08;
 2
 3  import gpjava.Spreadsheet;
 4
 5  public class ProductData {
 6      String name;
 7      int price;
 8
 9      ProductData(String name, int price) {
10          this.name = name;
11          this.price = price;
12      }
13
14      void showProductData(int row) {
15          Spreadsheet.setString(row, 0, name);
16          Spreadsheet.setInt(row, 1, price);
17      }
18
19      void header() {
20          Spreadsheet.setString(0, 0, "商品名");
21          Spreadsheet.setString(0, 1, "単価");
22      }
23  }
24
```

配列と複合データを用いた表形式データの作成
表へのレコードの登録(考え方)




```
int numProducts = 3;
ProductData [] list = new ProductData[numProducts];
list[0] = new ProductData("apple", 100);
list[1] = new ProductData("grape", 200);
list[2] = new ProductData("orange", 300);
```

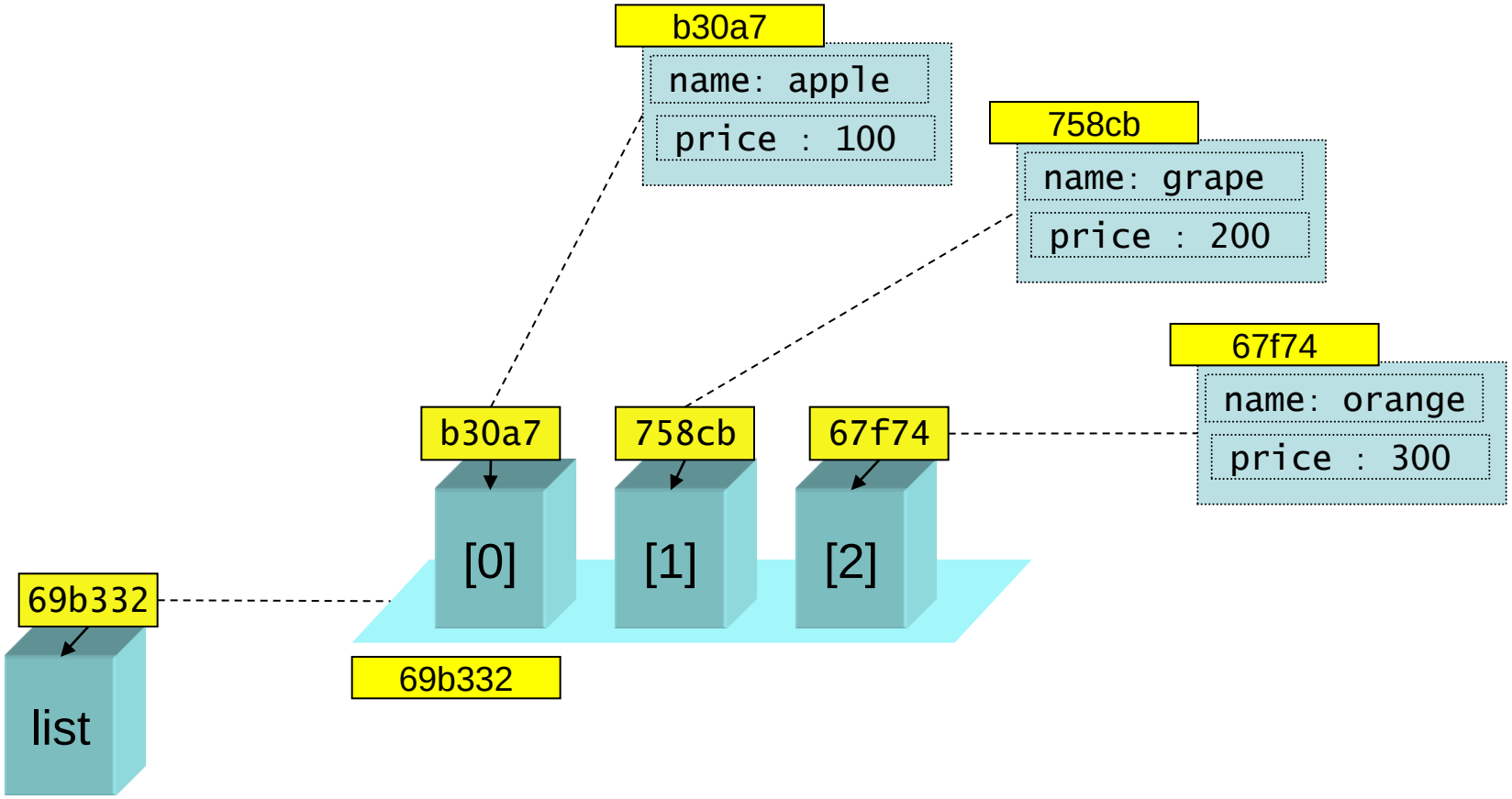
配列と複合データを用いた表形式データ(詳細)



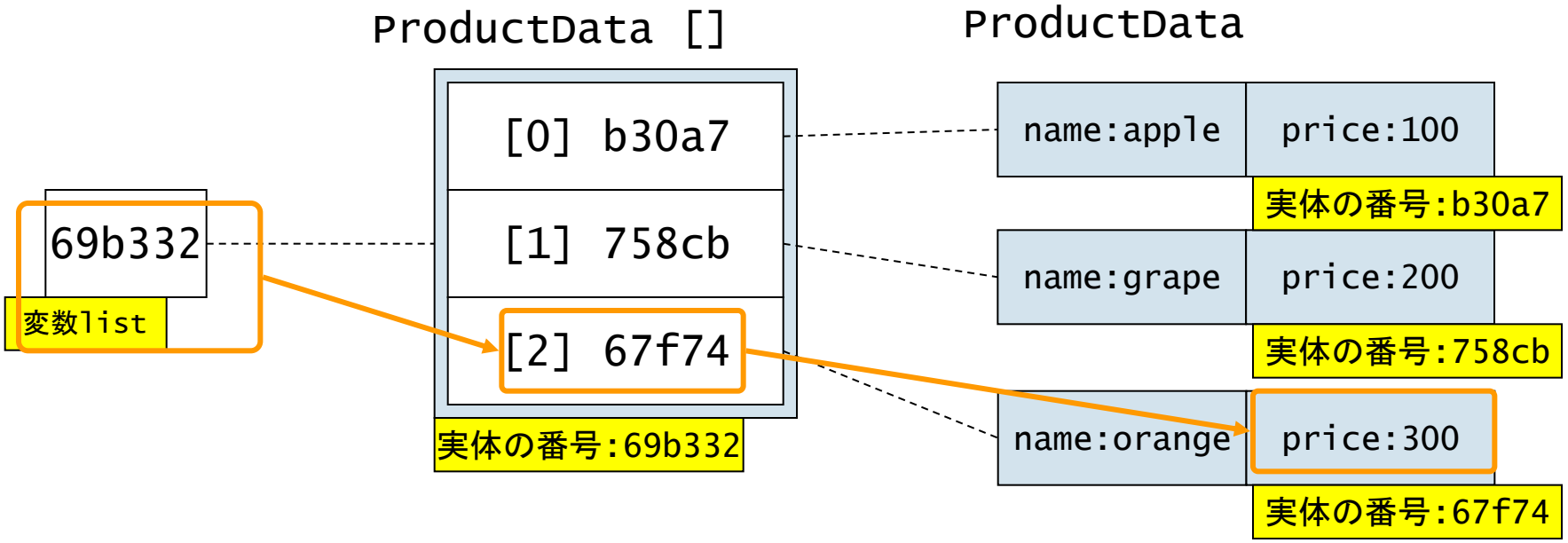
実際には、配列変数listには配列実体への参照が格納される。
また、配列実体にはインスタンスの参照が格納される。

```
int numProducts = 3;
ProductData [] list = new ProductData[numProducts];
list[0] = new ProductData("apple", 100);
list[1] = new ProductData("grape", 200);
list[2] = new ProductData("orange", 300);
```

箱の絵を用いたイメージ図（参考）



配列に含まれるインスタンスへのアクセス(詳細)



配列に含まれるインスタンスへのアクセス(list[2].priceの例)

```
System.out.println(list[2].price);
```

list[2] : 69b332番の配列データの2番目の内容を取り出す→67f74
list[2].price: 67f74番のデータのインスタンス変数priceの内容を取りだす→300)
println(list2[2].price): 300を表示

表の走査(scan)

表に登録されているデータを一つ一つ調べることを、走査(scan)という。
砕けた言葉では、表を「なめる」という言い方もある。

配列の先頭から調べる場合の例は、下記である。

```
for (int i = 0; i < list.length; i++ ) {
  // ここに処理を書く。list[i]として、
  //インスタンスを一つ一つ取り出す。
}
```

ProductData []


```
for (int i = 0; i < list.length; i++ ) {
  ProductData p = list[i];
  System.out.println(p.price);
}
```

```
System.out.println(list[0].price);
System.out.println(list[1].price);
System.out.println(list[2].price);
```

出力

```
100
200
300
```

■例題41(1) より

問題: ProductDataのインスタンスを配列に格納するようにせよ。さらに以下のメソッドをEx41ProductListの中に作成し、動作を確認せよ。

返り値の型	メソッド名(引数)	機能
void	contents(ProductData [ ] list)	listの各ProductDataインスタンスの文字列表現を表示する。

(クラス名： Ex41ProductList)

例題41(1)

```
1  package j2.lesson08;
2
3  import gpjava.Spreadsheet;
4
5  public class Ex41ProductList_1 {
6
7      public static void main(String[] args) {
8          new Ex41ProductList_1().start();
9      }
10
11     void start() {
12         int numProducts = 3;
13         ProductData [] list = new ProductData[numProducts];
14         list[0] = new ProductData("apple", 100);
15         list[1] = new ProductData("grape", 200);
16         list[2] = new ProductData("orange", 300);
17
18         Spreadsheet.show();
19         list[0].header();
20         contents(list);
21     }
22
23     void contents(ProductData [] list) {
24         for (int i = 0; i < list.length; i++ ) {
25             list[i].showProductData(i + 1);
26         }
27     }
28 }
```

表(配列)を走査し、要素の内容を順次表示していく

テーマ：表形式データ(1)

配列と複合データを用いた表形式データ

- データの登録
- データの検索
- データの更新

データの検索

表に登録されているデータの集合から、欲しい特定のデータを取り出すことを「検索」(search)と言う。検索は、配列内を走査(スキャン)して、検索条件に合う要素を列挙する作業である。

基本の検索条件は、名前の一致である。次のメソッドgetは、listのなかから商品名がnameであるインスタンスを検索するメソッドである。

