

## オブジェクト指向編 正誤表

これは第 1 刷のための正誤表です。

V～VI ページ	V ページ下段～VI ページにかけての「サポートウェブと補足資料」にある文章を次のように訂正 サポートウェブの URL を巻末資料で確認し、サポートデータをダウンロードしてください。通過テストの解答と解答用紙、プログラム演習の解答などがあります。また、巻末に URL を掲載している著者のウェブでは JDK の最新版や日本語化済みの Eclipse 最新版、関連ドキュメントをダウンロードできます。ぜひ一度アクセスしてください。
----------	---

### [ 1 章 ]

8 ページ	注釈の枠内 3 行目 誤… Eclipse の使い方は <u>巻末資料</u> にあります 正… Eclipse の使い方はサポートウェブから PDF ファイルをダウンロードできます
-------	---

### [ 2 章 ]

25 ページ 練習 2-1	表の中 誤… tips 正… chips
46 ページ コラム	12 行目 誤… た」とう意見は 正… た」という意見は

### [ 3 章 ]

47 ページ 扉の図	誤… p=new Product("cx-10"); 正… p=new Product();
48 ページ 本文 1 行目	誤… <u>クラス変数</u> の仕組みが明らかになりましたので、 正… <u>クラス</u> の仕組みが明らかになりましたので、
56 ページ 本文 1～2 行目	赤い線を引いた部分を削除してください  これまでクラスやその構成要素であるフィールド、コンストラクタ、メソッドに public や private を付けてアクセス条件を設定しました。これを <b>アクセス修飾子</b> といいます。「アクセスできる」とは「見える(可視)」ということです。見えていればそれらを利用することができます。

## 63 ページ

## 例題 8

## 7行目

誤… if(code!=null) code = c;正… if(c!=null) code = c;

2行目と3行目の間に引数のないコンストラクタを追加（例題9で使います）

`public Product() {}`

誤

```
class Product {
    private String code;           // 商品番号
    public Product(String code) {
        this.code = code;
    }
    public void setCode(String c) { // セッターメソッド
        if (code != null) code = c;
    }
    public String getCode() {      // ゲッターメソッド
        return code;
    }
}
```

正

```
class Product {
    private String code;           // 商品番号
    public Product() {}           // デフォルトコンストラクタ
    public Product(String code) {
        this.code = code;
    }
    public void setCode(String c) { // セッターメソッド
        if (c != null) code = c;
    }
    public String getCode() {      // ゲッターメソッド
        return code;
    }
}
```

## 64 ページ

## 例題 9

例題9のプログラムの下線部分を訂正（“cx10”を削除、p.getCode()はcに訂正）

誤

```
public class UseProduct {
    public static void main(String[] args){
        Product p = new Product("cx10"); // オブジェクトを作成し参照を得る
        p.setCode("cx10");                // 商品番号をセットする
        String c = p.getCode();           // 商品番号を取得する
        System.out.println(p.getCode()); // 表示
    }
}
```

正

```
public class UseProduct {
    public static void main(String[] args){
        Product p = new Product();        // オブジェクトを作成し参照を得る
        p.setCode("cx10");                // 商品番号をセットする
        String c = p.getCode();           // 商品番号を取得する
        System.out.println(c);           // 表示
    }
}
```

## 65 ページ

## 説明図

説明図 最上段のコンストラクタの引数を削除する

誤… p = new Product("cx10");

正… p = new Product();