```
ProductData
get (ProductData [] list, String name) {
  for (int i = 0; i < list.length; i++ ) {
    if (list[i].name.equals(name)){
      return list[i];
    }
  }
  return null;
}
```

表を走査する。list[i]のインスタンス変数nameの内容が、引数nameと同じ場合に、そのインスタンス(list[i])を返り値として返す。

ここに到達した、ということは、条件に合うインスタンスが配列に含まれていなかったことになる。

定数nullを返す。

検索条件の例

名前が”apple”であるものを返す。

```
for (int i = 0; i < list.length; i++ ) {
    if (list[i].name.equals(“apple”)){
      return list[i];
    }
  }
```

値段が100円以上150円未満のものを返す。

```
for (int i = 0; i < list.length; i++ ) {
    int price = list[i].price;
    if (100 <= price && price < 150){
      return list[i];
    }
  }
```

一般には、複数のデータが条件に合致する(マッチする)ことがあるが、この例では、配列の先頭から探して、最初に見つかったインスタンス一つを返している。

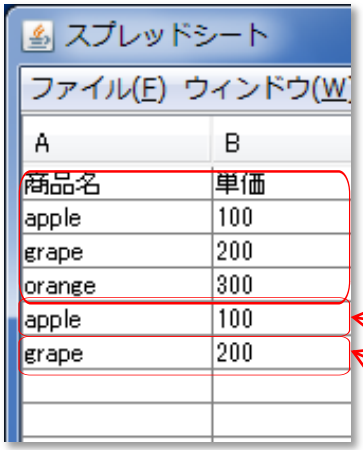
■例題41(2) より

問題:次のメソッドgetを作成し、動作を確認せよ。

返り値の型	メソッド名(引数)	機能
ProductData	get(ProductData [ ] list, String name)	list中に格納されているProductDataインスタンスのうち,引数のnameと同じ名前nameを持つものを返す。該当する名前が見つからない場合は、nullを返す。
既存メソッド(Ex41ProductList内)		
String contents(ProductData [ ] list)		

※既存のEx41ProductListを編集する

例題41(2)



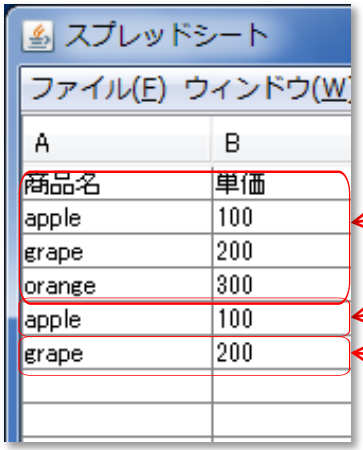

```java
package j2.lesson08;

import gpjava.Canvas;
import gpjava.Spreadsheet;

public class Ex41ProductList_2 {

    public static void main(String[] args) {
        new Ex41ProductList_2().start();
    }

    void start() {
        int numProducts = 3;
        ProductData[] list = new ProductData[numProducts];
        list[0] = new ProductData("apple", 100);
        list[1] = new ProductData("grape", 200);
        list[2] = new ProductData("orange", 300);

        Spreadsheet.show();
        Canvas.show();
        list[0].header();
        contents(list);

        ProductData p = get(list, "apple");
        p.showProductData(list.length + 1);

        Canvas.waitForCountdown(1000);
        p = get(list, "grape");
        p.showProductData(list.length + 2);

        Canvas.waitForCountdown(1000);
        p = get(list, "cherry");
        p.showProductData(list.length + 3);
        //pにはnullが入るので、エラー
    }

        for (int i = 0; i < list.length; i++) {
            list[i].showProductData(i + 1);
        }
    }

    ProductData get(ProductData[] list, String name) {
        for (int i = 0; i < list.length; i++) {
            if (list[i].name.equals(name)) {
                return list[i];
            }
        }
        return null;
    }

}
```

コンソール　アウトライン

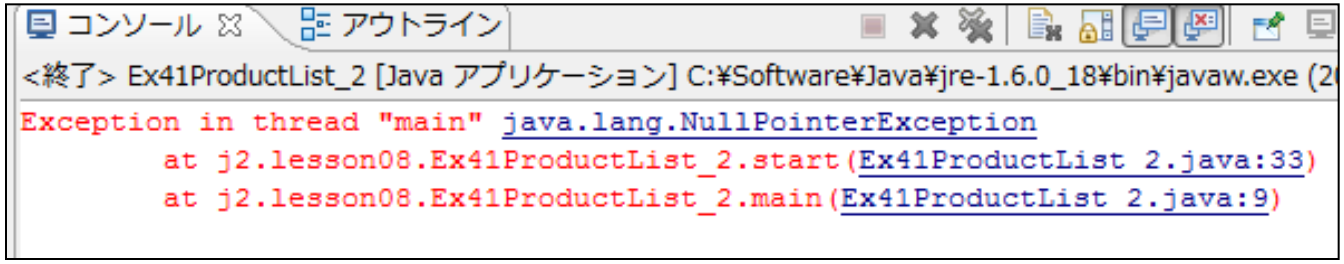
<終了> Ex41ProductList_2 [Java アプリケーション] C:¥Software¥Java¥jre-1.6.0_18¥bin¥javaw.exe (2

```
Exception in thread "main" java.lang.NullPointerException
        at j2.lesson08.Ex41ProductList_2.start(Ex41ProductList_2.java:33)
        at j2.lesson08.Ex41ProductList_2.main(Ex41ProductList_2.java:9)
```

例題41(2)実行の流れ （前頁のものを拡大）




```
11
12    void start() {
13        int numProducts = 3;
14        ProductData[] list = new ProductData[numProducts];
15        list[0] = new ProductData("apple", 100);
16        list[1] = new ProductData("grape", 200);
17        list[2] = new ProductData("orange", 300);
18
19        Spreadsheet.show();
20        Canvas.show();
21        list[0].header();
22        contents(list);
23
24        ProductData p = get(list, "apple");
25        p.showProductData(list.length + 1);
26
27        Canvas.waitForCountdown(1000);
28        p = get(list, "grape");
29        p.showProductData(list.length + 2);
30
31        Canvas.waitForCountdown(1000);
32        p = get(list, "cherry");
33        p.showProductData(list.length + 3);
34        // p にはnullが入るので、エラー
35    }
```



NullPointerException

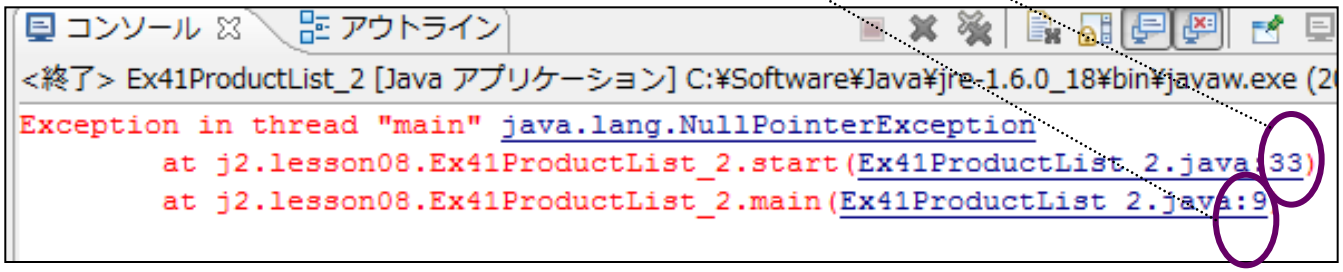
（参照しているインスタンスがない（null）にもかかわらず、その参照をたどって、インスタンス変数などにアクセスしようとした時に発生するエラー（例外））

showProductData を呼び出す33行目で、エラー（例外）が発生。
（pの値がnullであるにもかかわらず、p.showProductData()として、インスタンスメソッドにアクセスしようとしたため。）

```
32          p = get(list, "cherry");
33          p.showProductData(list.length + 3);
34          // p にはnullが入るので、エラー
```

上記メソッドコールは start() メソッド内の33行目にあり、これを呼び出したのがmainメソッド内（Ex41ProductListの9行目）

⇨ 33行目のpがnullであることが原因



テーマ：表形式データ(1)

配列と複合データを用いた表形式データ

- データの登録
- データの検索
- データの更新

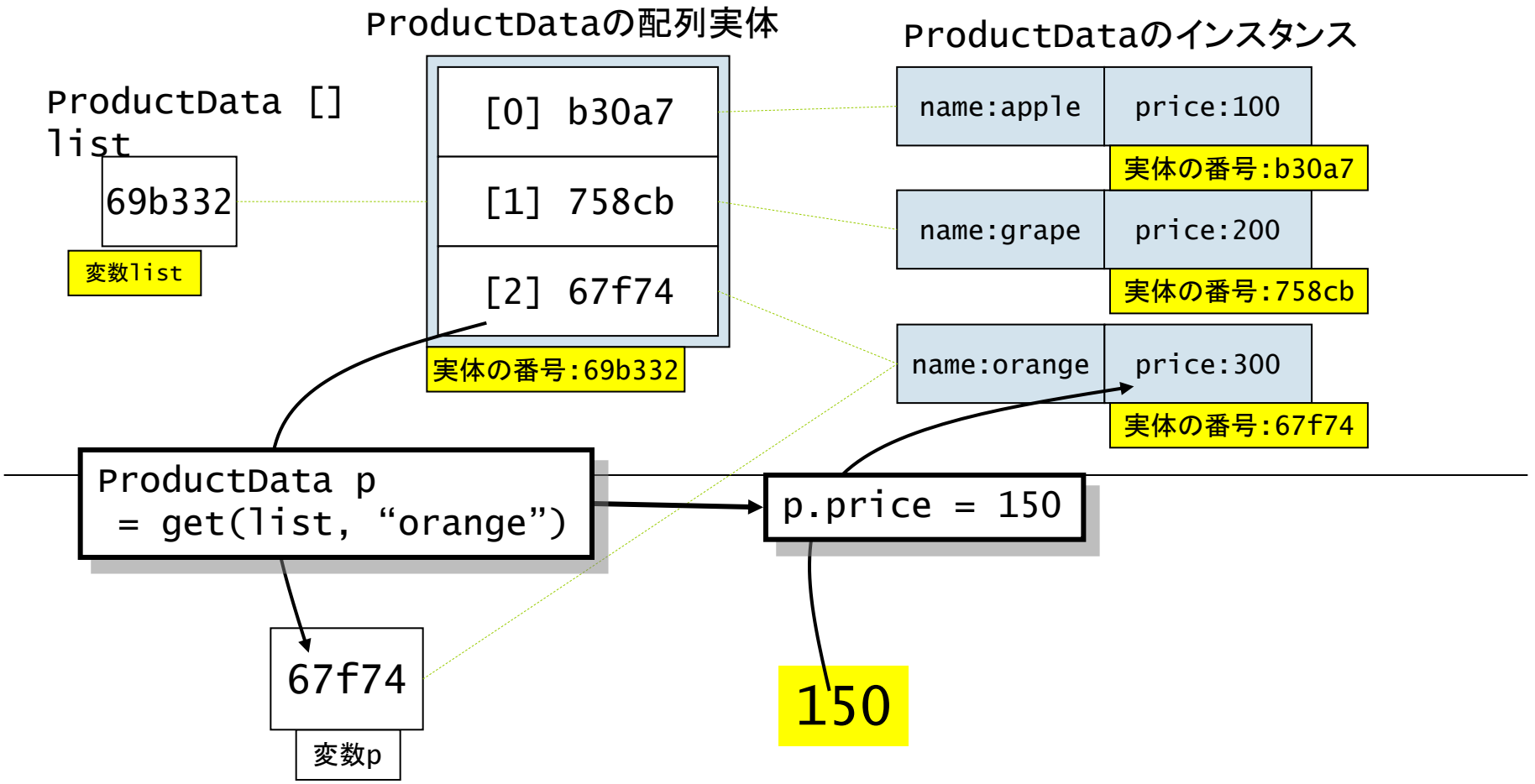
データの更新

表に登録されているデータを修正することを、データの「更新」(update)という。今回は以下の方法を紹介する。

(1) 更新したいデータ(インスタンス)を「検索」によって取り出す。
(2) (1)で取り出したデータのインスタンス変数に、更新したい値を上書きする。

```
int numProducts = 3;
ProductData [] list = new ProductData[numProducts];
list[0] = new ProductData("apple", 100);
list[1] = new ProductData("grape", 200);
list[2] = new ProductData("orange", 300);

ProductData p = get(list, “orange”);
p.price = 150;
```

データの更新(詳細)




```
int numProducts = 3;
ProductData [] list = new ProductData[numProducts];
list[0] = new ProductData("apple", 100);
   … // 省略
ProductData p = get(list, “orange”);
p.price = 150;
```

まとめ：表形式データ(1)

配列と複合データを用いた表形式データ

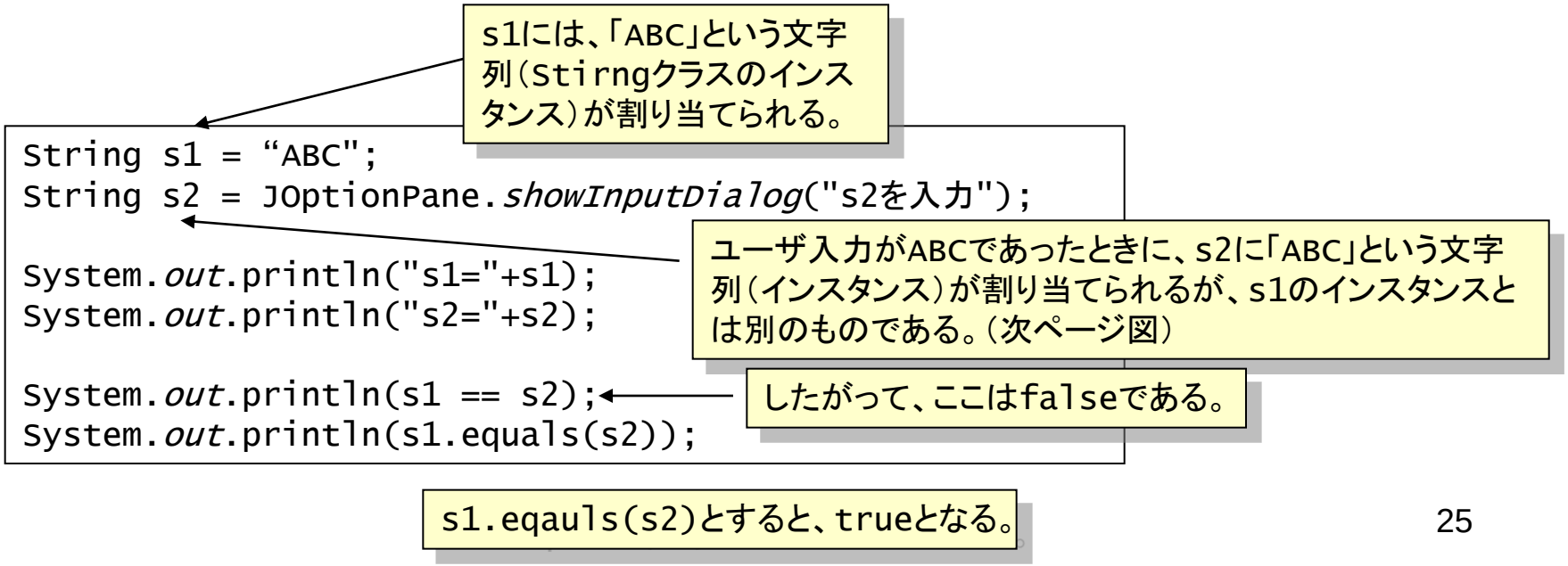
- データの登録
- データの検索
- データの更新

表形式データに対する必要な処理として、データの追加、削除がある。
これらについては次回扱う。

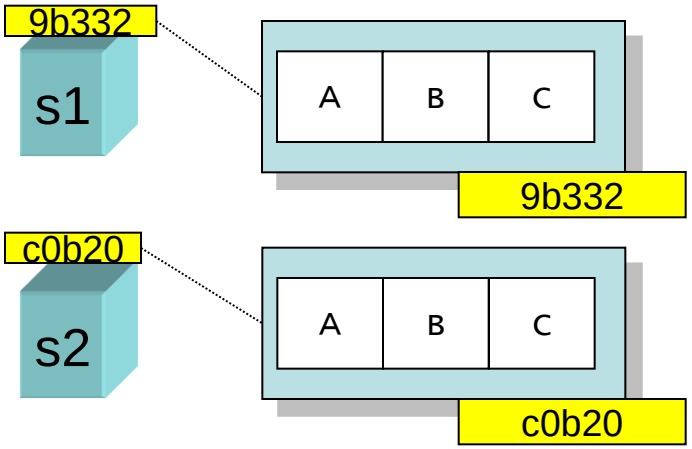
補足(文字列同士の比較)

文字列s1,s2の比較は、s1.equals(s2)という形で行う。なぜs1 == s2ではいけないのか？

これは、文字列型のデータは、実はクラスStringのインスタンスだからである。
(ただし、クラスStringとそのインスタンスは、扱いが少し特別である。)
次のプログラムを考えてみる。



```
String s1 = “ABC";
String s2 = JOptionPane.showInputDialog("s2を入力");

System.out.println("s1="+s1);
System.out.println("s2="+s2);

System.out.println(s1 == s2);
System.out.println(s1.equals(s2));
```



s1とs2は異なるインスタンスを
参照している。

9b332　c0b20 → false

s1==s2

s1.equals(s2) → true

s1の1文字目 == s2の1文字目 → true

s1の2文字目 == s2の2文字目 → true

s1の3文字目 == s2の3文字目 → true

同じ位置の文字が
全て等しいかどうか
調べる。

コラム（データの更新と不要データ）

データの更新の他の方法として、新しいインスタンスを配列に上書きする方法もある。

```
int numProducts = 3;
ProductData [] list = new ProductData[numProducts];
list[0] = ProductData.createProductData("apple", 100);
list[1] = ProductData.createProductData("grape", 200);
list[2] = ProductData.createProductData("orange", 300);

list[2] = ProductData.createProductData(“orange”, 150);
```

この場合、もともと登録されていたデータ（“organe”， 300）は、二度と利用されない不要なデータとなる。

Java言語では、不要データが占めていた記憶領域（メモリ）は、自動的に再利用される。この仕組みを、ガベージコレクション（GC: Garbage Collection ： ゴミ集めの意）という。

C言語などGCが用意されていない言語では、不要になるデータを再利用するためには、別に処理が必要である。

本日の例題と問題

- NamedColor のリストを利用した演習
  - Ex11, Ex12, Ex13(0), Ex13(1), Ex21(1), Ex21(2), Ex21(3), Ex31(1), Ex31(2)
- 商品リストを利用した演習
  - Ex40, Ex41(1), Ex41(2), Ex41(3), Ex41(4), Ex51(1)*, Ex51(2)*
- 得点表を利用した課題
  - Q11, Q12*, Q13(0), Q13(1), Q14(1), Q14(2), Q14(3)
- NamedRectangle のリストを利用した課題
  - (Q21(1), Q21(2), Q21(3), Q31(1)*, Q31(2)*)

(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

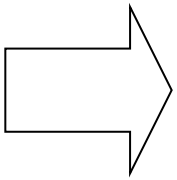
パッケージ「j2.lesson08」を作成する。

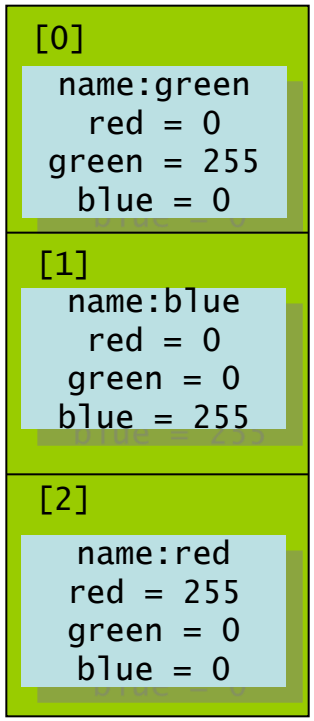
パッケージやクラスの作成，実行の仕方の説明は省略する。
作り方を忘れた場合は過去のスライドや
http://java2010.cis.k.hosei.ac.jp/01/material-01/
を参考にせよ

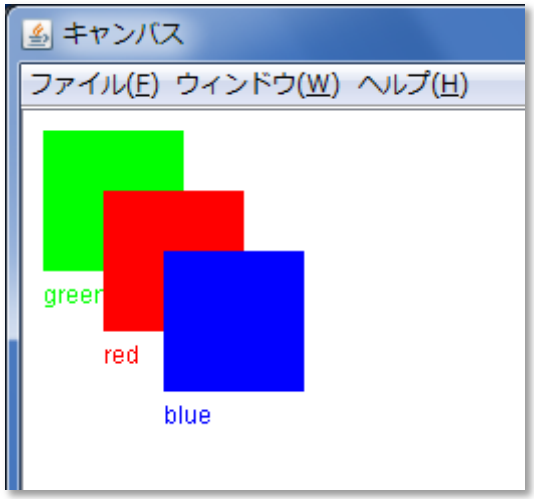
例題11～31:色のリスト

色の表を作成する。色の名前と、RGBによる16進数表現を組にして管理する。

色名	赤成分	緑成分	青成分
green	0	255	0
blue	0	0	255
red	255	0	0







■例題11

問題：次のクラスNamedColorを作成せよ。

インスタンス変数

変数の型と名前	初期値	説明
String name	無し	色の名前
int red	無し	色の赤成分
int green	無し	色の緑成分
int blue	無し	色の青成分

コンストラクタ

コンストラクタ(引数)	機能
NamedColor(String name, int red, int green, int blue)	NamedColorクラスのインスタンス変数の値がそれぞれ引数のname,red, green, blueであるNamedColorインスタンスを作成する。

（クラス名： NamedColor）

例題11