<p><b>66</b> ページ 練習 3-4</p>	<p>2. のプログラム 6 行目のメソッド名に s が抜けている 誤… setSecond(int s) 正… setSeconds(int s);</p>
<p><b>69</b> ページ 説明図</p>	<p>説明図 最上段のコンストラクタの引数を削除する 誤… p = new Product("cx10"); 正… p = new Product();</p>
<p><b>71</b> ページ 通過テスト 4</p>	<pre> 誤 0   import ex3.p1.Dog; 1   package ex3.p1; 2   public class Dog { 3       private String name; 4       private String getName(){return this.name; } 5       void    setName(String s) {name = s;} 6   }  1   package ex3.p2; 2   public class Exec { 3       public static void main(String[] args) { 4           Dog d = new Dog(); 5           d.setName("トム"); 6       } 7   }  正 1   package ex3.p1; 2   public class Dog { 3       private String name; 4       private String getName(){return this.name; } 5       void    setName(String s) {name = s;} 6   }  0   package ex3.p2; 1   import ex3.p1.Dog; 2   public class Exec { 3       public static void main(String[] args) { 4           Dog d = new Dog(); 5           d.setName("トム"); 6       } 7   } </pre> <p>Exec クラスの package 文の下に移動する</p>

[4章]

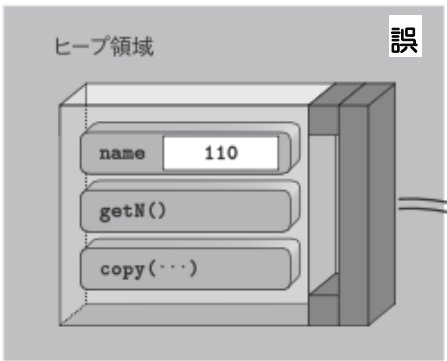
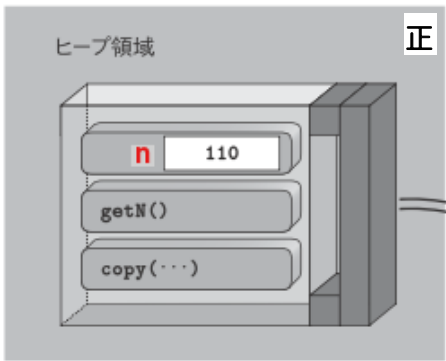
<p>74 ページ 下段の図</p>	<p>点線枠がずれていました</p> 
<p>84 ページ</p>	<p>1 行目 誤… ShowBook クラスは sample20 パッケージにあり、Book クラスは sample21 パッケージにあります。 正… ShowBook クラスは sample21 パッケージにあり、Book クラスは sample20 パッケージにあります。</p>
<p>94 ページ 通過テスト 4.1</p>	<p>1 行目 誤… 次の中で正しいものはどれかすべて答えなさい。 正… 次の中で正しいものはどれか <u>答えなさい</u> (正解はひとつ) 。</p>
<p>95 ページ 通過テスト 4.3</p>	<p>1 行目 誤… <u>1</u> 正… 1 と表示する</p>
<p>95 ページ 通過テスト 4.4</p>	<p>4 行目 cat クラスのコンストラクタ…先頭の c を大文字 C にする 誤… public <u>cat</u>( <input style="width: 100px;" type="text" value="②"/> ); 正… public <b>cat</b>( <input style="width: 100px;" type="text" value="②"/> );</p>

[5章]

<p>112 ページ 練習 5-3</p>	<p>選択肢の下線を引いた部分を以下のように訂正してください。</p> <div style="border: 1px solid black; padding: 10px;"> <pre> A. Tool(int n, double w){ number=n; weight=w; } B. Tool(){ } C. public Tool(int n){ number=n; } D. <u>publ</u>ic tool(double w, int n) { number=n; weight=w; } E. public void Tool(int n, double w){ number=n; weight=w; } F. public Tool(double w) { number=0; weight=w; } G. public Tool(String s) { number=0; <u>weight</u>=s; } H. public Tool(char a, double b) { number=a; weight=b; } </pre> <p>【解答】 _____</p> </div>
---------------------------	--

<b>122</b> ページ まとめ	「コンストラクタの基本」の2行目 誤… □クラスと同じ名前を持ち戻り値を書かない 正… □クラスと同じ名前を持ち、戻り値を書かない
<b>124</b> ページ 通過テスト2	プログラム9行目 変数名の誤り <code>this.n</code> → <code>this.size</code> 誤… <code>System.out.println(this.n);</code> 正… <code>System.out.println(this.size);</code>