```
1  package j2.lesson08;
2
3  public class NamedColor {
4      String name;
5      int red;
6      int green;
7      int blue;
8
9      NamedColor(String name, int red, int green, int blue) {
10         this.name = name;
11         this.red = red;
12         this.green = green;
13         this.blue = blue;
14     }
15 }
```

■例題12

問題：NamedColorのインスタンスを元に、実行例のように四角形で色見本を表示せよ。

NamedColor 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	showNamedColor(int x, int y, int size)	NamedColorの赤、緑、青で座標(x,y)より辺の長さがsizeの正方形を描画し、その左下に引数cのnameをRGB色で描画する

（クラス名： Ex12ShowNamedColor）

例題12 (NamedColor)

```java
package j2.lesson08;

import gpjava.Canvas;

public class NamedColor {
    String name;
    int red;
    int green;
    int blue;

    NamedColor(String name, int red, int green, int blue) {
        this.name = name;
        this.red = red;
        this.green = green;
        this.blue = blue;
    }

    void showNamedColor(int x, int y, int size) {
        Canvas.setColor(red, green, blue);
        Canvas.fillRect(x, y, size, size);
        Canvas.drawString(x, y + size + 16, name);
    }
}
```

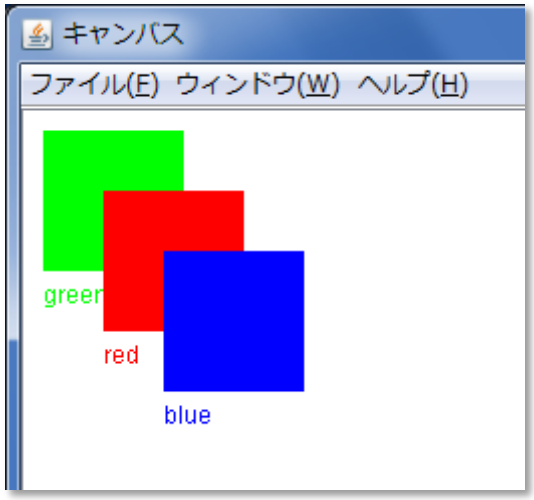
返り値の型	メソッド名(引数)	機能
void	showNamedColor(int x, int y, int size)	NamedColor のインスタンスを、指定のRGB色で、座標(x,y)より辺の長さがsizeの正方形を描画し、その左下にnameを表示する。

例題12 （Ex12ShowNamedColor）

```
 1 package j2.lesson08;
 2
 3 import gpjava.Canvas;
 4
 5 public class Ex12ShowNamedColor {
 6
 7     public static void main(String[] args) {
 8         new Ex12ShowNamedColor().start();
 9     }
10
11     void start() {
12         NamedColor c0 = new NamedColor("green", 0, 255, 0);
13         NamedColor c1 = new NamedColor("red", 255, 0, 0);
14
15         int size = 70;
16         Canvas.show();
17         c0.showNamedColor(10, 10, size);
18         c1.showNamedColor(40, 40, size);
19     }
20 }
```

■例題13(0)

問題:NamedColorのインスタンスを配列に格納するようにさせよ（この配列を色リストと呼ぶ）。



（クラス名： Ex13ShowColorList_0）

例題13(0)

```
1  package j2.lesson08;
2
3  import gpjava.Canvas;
4
5  public class Ex13ShowColorList_0 {
6      public static void main(String[] args) {
7          new Ex13ShowColorList_0().start();
8      }
9
10     void start() {
11         int numColor = 3;
12         NamedColor [] list = new NamedColor[numColor];
13         list[0] = new NamedColor("green", 0, 255, 0);
14         list[1] = new NamedColor("red", 255, 0, 0);
15         list[2] = new NamedColor("blue", 0, 0, 255);
16
17         int size = 70;
18         Canvas.show();
19         int x = 10;
20         for (int i = 0; i < list.length; i++ ) {
21             list[i].showNamedColor(x, x, size);
22             x += 30;
23         }
24     }
25 }
```

配列listにNamedColorのインスタンスを登録する。

■例題13(1)

問題：色リストに含まれるNamedColorインスタンスの色見本を並べて表示するメソッドshowListを作成せよ。

返り値の型	メソッド名(引数)	機能
void	showList(NamedColor [] list, int x, int y)	showNamedColorメソッドを用いて引数listの要素を先頭から表示する。

※既存のEx13ShowColorListを編集する

例題13(1)

```
1  package j2.lesson08;
2
3  import gpjava.Canvas;
4
5  public class Ex13ShowColorList_1 {
6      public static void main(String[] args) {
7          new Ex13ShowColorList_1().start();
8      }
9
10     void start() {
11         int numColor = 3;
12         NamedColor [] list = new NamedColor[numColor];
13         list[0] = new NamedColor("green", 0, 255, 0);
14         list[1] = new NamedColor("red", 255, 0, 0);
15         list[2] = new NamedColor("blue", 0, 0, 255);
16
17         Canvas.show();
18         showList(list, 30, 100);
19     }
20
21     void showList(NamedColor [] list, int x, int y) {
22         int size = 70;
23         for (int i = 0; i < list.length; i++ ) {
24             list[i].showNamedColor(x, y, size);
25             x += size;
26         }
27     }
28 }
```

配列listを走査して、要素となる
インスタンスをひとつずつキャンバスに表示。

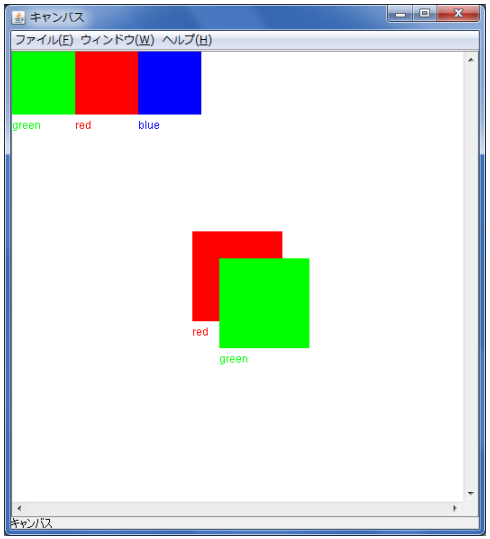
■例題21(1)

問題：次のメソッドgetを作成し、動作を確認せよ。

返り値の型	メソッド名(引数)	機能
NamedColor	get(NamedColor [ ] list, String colorName)	色リストlist中に格納されているNamedColorインスタンスのうち,引数のcolorNameと同じ名前nameを持つものを一つ返す。該当する名前が見つからない場合は、nullを返す。
既存メソッド(Ex21ColorList内)		
void showList(NamedColor [ ] list, int x, int y)		

Ex21ColorList

例題21(1)



Ex13(1)と同じ

```
1  package j2.lesson08;
2
3  import gpjava.Canvas;
4
5  public class Ex21ColorList_1 {
6      public static void main(String[] args) {
7          new Ex21ColorList_1().start();
8      }
9
10     void start() {
11         int numColor = 3;
12         NamedColor [] list = new NamedColor[numColor];
13         list[0] = new NamedColor("green", 0, 255, 0);
14         list[1] = new NamedColor("red", 255, 0, 0);
15         list[2] = new NamedColor("blue", 0, 0, 255);
16
17         int size = 100;
18         Canvas.show();
19         showList(list, 0, 0);
20         NamedColor c = get(list, "red");
21         c.showNamedColor(200, 200, size);
22         Canvas.waitForCountdown(1000);
23         c = get(list, "green");
24         c.showNamedColor(230, 230, size);
25         Canvas.waitForCountdown(1000);
26
27         c = get(list, "magenta");
28         c.showNamedColor(260, 260, size);
29         // cにはnullが入るので、エラー
30
31     }
32
33     void showList(NamedColor [] list, int x, int y) {..
40
41     NamedColor get(NamedColor [] list, String colorName) {
42         for (int i = 0; i < list.length; i++ ) {
43             if (list[i].name.equals(colorName)){
44                 return list[i];
45             }
46         }
47         return null;
48     }
49 }
```

■例題21(2)

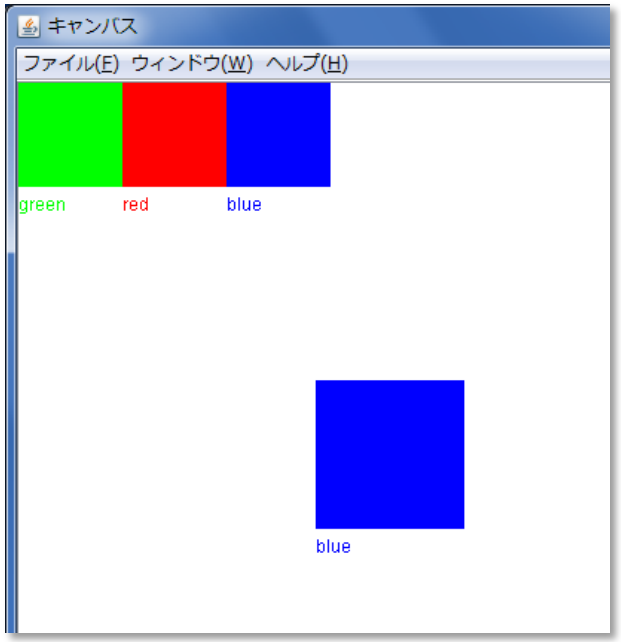
問題:ユーザーが入力した色の色見本を表示せよ。ただし、該当する名前がリスト内に見つからない場合には、メッセージを出すようにせよ。

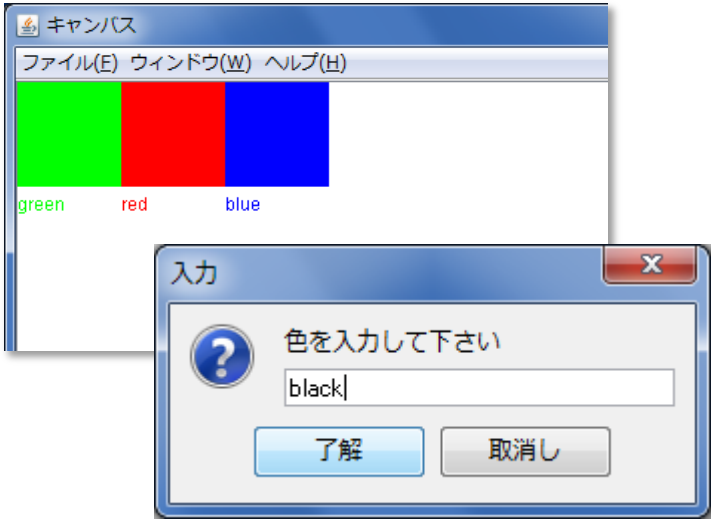
返り値の型	メソッド名(引数)	機能
NamedColor	getNamedColorFromUser(NamedColor [ ] list)	入力ダイアログで得た色名に対応するNamedColorインスタンスをgetメソッドを利用して得る。
既存メソッド(Ex21ColorList内)		
void showList(NamedColor [] list, int x, int y)		
NamedColor get(NamedColor [] list, String colorName)		

※既存のEx21ColorListを編集する

実行例


キャンバス
ファイル(F) ウィンドウ(W) ヘルプ(H)
green
red
blue
入力
色を入力して下さい
blue
了解
取消し


キャンバス
ファイル(F) ウィンドウ(W) ヘルプ(H)
green
red
blue
blue


キャンバス
ファイル(F) ウィンドウ(W) ヘルプ(H)
green
red
blue
入力
色を入力して下さい
black
了解
取消し


キャンバス
ファイル(F) ウィンドウ(W) ヘルプ(H)
green
red
blue
メッセージ
その名前の色は登録されていません
了解

例題21(2)

```
1  package j2.lesson08;
2
3  import javax.swing.JOptionPane;
4  import gpjava.Canvas;
5
6  public class Ex21ColorList_2 {
7      public static void main(String[] args) {
8          new Ex21ColorList_2().start();
9      }
10
11     void start() {
12         int numColor = 3;
13         NamedColor[] list = new NamedColor[numColor];
14         list[0] = new NamedColor("green", 0, 255, 0);
15         list[1] = new NamedColor("red", 255, 0, 0);
16         list[2] = new NamedColor("blue", 0, 0, 255);
17
18         int size = 100;
19         Canvas.show();
20         showList(list, 0, 0);
21         NamedColor c = getNamedColorFromUser(list);
22         if (c != null) {
23             c.showNamedColor(200, 200, size);
24         } else {
25             JOptionPane.showMessageDialog(null,
26                     "その名前の色は登録されていません");
27         }
28     }
29
30     void showList(NamedColor[] list, int x, int y) {..
37
38     NamedColor get(NamedColor[] list, String colorName) {..
46
47     NamedColor getNamedColorFromUser(NamedColor[] list) {
48         String colorName = JOptionPane.showInputDialog("色を入力して下さい");
49         return get(list, colorName);
50     }
51 }
```

Ex13(1)
Ex21(1)
と同じ

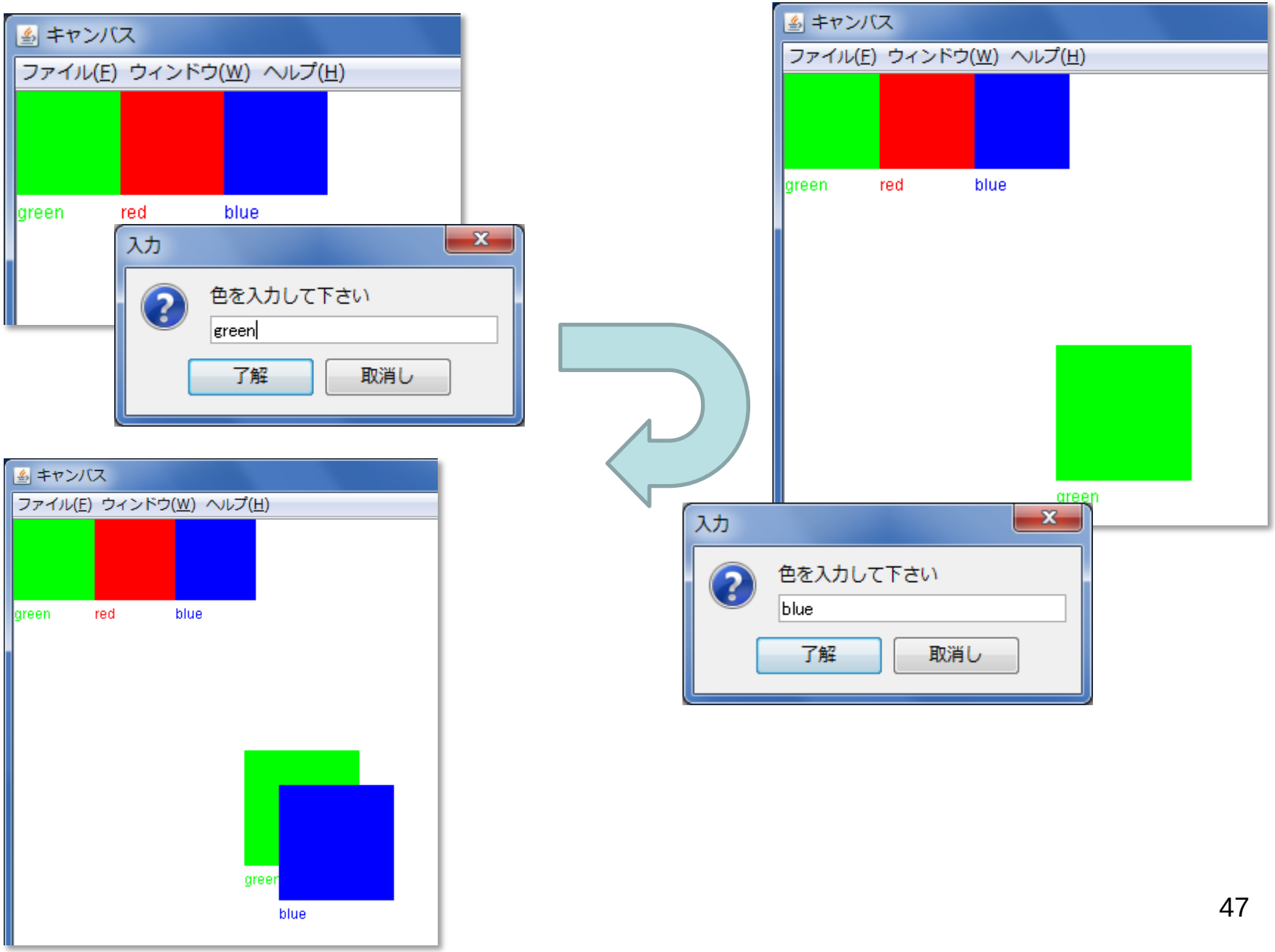
■例題21(3)

問題：ユーザーが入力した色の色見本を表示せよ。ここでは、連続で入力できるようにせよ。該当する名前がリスト内に見つからない場合に、入力を終了するものとする。

既存メソッド(Ex21ColorList内)
void showList(NamedColor [] list, int x, int y)
NamedColor get(NamedColor [] list, String colorName)
NamedColor getNamedColorFromUser(NamedColor[] list)

※既存のEx21ColorListを編集する

例題21(3) 実行例


キャンバス
ファイル(F) ウィンドウ(W) ヘルプ(H)
green
red
blue
入力
色を入力して下さい
green
了解
取消し
キャンバス
ファイル(F) ウィンドウ(W) ヘルプ(H)
green
red
blue
green
入力
色を入力して下さい
blue
了解
取消し
キャンバス
ファイル(F) ウィンドウ(W) ヘルプ(H)
green
red
blue
green
blue

例題21(3)

```
1  package j2.lesson08;
2
3  import javax.swing.JOptionPane;
4  import gpjava.Canvas;
5
6  public class Ex21ColorList_3 {
7      public static void main(String[] args) {
8          new Ex21ColorList_3().start();
9      }
10
11     void start() {
12         int numColor = 3;
13         NamedColor [] list = new NamedColor[numColor];
14         list[0] = new NamedColor("green", 0, 255, 0);
15         list[1] = new NamedColor("red", 255, 0, 0);
16         list[2] = new NamedColor("blue", 0, 0, 255);
17
18         int size = 100;
19         Canvas.show();
20         showList(list, 0, 0);
21         int x = 200;
22         NamedColor c = getNamedColorFromUser(list);
23         while (c != null) {
24             c.showNamedColor(x, x, size);
25             x += 30;
26             c = getNamedColorFromUser(list);
27         }
28     }
29
30     void showList(NamedColor [] list, int x, int y) {..
37
38     NamedColor get(NamedColor [] list, String colorName) {..
46
47     NamedColor getNamedColorFromUser(NamedColor [] list) {..
51 }
```

Ex13(1)
Ex21(1)
Ex21(2)
と同じ

例題21(3) 補足

前頁のソースより

```
22          NamedColor c = getNamedColorFromUser(list);
23          while (c != null) {
24              c.showNamedColor(x, x, size);
25              x += 30;
26              c = getNamedColorFromUser(list);
27          }
```

赤枠の部分は、次のように書くことですっきりする。
（getNamedColorFromUser メソッドの呼び出しがプログラム上で一カ所になる。）