## [6章]

<b>128</b> ページ 例題1	Dog クラスの定義の最後 (同じクラスなのでメソッドを使わなくても直接アクセスできる) 誤… <code>System.out.println(name+"の友達は"+other.getName());</code> 正… <code>System.out.println(name+"の友達は"+other.name);</code>
<b>134</b> ページ 説明図	図のフィールド名を訂正 誤… <u>name</u> 正… <u>n</u> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;"> <p>copyメソッドで作成されたオブジェクト</p> <p>ヒープ領域</p>  <p>誤</p> </div> <div style="text-align: center;"> <p>copyメソッドで作成されたオブジェクト</p> <p>ヒープ領域</p>  <p>正</p> </div> </div>
<b>140</b> ページ 練習6-3の1.	問題文の例題番号を1→3に訂正 誤… (1) 要素が3のShoesクラスの配列を作成しなさい。配列の作成には例題1の方法を使いなさい。 正… (1) 要素が3のShoesクラスの配列を作成しなさい。配列の作成には例題3の方法を使いなさい。
<b>141</b> ページ 練習6-3の2.	問題文例題番号を2→4に訂正 誤… (1) 例題2に示した初期値を与える方法で、 正… (1) 例題4に示した初期値を与える方法で、
<b>147</b> ページ 通過テスト2の問2	問題文を分かりやすくするために修正 問2 Pointクラスを使って次の計算をし、結果を表示するプログラムを作成しなさい。クラス名はExec2とします。 <div style="text-align: center;">↓</div> 問2 Pointクラスを使って次の計算をし、問1で作成したtoStringメソッドを使ってp0を表示するプログラムを作成しなさい。クラス名はExec2とします。

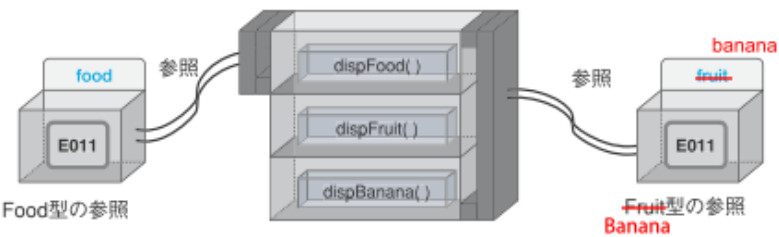
[7章]

<p>153 ページ 例題 1</p>	<p>ExecMyAdder クラスの網掛けの位置を訂正してください (ma.getX()に網掛けします)</p> <p>誤… System.out.println(ma.getX()); System.out.println(ma.getM());</p> <p>正… System.out.println(ma.getX()); System.out.println(ma.getM());</p>																									
<p>172 ページ 練習 7-4 の 1</p>	<p>問題文 2 行目</p> <p>誤… Tree クラスのオブジェクトを作成する</p> <p>正… Three クラスのオブジェクトを作成する</p>																									
<p>176 ページ 練習 7-5 の 1</p>	<p>問題文 4 行目の下線部分を削除してください (⑤は欠番になります)</p> <p>誤… これに対して private を付けると [⑤]。見えるのは [⑥] である。</p> <p>正… これに対して private を付けると見えるのは [⑥] である。</p> <p>選択肢 J を削除してください</p> <p>誤… J. 同一パッケージのクラスからは見える</p> <p>正… <del>J. 同一パッケージのクラスからは見える</del> (削除)</p>																									
<p>178 ページ まとめ</p>	<p>「クラスの階層」</p> <p>誤… コンストラクタと静的メンバはオブジェクトではないので</p> <p>正… コンストラクタと静的メンバはオブジェクトには含まれないので</p>																									
<p>179 ページ まとめ</p>	<p>「継承とデフォルトコンストラクタ」 図に以下の説明を追加</p> <p>図は、例題 2 の Calculator クラスで、Adder ← MyAdder ← Calculator と継承している</p>																									
<p>179 ページ まとめ</p>	<p>「弱いカプセル化」 説明を補足</p> <p>誤… <input type="checkbox"/> サブクラスに限ってアクセスを認めるのは protected 修飾子</p> <p>正… <input type="checkbox"/> 別のパッケージであってもサブクラスに限ってアクセスを認めるのは protected 修飾子</p> <p>表の訂正</p> <p>誤… 同じパッケージのサブクラスから 同じパッケージのサブクラス以外のクラスから</p> <p>誤… 別のパッケージのサブクラスから 別のパッケージのサブクラス以外のクラスから</p> <div data-bbox="443 1464 1289 1809" style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p style="text-align: center; color: #0070c0;">● 弱いカプセル化 ●</p> <p>【アクセス修飾子一覧】</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr style="background-color: #d9e1f2;"> <th style="width: 40%;">可視性</th> <th style="width: 10%;">public</th> <th style="width: 10%;">protected</th> <th style="width: 10%;">デフォルト</th> <th style="width: 10%;">private</th> </tr> </thead> <tbody> <tr> <td>同じクラスから</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>同じパッケージの他のクラスから</td> <td>○</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td><span style="color: red;">別の</span> 同じパッケージのサブクラスから</td> <td>○</td> <td>○</td> <td>×</td> <td>×</td> </tr> <tr> <td><span style="color: red;">別の</span> 同じパッケージのサブクラス以外のクラスから</td> <td>○</td> <td>×</td> <td>×</td> <td>×</td> </tr> </tbody> </table> <p><input type="checkbox"/> サブクラスに限ってアクセスを認めるのはprotected修飾子</p> </div>	可視性	public	protected	デフォルト	private	同じクラスから	○	○	○	○	同じパッケージの他のクラスから	○	○	○	×	<span style="color: red;">別の</span> 同じパッケージのサブクラスから	○	○	×	×	<span style="color: red;">別の</span> 同じパッケージのサブクラス以外のクラスから	○	×	×	×
可視性	public	protected	デフォルト	private																						
同じクラスから	○	○	○	○																						
同じパッケージの他のクラスから	○	○	○	×																						
<span style="color: red;">別の</span> 同じパッケージのサブクラスから	○	○	×	×																						
<span style="color: red;">別の</span> 同じパッケージのサブクラス以外のクラスから	○	×	×	×																						

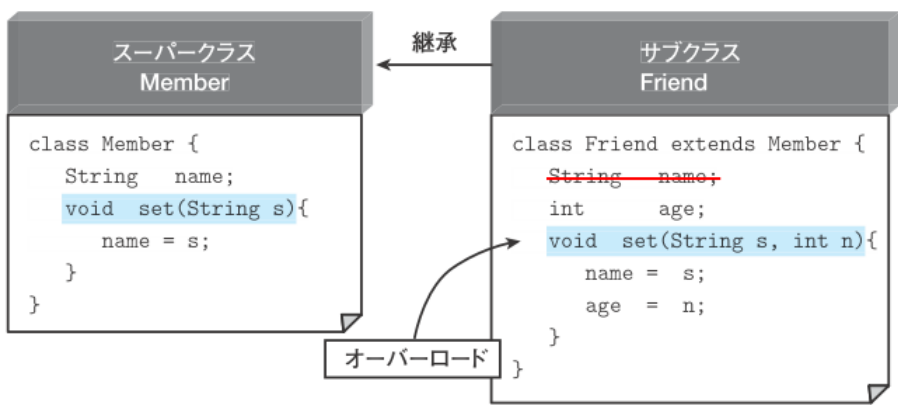
<p><b>182</b> ページ 通過テスト 4</p>	<p>選択肢の行番号を訂正 (11→12, 12→13)</p> <p>誤… A. hello と表示する B. 9 行目でコンパイルエラー C. <u>11</u> 行目でコンパイルエラー D. <u>12</u> 目でコンパイルエラー E. <u>11</u> 行目で実行時例外 F. <u>12</u> 行目で実行時例外</p> <p>正… A. hello と表示する B. 9 行目でコンパイルエラー C. <b>12</b> 行目でコンパイルエラー D. <b>13</b> 行目でコンパイルエラー E. <b>12</b> 行目で実行時例外 F. <b>13</b> 行目で実行時例外</p>
<p><b>182</b> ページ 通過テスト 4</p>	<p>ソースコード 12 行目 cat の c を大文字にする</p> <p>誤… cat tom = new Cat(); 正… <b>C</b>at tom = new Cat();</p>