```
NamedColor c;
while((c = getNamedColorFromUser(list)) != null) {
        c.showNamedColor(x, x, size);
        x += 30;
}
```

■例題31(1)

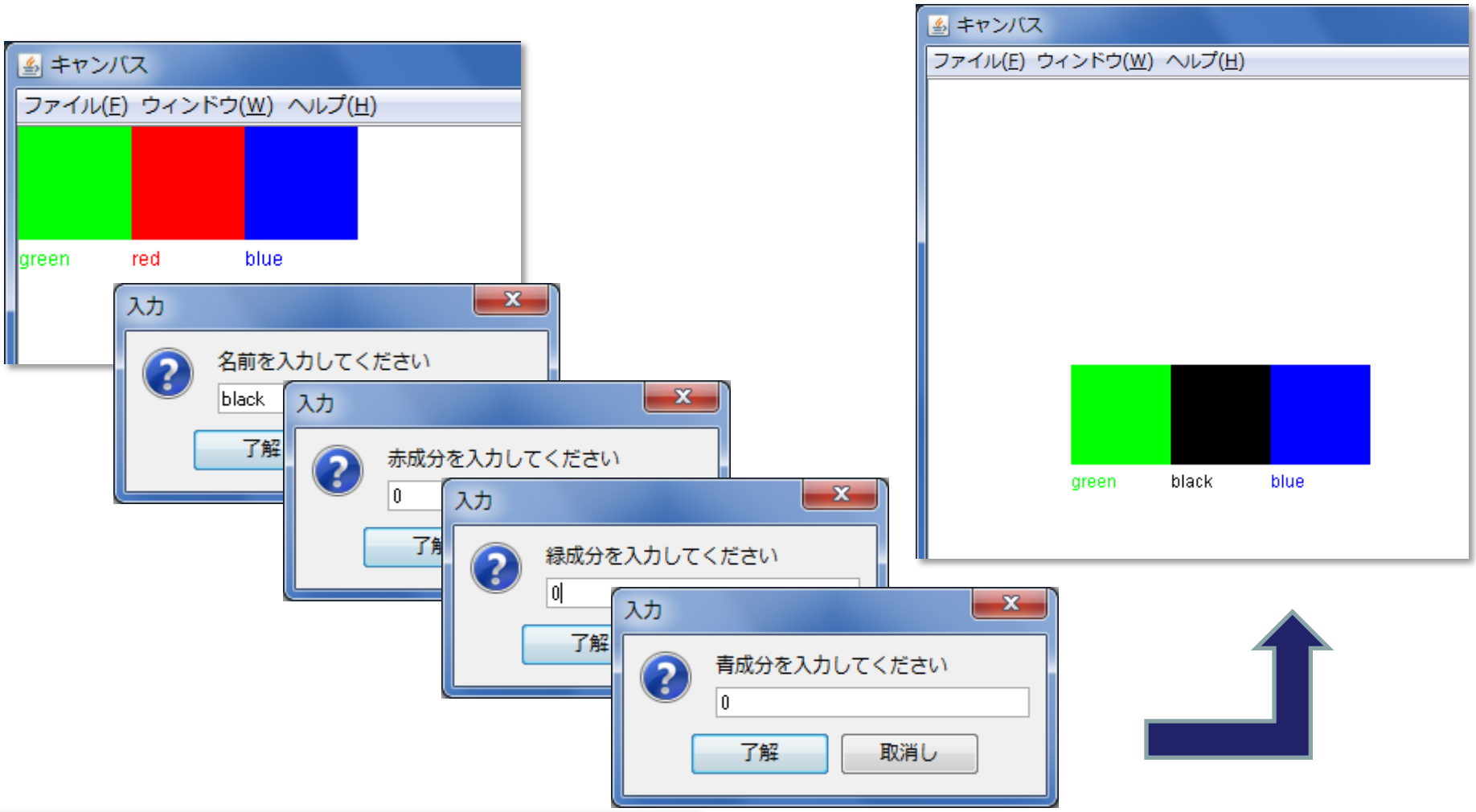
問題：NamedColorのインスタンスの値をユーザが設定するための下記のインスタンスメソッドをNamedColor内に作成し、さらに、動作を確認するためのEx31ColorListを作成せよ。

NamedColor 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	setNameFromUser()	入力ダイアログで色の名前を取得し、インスタンス変数nameに代入する
void	setColorFromUser()	入力ダイアログでRGB色を取得し、インスタンス変数red, green, blue にそれぞれ代入する

※既存のNamedColorを編集する
さらにEx31ColorListを作成

例題31(1) 実行例




```
NamedColor c = get(list, "red");
c.setNameFromUser();
c.setColorFromUser();
```

例題31(1)
NamedColor

```
1  package j2.lesson08;
2
3  import javax.swing.JOptionPane;
4  import gpjava.Canvas;
5
6  public class NamedColor {
7      String name;
8      int red;
9      int green;
10     int blue;
11
12     NamedColor(String name, int red, int green, int blue) {
13         this.name = name;
14         this.red = red;
15         this.green = green;
16         this.blue = blue;
17     }
18
19     void showNamedColor(int x, int y, int size) {
20         Canvas.setColor(red, green, blue);
21         Canvas.fillRect(x, y, size, size);
22         Canvas.drawString(x, y + size + 16, name);
23     }
24
25     void setNameFromUser() {
26         String name = JOptionPane.showInputDialog("名前を入力してください");
27         this.name = name;
28     }
29
30     void setColorFromUser() {
31         String redString = JOptionPane.showInputDialog("赤成分を入力してください");
32         String greenString = JOptionPane.showInputDialog("緑成分を入力してください");
33         String blueString = JOptionPane.showInputDialog("青成分を入力してください");
34         this.red = Integer.parseInt(redString);
35         this.green = Integer.parseInt(greenString);
36         this.blue = Integer.parseInt(blueString);
37     }
38 }
```

例題31(1) Ex31ColorList

```
1  package j2.lesson08;
2
3  import gpjava.Canvas;
4
5  public class Ex31ColorList_1 {
6      public static void main(String[] args) {
7          new Ex31ColorList_1().start();
8      }
9
10     void start() {
11         int numColor = 3;
12         NamedColor [] list = new NamedColor[numColor];
13         list[0] = new NamedColor("green", 0, 255, 0);
14         list[1] = new NamedColor("red", 255, 0, 0);
15         list[2] = new NamedColor("blue", 0, 0, 255);
16
17         Canvas.show();
18         showList(list, 0, 0);
19
20         NamedColor c = get(list, "red");
21         c.setNameFromUser();
22         c.setColorFromUser();
23
24         Canvas.clear();
25         showList(list, 100, 200);
26
27     }
28
29     void showList(NamedColor [] list, int x, int y) {
36
37     NamedColor get(NamedColor [] list, String colorName) {
45 }
```

Ex13(1)
Ex21(1)
と同じ

■例題31(2)

問題：例題31(1)のプログラムを拡張して次のようにせよ

(a) 色リストの色見本を表示させよ。

(b) ユーザに色名を入力させ色データを一つ取り出しこのデータを修正させよ。

(c) 色見本を表示しなおせ。

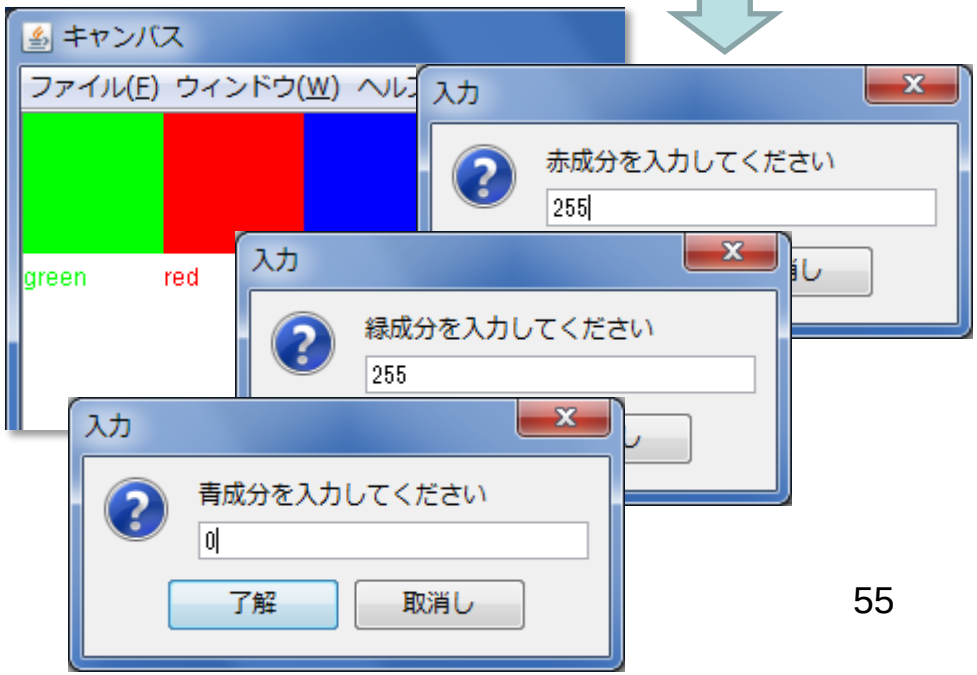
返り値の型	メソッド名(引数)	機能
NamedColor	getNamedColorFromUser(NamedColor [ ] list)	入力ダイアログで得た色名に対応するNamedColorインスタンスをgetメソッドを利用して得る。
既存メソッド(Ex31ColorList内)		
void showList(NamedColor [] list, int x, int y)		
NamedColor get(NamedColor [] list, String colorName)		

※既存のEx31ColorListを編集する

実行例


キャンバス
ファイル(F) ウィンドウ(W) ヘルプ(H)
green
red
blue
入力
色を入力して下さい
green
了解
取消し


キャンバス
ファイル(F) ウィンドウ(W) ヘルプ(H)
green
red
blue
入力
名前を入力してください
yellow
了解
取消し


キャンバス
ファイル(F) ウィンドウ(W)
green
red
入力
赤成分を入力してください
255
入力
緑成分を入力してください
255
入力
青成分を入力してください
0
了解
取消し


キャンバス
ファイル(F) ウィンドウ(W) ヘルプ(H)
yellow
red
blue

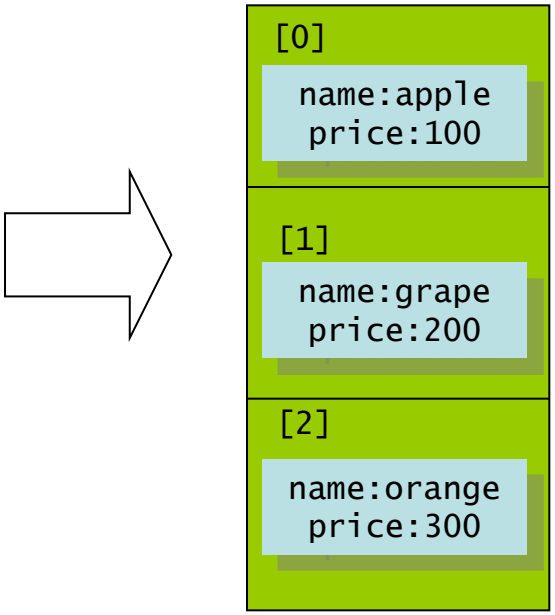
例題31(2)