**[8 章]**

<p><b>189</b> ページ 下段の図</p>	<p>図の説明を訂正</p> <p>誤… <u>food は全体を参照する</u> 正… <b>food が参照する部分は実体がある</b></p> <div data-bbox="539 958 1321 1279" data-label="Diagram"> </div>
<p><b>192</b> ページ 練習 8-2 の 3</p>	<p>プログラム 9 行目</p> <p>誤… public static void main(String[] <u>args</u>){ 正… public static void main(String[] <b>args</b>){</p> <p>注 : main メソッドの引数名は慣用的に args が用いられますが、文法的には<u>どんな名前の変数を使っても構いません</u>。この訂正はミスプリントの訂正です。</p>

<p><b>193</b> ページ 図の説明</p>	<p>図の説明の中の Fruit 型を Banana 型に訂正してください  誤… <u>fruit</u>                    誤… <u>Fruit</u> 型の参照  正… <u>banana</u>                    正… <u>Banana</u> 型の参照</p> <p style="text-align: center;">オブジェクトはひとつだけでも、いろいろな参照が可能</p> 
<p><b>202</b> ページ まとめ</p>	<p>「instanceof 演算子」説明をわかりやすくするために訂正  誤… □変数が参照するオブジェクトの本当の型は何かを調べることができる  正… □変数が参照するオブジェクトの本当の型が指定の型かまたはそのサブクラスであるかどうかを調べることができる</p>

[9章]

<p><b>208</b> ページ 上段の説明図</p>	<p>図右側の String name; を削除してください</p> 
<p><b>212</b> ページ 練習 9-1 の 3</p>	<p>Main メソッドのコメントを訂正  誤… tools.display(x) // 引数は double  tools.display(str) // 引数は String  正… tools.display(x) // 引数は double の配列型  tools.display(str) // 引数は String の配列型</p>
<p><b>216</b> ページ 本文 最下行</p>	<p>誤… オーバーライトメソッド  正… オーバーライドメソッド</p>
<p><b>221</b> ページ 練習 9-3 の 1</p>	<p>青い字の注釈 (c を大文字の C に)  誤… color  正… Color</p>
<p><b>236</b> ページ 本文 下から 5 行目</p>	<p>誤… Porker クラスが Game クラスを継承して、  正… Porker クラスが CardGame クラスを継承して、</p>
<p><b>239</b> ページ</p>	<p>誤… data フールドをアクセスしています。</p>



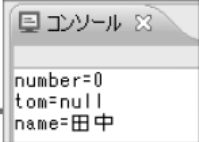
本文の中ほど	正… data <b>フィールド</b> をアクセスしています。
<b>245</b> ページ まとめ 多態性の項	誤… オーバーライトメソッドの場合、 正… オーバーライドメソッドの場合、

**[10 章]**

<b>255</b> ページ 練習 10-1 問 3	誤… ・作成したオブジェクトの参照 <b>r</b> を使って 正… ・作成したオブジェクトの参照 <b>mr</b> を使って
<b>259</b> ページ 本文 1 行目	誤… <u>抽象クラス</u> だけからなるのが特徴で、 正… <b>定数や抽象メソッド</b> だけからなるのが特徴で、
<b>274</b> ページ まとめ	「インタフェースとは」 誤… □クラスに準じる存在で、メソッドと定数だけを定義できる 正… □クラスに準じる存在で、 <b>抽象メソッド</b> と定数だけを定義できる
<b>276</b> ページ 通過テスト 1	問 2 の問題文をわかりやすくするために、末尾に指示を追加 main メソッドを持つ Exec クラスを作成し、次の SPD に示す処理を実行しなさい。なお、 <b>Factory クラスのオブジェクトを作成するにあたって、製品名や価格は任意に決めてよい。</b>
<b>277</b> ページ 通過テスト 2	解答が複数あることを明示するため問題文に「すべて」を挿入 …コンパイルエラーにならないコードを <b>すべて</b> 答えなさい