```
1  package j2.lesson08;
2
3  import javax.swing.JOptionPane;
4  import gpjava.Canvas;
5
6  public class Ex31ColorList_2 {
7      public static void main(String[] args) {
8          new Ex31ColorList_2().start();
9      }
10
11     void start() {
12         int numColor = 3;
13         NamedColor [] list = new NamedColor[numColor];
14         list[0] = new NamedColor("green", 0, 255, 0);
15         list[1] = new NamedColor("red", 255, 0, 0);
16         list[2] = new NamedColor("blue", 0, 0, 255);
17
18         Canvas.show();
19         showList(list, 0, 0);
20
21         NamedColor c = getNamedColorFromUser(list);
22         c.setNameFromUser();
23         c.setColorFromUser();
24
25         Canvas.clear();
26         showList(list, 100,200);
27
28     }
29
30     void showList(NamedColor [] list, int x, int y) {..
37
38     NamedColor get(NamedColor [] list, String colorName) {..
46
47     NamedColor getNamedColorFromUser(NamedColor [] list) {
48         String colorName = JOptionPane.showInputDialog("色を入力して下さい");
49         return get(list, colorName);
50     }
51 }
```

例題41～51:商品の価格表

商品の価格表を作成する。商品の名前と、価格を組にして管理する。

商品名	単価
apple	100
grape	200
orange	300



■例題40

問題：次のクラスProductDataを作成せよ。このクラスのインスタンスは商品の情報を保持する。

インスタンス変数

変数の型と名前	初期値	説明
String name	無し	商品の名前
int price	無し	商品の単価

コンストラクタ

コンストラクタ(引数)	機能
ProductData (String name, int price)	ProductDataクラスのインスタンスを生成し、引数のname,priceを各インスタンス変数に代入する。

インスタンスメソッド

返り値の型	メソッド名(引数)	機能
void	showProductData(int row)	ProductDataインスタンスの文字列表現をSpreadsheetの row行に表示する。
void	header()	Spreadsheet のヘッダ行を表示する。

（クラス名： ProductData）

例題40

```
 1  package j2.lesson08;
 2
 3  import gpjava.Spreadsheet;
 4
 5  public class ProductData {
 6      String name;
 7      int price;
 8
 9      ProductData(String name, int price) {
10          this.name = name;
11          this.price = price;
12      }
13
14      void showProductData(int row) {
15          Spreadsheet.setString(row, 0, name);
16          Spreadsheet.setInt(row, 1, price);
17      }
18
19      void header() {
20          Spreadsheet.setString(0, 0, "商品名");
21          Spreadsheet.setString(0, 1, "単価");
22      }
23  }
24
```

■例題41(1)

問題: ProductDataのインスタンスを配列に格納するようにせよ。さらに以下のメソッドをEx41ProductListの中に作成し、動作を確認せよ。

返り値の型	メソッド名(引数)	機能
void	contents(ProductData [ ] list)	listの各ProductDataインスタンスの文字列表現を表示する。

(クラス名： Ex41ProductList)

例題41(1)

```
1  package j2.lesson08;
2
3  import gpjava.Spreadsheet;
4
5  public class Ex41ProductList_1 {
6
7      public static void main(String[] args) {
8          new Ex41ProductList_1().start();
9      }
10
11     void start() {
12         int numProducts = 3;
13         ProductData [] list = new ProductData[numProducts];
14         list[0] = new ProductData("apple", 100);
15         list[1] = new ProductData("grape", 200);
16         list[2] = new ProductData("orange", 300);
17
18         Spreadsheet.show();
19         list[0].header();
20         contents(list);
21     }
22
23     void contents(ProductData [] list) {
24         for (int i = 0; i < list.length; i++ ) {
25             list[i].showProductData(i + 1);
26         }
27     }
28 }
```

表(配列)を走査し、要素の内容の文字列表現をresultに加えていく

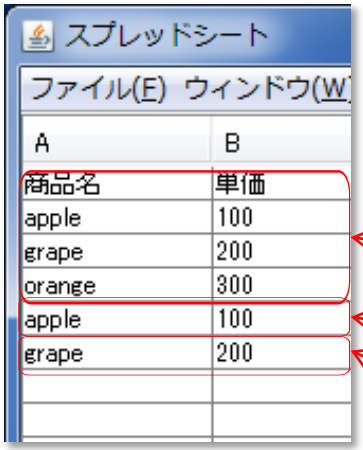
■例題41(2)

問題:次のメソッドgetを作成し、動作を確認せよ。

返り値の型	メソッド名(引数)	機能
ProductData	get(ProductData [ ] list, String name)	list中に格納されているProductDataインスタンスのうち,引数のnameと同じ名前nameを持つものを返す。該当する名前が見つからない場合は、nullを返す。
既存メソッド(Ex41ProductList内)		
String contents(ProductData [ ] list)		

※既存のEx41ProductListを編集する

例題41(2)



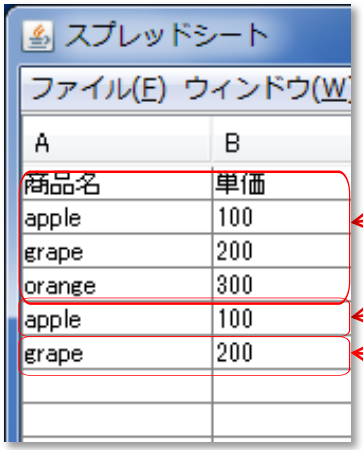

```java
package j2.lesson08;

import gpjava.Canvas;
import gpjava.Spreadsheet;

public class Ex41ProductList_2 {

    public static void main(String[] args) {
        new Ex41ProductList_2().start();
    }

    void start() {
        int numProducts = 3;
        ProductData[] list = new ProductData[numProducts];
        list[0] = new ProductData("apple", 100);
        list[1] = new ProductData("grape", 200);
        list[2] = new ProductData("orange", 300);

        Spreadsheet.show();
        Canvas.show();
        list[0].header();
        contents(list);

        ProductData p = get(list, "apple");
        p.showProductData(list.length + 1);

        Canvas.waitForCountdown(1000);
        p = get(list, "grape");
        p.showProductData(list.length + 2);

        Canvas.waitForCountdown(1000);
        p = get(list, "cherry");
        p.showProductData(list.length + 3);
        // cにはnullが入るので、エラー
    }

        for (int i = 0; i < list.length; i++) {
            list[i].showProductData(i + 1);
        }
    }

    ProductData get(ProductData[] list, String name) {
        for (int i = 0; i < list.length; i++) {
            if (list[i].name.equals(name)) {
                return list[i];
            }
        }
        return null;
    }

}
```

コンソール　アウトライン

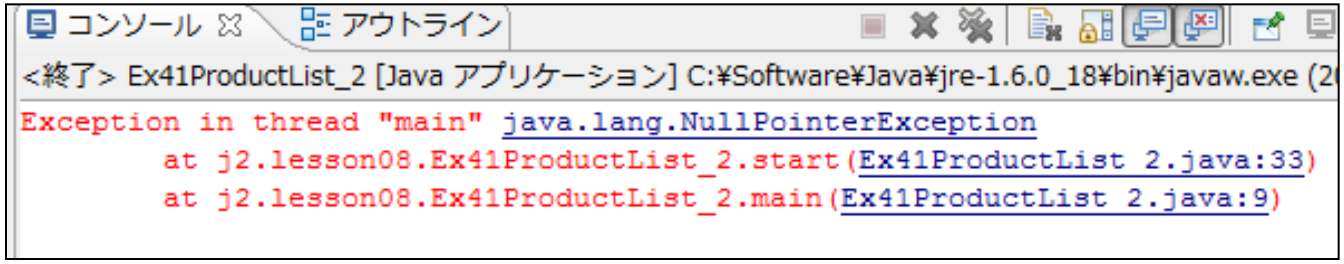
<終了> Ex41ProductList_2 [Java アプリケーション] C:¥Software¥Java¥jre-1.6.0_18¥bin¥javaw.exe (2

```
Exception in thread "main" java.lang.NullPointerException
        at j2.lesson08.Ex41ProductList_2.start(Ex41ProductList_2.java:33)
        at j2.lesson08.Ex41ProductList_2.main(Ex41ProductList_2.java:9)
```

例題41(2)実行の流れ （前頁のものを拡大）



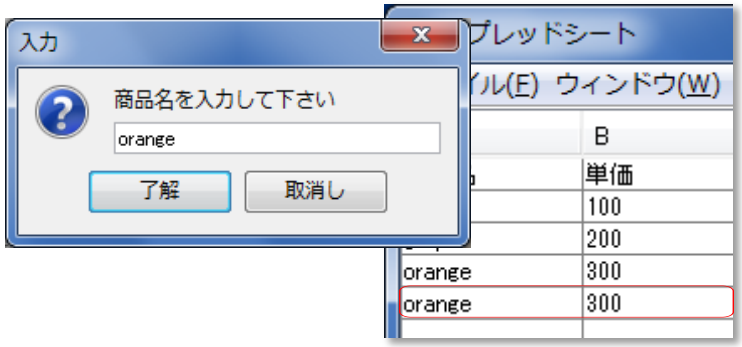

```
11
12    void start() {
13        int numProducts = 3;
14        ProductData[] list = new ProductData[numProducts];
15        list[0] = new ProductData("apple", 100);
16        list[1] = new ProductData("grape", 200);
17        list[2] = new ProductData("orange", 300);
18
19        Spreadsheet.show();
20        Canvas.show();
21        list[0].header();
22        contents(list);
23
24        ProductData p = get(list, "apple");
25        p.showProductData(list.length + 1);
26
27        Canvas.waitForCountdown(1000);
28        p = get(list, "grape");
29        p.showProductData(list.length + 2);
30
31        Canvas.waitForCountdown(1000);
32        p = get(list, "cherry");
33        p.showProductData(list.length + 3);
34        // p にはnullが入るので、エラー
35    }
```

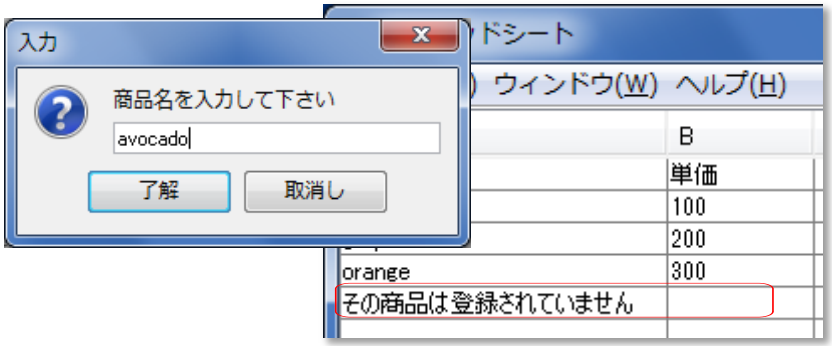


■例題41(3)

問題:ユーザーに商品名を入力させ、その商品情報を表示せよ。ただし、該当する名前がリスト内に見つからない場合には、メッセージを出すようにせよ。

返り値の型	メソッド名(引数)	機能
ProductData	getFromUser(ProductData[ ] list)	入力ダイアログで得た商品名に対応するProductDataインスタンスをgetメソッドを利用して得る。
既存メソッド(Ex41ProductList内)		
String contents(ProductData [ ] list)		
ProductData get(ProductData [ ] list, String name)		





※既存のEx41ProductListを編集する

例題41(3)