**[11 章]**

<b>283</b> ページ 練習 11-1	エラーメッセージを判読する練習であることを明示するため問題文に説明を補足 …、 <b>エラーメッセージを見て</b> 、例外の発生した行番号と表示される例外のクラス名を答えなさい。
<b>283</b> ページ 練習 11-2	別ファイルであることが分かるように囲みを 2 つに分ける  <div style="display: flex; align-items: center; justify-content: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 20px;"> <p>誤</p> <pre>class Blue {     String colorName; } public static void doIt(){     Blue b = null;     System.out.println(b.colorName); }</pre> </div> <div style="font-size: 2em; margin-right: 20px;">→</div> <div style="border: 1px solid black; padding: 5px;"> <p>正</p> <pre>class Blue {     String colorName; } public static void doIt(){     Blue b = null;     System.out.println(b.colorName); }</pre> </div> </div>
<b>303</b> ページ 練習 2-1	下段の実行例、3 行目 誤… java.lang.ArithmeticException: 引数がゼロ ← ② 正… java.lang. <b>Exception</b> : 引数がゼロ ← ②
<b>303</b> ページ 練習 2-1	下段の実行例、4 行目 誤… java.lang.ArithmeticException: 引数がゼロ ← ③ 正… java.lang. <b>Exception</b> : 引数がゼロ ← ③

<p><b>305</b> ページ 練習 11-5 の 1</p>	<p>誤… 引数 <code>s</code> を表示するだけのクラスです。 正… 引数 <code>s</code> を表示するだけのメソッドです。</p>
<p><b>316</b> ページ 通過テスト 1</p>	<p><code>return</code> を追加します</p> <p>誤…</p> <pre>int doIt(String s){     try{         s.equals("abc")? 0:1;     }     . . . . . }</pre> <p>正…</p> <pre>int doIt(String s){     try{         return s.equals("abc")? 0:1;     }     . . . . . }</pre>
<p><b>[12 章]</b></p>	
<p><b>344</b> ページ 本文</p>	<p>中段の文で「点線枠」を「青枠」訂正</p> <p>誤… 次に, <code>ExecWriteObject</code> クラスの点線枠の部分がオブジェクトの出力処理です 正…次に, <code>ExecWriteObject</code> クラスの青枠の部分がオブジェクトの出力処理です</p>
<p><b>353</b> ページ 例題 9</p>	<p>下図のように <code>in.close()</code>; を挿入してください。</p> <pre>         System.out.println("number=" + f.number);         System.out.println("tom=" + f.tom);         System.out.println("name=" + f.name);         in.close();     } catch (Exception e) {} } </pre>  <p>実行結果▶</p>
<p><b>353</b> ページ 練習 12-9</p>	<p>問題 1 の問題文にヒントを追加</p> <p>誤… 1. 次のコードを実行したとき正しいものはどれか答えなさい。 正… 1. 次のコードを実行したとき正しいものはどれか答えなさい。</p> <p>&lt;ヒント&gt;</p> <ul style="list-style-type: none"> <li>・クラス変数は直列化の対象にはなりません</li> <li>・p.261 の「<code>static</code> とは」を読んでクラス変数の知識を確認してください</li> </ul>
<p><b>358</b> ページ まとめ</p>	<p>「ファイルパスと <code>File</code> クラス」</p> <p>誤… <code>File</code> パス文字列を引数にして <code>File</code> オブジェクトを作成する。 正… <u>ファイル</u>パス文字列を引数にして <code>File</code> オブジェクトを作成する。</p>