```
1  package j2.lesson08;
2
3  import gpjava.Spreadsheet;
4  import javax.swing.JOptionPane;
5
6  public class Ex41ProductList_3 {
7      public static void main(String[] args) {
8          new Ex41ProductList_3().start();
9      }
10
11     void start() {
12         int numProducts = 3;
13         ProductData[] list = new ProductData[numProducts];
14         list[0] = new ProductData("apple", 100);
15         list[1] = new ProductData("grape", 200);
16         list[2] = new ProductData("orange", 300);
17
18         Spreadsheet.show();
19         list[0].header();
20         contents(list);
21
22         ProductData p = getFromUser(list);
23         if (p != null) {
24             p.showProductData(list.length + 1);
25         } else {
26             Spreadsheet.setString(list.length +1, 0, "その商品は登録されていません");
27         }
28     }
29
30     void contents(ProductData [] list) {..
35
36     ProductData get(ProductData [] list, String name) {..
44
45     ProductData getFromUser(ProductData[] list) {
46         String name = JOptionPane.showInputDialog("商品名を入力して下さい");
47         return get(list, name);
48     }
49 }
```

Ex41(2)と同じ

■例題41(4)

問題:ユーザーに商品名を入力させ、その商品情報を表示せよ。ここでは、連続で入力できるようにせよ。該当する商品名がリスト内に見つからない場合に、入力を終了するものとする。

既存メソッド(Ex41ProductList内)
String contents(ProductData [ ] list)
ProductData get(ProductData [ ] list, String name)
ProductData getFromUser(ProductData[] list)

※既存のEx41ProductListを編集する

例題41(4)

```
1  package j2.lesson08;
2
3  import gpjava.Spreadsheet;
4  import javax.swing.JOptionPane;
5
6  public class Ex41ProductList_4 {
7      public static void main(String[] args) {
8          new Ex41ProductList_4().start();
9      }
10
11     void start() {
12         int numProducts = 3;
13         ProductData[] list = new ProductData[numProducts];
14         list[0] = new ProductData("apple", 100);
15         list[1] = new ProductData("grape", 200);
16         list[2] = new ProductData("orange", 300);
17
18         Spreadsheet.show();
19         list[0].header();
20         contents(list);
21
22         ProductData p = getFromUser(list);
23
24         int index = list.length + 1;
25         while (p != null) {
26             p.showProductData(index);
27             index++;
28             p = getFromUser(list);
29         }
30     }
31
32     void contents(ProductData[] list) {..
37
38     ProductData get(ProductData[] list, String name) {..
46
47     ProductData getFromUser(ProductData[] list) {..
51 }
```

Ex41(2)
Ex41(3)
と同じ

■例題51(1)

問題：ProductDataのインスタンスの値をユーザが設定するための下記のメソッドをProductData内に作成し、さらに、動作を確認するためのEx51ProductListを作成せよ。

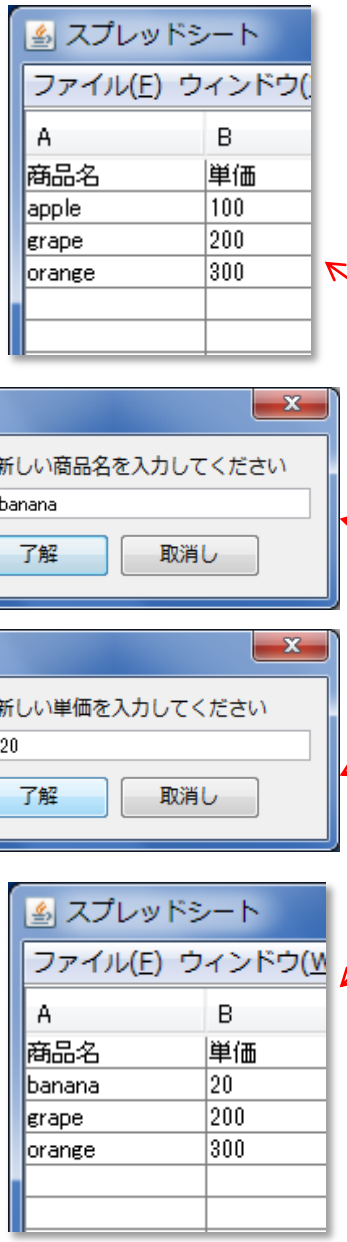
返り値の型	メソッド名(引数)	機能
void	setNameFromUser()	入力ダイアログで文字列を取得し、インスタンス変数nameに代入する
void	setPriceFromUser()	入力ダイアログで文字列を取得し、intに変換して引数pのインスタンス変数priceに代入する

※既存のProductDataを編集する
さらにEx51ProductListを作成

例題51(1) クラスProductData

```
1  package j2.lesson08;
2
3  import gpjava.Spreadsheet;
4
5  import javax.swing.JOptionPane;
6
7  public class ProductData {
8      String name;
9      int price;
10
11     ProductData(String name, int price) {
12         this.name = name;
13         this.price = price;
14     }
15
16     void showProductData(int row) {
17         Spreadsheet.setString(row, 0, name);
18         Spreadsheet.setInt(row, 1, price);
19     }
20
21     void header() {
22         Spreadsheet.setString(0, 0, "商品名");
23         Spreadsheet.setString(0, 1, "単価");
24     }
25
26     void setNameFromUser() {
27         String name = JOptionPane.showInputDialog("新しい商品名を入力してください");
28         this.name = name;
29     }
30
31     void setPriceFromUser() {
32         String inputPrice = JOptionPane.showInputDialog("新しい単価を入力してください");
33         this.price = Integer.parseInt(inputPrice);
34     }
35 }
```

例題51(1)
クラスEx51ProductList




```
1  package j2.lesson08;
2
3  import gpjava.Spreadsheet;
4
5  public class Ex51ProductList_1 {
6      public static void main(String[] args) {
7          new Ex51ProductList_1().start();
8      }
9
10     void start() {
11         int numProducts = 3;
12         ProductData [] list = new ProductData[numProducts];
13         list[0] = new ProductData("apple", 100);
14         list[1] = new ProductData("grape", 200);
15         list[2] = new ProductData("orange", 300);
16
17         Spreadsheet.show();
18         list[0].header();
19         contents(list);
20
21         ProductData p = get(list, "apple");
22         p.setNameFromUser();
23         p.setPriceFromUser();
24
25         contents(list);
26     }
27
28     void contents(ProductData[] list) {
29         for (int i = 0; i < list.length; i++) {
30             list[i].showProductData(i + 1);
31         }
32     }
33
34     ProductData get(ProductData[] list, String name) {
35         for (int i = 0; i < list.length; i++) {
36             if (list[i].name.equals(name)) {
37                 return list[i];
38             }
39         }
40         return null;
41     }
42 }
```

■例題51(2)

問題：例題51(1)のプログラムを拡張して次のようにせよ。

(a) 商品リストの情報を表示させよ。

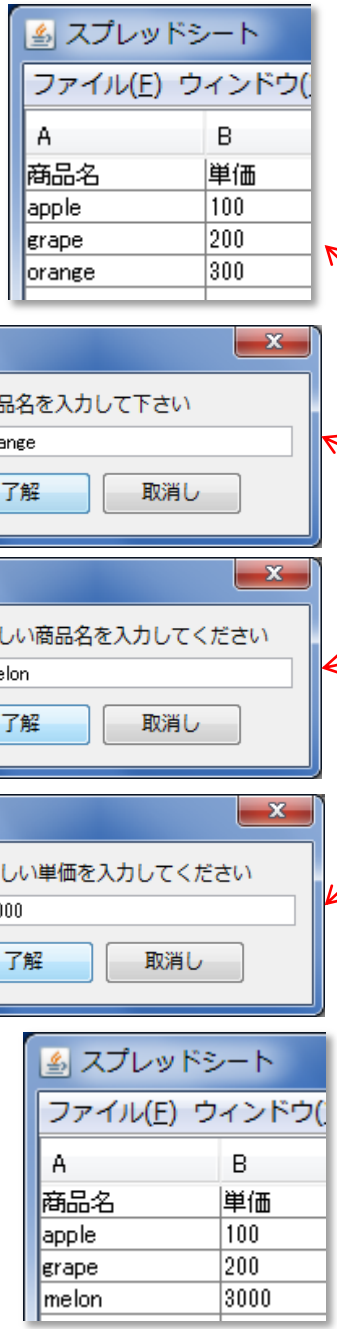
(b) ユーザに商品名を入力させProductDataのインスタンスを一つ取り出し、このデータを修正させよ。

(c) 商品リストを表示しなおせ

返り値の型	メソッド名(引数)	機能
ProductData	getFromUser(ProductData[ ] list)	入力ダイアログで得た商品名に対応するProductDataインスタンスをgetメソッドを利用して得る。
既存メソッド(Ex51ProductList内)		
String contents(ProductData [ ] list)		
ProductData get(ProductData [ ] list, String name)		

※既存のEx51ProductListを編集する

■例題51(2)




```
1  package j2.lesson08;
2
3  import javax.swing.JOptionPane;
4  import gpjava.Spreadsheet;
5
6  public class Ex51ProductList_2 {
7      public static void main(String[] args) {
8          new Ex51ProductList_2().start();
9      }
10
11     void start() {
12         int numProducts = 3;
13         ProductData [] list = new ProductData[numProducts];
14         list[0] = new ProductData("apple", 100);
15         list[1] = new ProductData("grape", 200);
16         list[2] = new ProductData("orange", 300);
17
18         Spreadsheet.show();
19         list[0].header();
20         contents(list);
21
22         ProductData p = getFromUser(list);
23         p.setNameFromUser();
24         p.setPriceFromUser();
25
26         contents(list);
27     }
28
29     void contents(ProductData[] list) {
30         for (int i = 0; i < list.length; i++) {
31             list[i].showProductData(i + 1);
32         }
33     }
34
35     ProductData get(ProductData[] list, String name) {
36         for (int i = 0; i < list.length; i++) {
37             if (list[i].name.equals(name)) {
38                 return list[i];
39             }
40         }
41         return null;
42     }
43
44     ProductData getFromUser(ProductData[] list) {
45         String name = JOptionPane.showInputDialog("商品名を入力して下さい");
46         return get(list, name);
47     }
48 }
```