<p><b>359</b> ページ まとめ</p>	<p>直列化 (シリアライズ) と直列化復元 (デシリアライズ)</p> <p>最下段の <code>ObjectInputStream</code> の例文</p> <p>誤…</p> <pre>ObjectInputStream out = new ObjectInputStream( new FileInputStream("my.ser")); obj = out.readObject(); // 直列化復元</pre> <p>正…</p> <pre>Ball obj; ObjectInputStream in = new ObjectInputStream( new FileInputStream("my.ser")); obj = (Ball)in.readObject(); // 直列化復元</pre>
-------------------------------	---

**[13 章]**

<p><b>372</b> ページ 本文の表</p>	<p>メソッド欄 2 行目の引数名を訂正します</p> <p>誤… <code>String substring(int to)</code></p> <p>正… <code>String substring(int from)</code></p> <hr/> <p>解説欄 3 行目 小文字の <code>s</code> に訂正</p> <p>誤… <code>S</code> を連結した文字列を返す</p> <p>正… <code>s</code> を連結した文字列を返す</p> <hr/> <p>メソッド欄の最後の行 <code>()</code> を付ける</p> <p>誤… <code>int length_</code></p> <p>正… <code>int length()</code></p>	<p><b>396</b> ページの「まとめ」にも同じ表が再掲されています。同様に訂正してください。</p>
--------------------------------	---	--

<p><b>378</b> ページ 本文の表</p>	<p>表の項目名を訂正してください</p> <p>&lt;上段の表&gt;</p> <p>誤… <code>String</code> クラスのメソッド</p> <p>正… <code>StringBuilder</code> クラスのコンストラクタ</p> <p>&lt;下段の表&gt;</p> <p>誤… <code>String</code> クラスのメソッド</p> <p>正… <code>StringBuilder</code> クラスのメソッド</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p>▼主なコンストラクタ</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #cccccc;">StringBuilderクラスのコンストラクタ</th> <th style="background-color: #cccccc;">説明</th> </tr> </thead> <tbody> <tr> <td><code>public StringBuilder()</code></td> <td>初期容量(16文字)で作成する</td> </tr> <tr> <td><code>public StringBuilder(int m)</code></td> <td>容量mバイトで作成する</td> </tr> <tr> <td><code>public StringBuilder(String s)</code></td> <td>文字列sを格納した状態で作成</td> </tr> </tbody> </table> <p>▼重要なメソッド</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #cccccc;">StringBuilderクラスのメソッド</th> <th style="background-color: #cccccc;">説明</th> </tr> </thead> <tbody> <tr> <td><code>SB append(X)</code></td> <td>Xを末尾に追加する</td> </tr> <tr> <td><code>SB insert(int p, X)</code></td> <td>Xを先頭からp番目の位置に挿入</td> </tr> </tbody> </table> </div>	StringBuilderクラスのコンストラクタ	説明	<code>public StringBuilder()</code>	初期容量(16文字)で作成する	<code>public StringBuilder(int m)</code>	容量mバイトで作成する	<code>public StringBuilder(String s)</code>	文字列sを格納した状態で作成	StringBuilderクラスのメソッド	説明	<code>SB append(X)</code>	Xを末尾に追加する	<code>SB insert(int p, X)</code>	Xを先頭からp番目の位置に挿入	<p><b>397</b> ページの「まとめ」にも同じ表が再掲されています。同様に訂正してください。</p>
StringBuilderクラスのコンストラクタ	説明															
<code>public StringBuilder()</code>	初期容量(16文字)で作成する															
<code>public StringBuilder(int m)</code>	容量mバイトで作成する															
<code>public StringBuilder(String s)</code>	文字列sを格納した状態で作成															
StringBuilderクラスのメソッド	説明															
<code>SB append(X)</code>	Xを末尾に追加する															
<code>SB insert(int p, X)</code>	Xを先頭からp番目の位置に挿入															

<p><b>384</b> ページ 表</p>	<p>表の小数桁数にドットがない</p> <p>誤… 小数桁数</p> <p>正… . 小数桁数</p>
-----------------------------	--

<p><b>397</b> ページ</p>	<p>上段の <code>StringBuilder</code>, <code>StringBuffer</code> クラスの表の最下行に 1 行追加する</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><code>void setCharAt(int p, char c)</code></td> <td style="padding: 2px;">p 番目の位置に c をセットする</td> </tr> </table>	<code>void setCharAt(int p, char c)</code>	p 番目の位置に c をセットする
<code>void setCharAt(int p, char c)</code>	p 番目の位置に c をセットする		

表	正…
388 ページ 本文	中段の Y および YYYY を小文字に訂正 誤… 書式化文字列では、年 (Y または YYYY) 、月 正… 書式化文字列では、年 (y または yyyy) 、月
399 ページ 通過テスト	ヒントの 3 行目 誤… 取り出した文字からマイナス記号の位置を求めてその前後を取り出します 正… <u>取り出した文字列の中のマイナス記号 (-) を空白に置き換えます</u>

## [14 章]

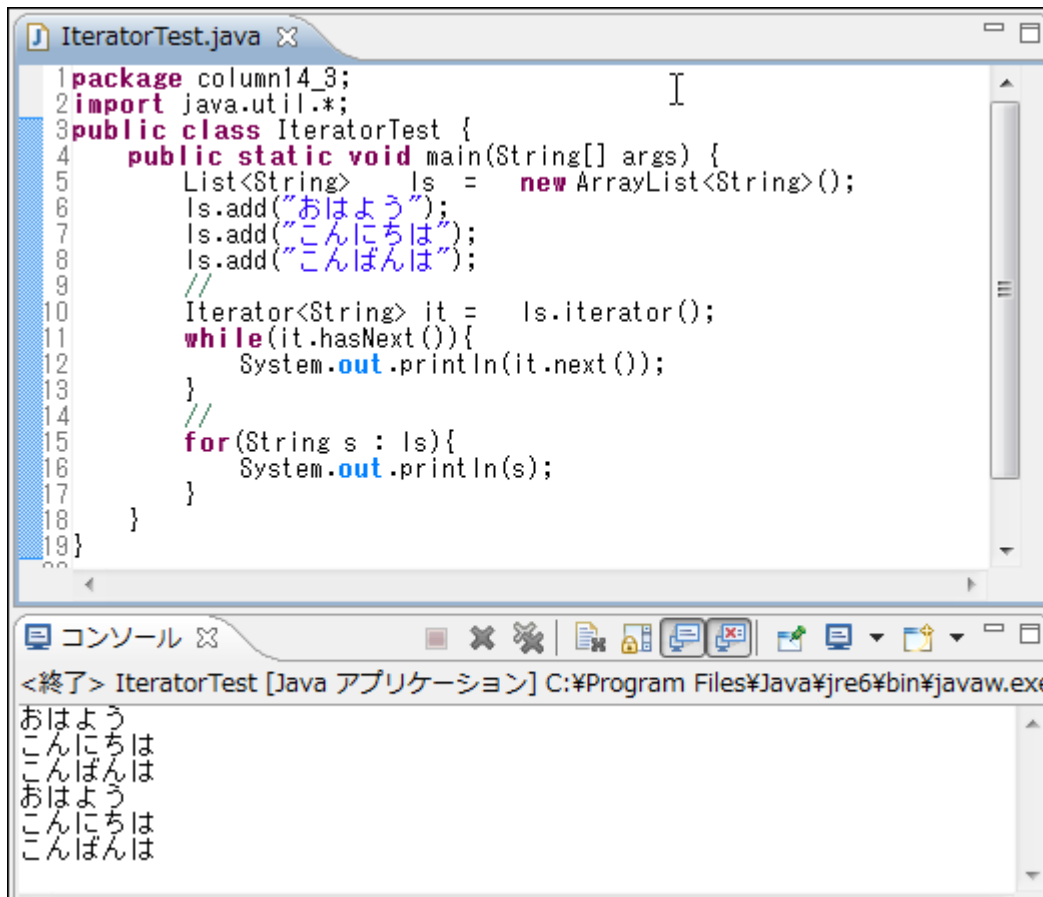
403 ページ 説明図	キュー構造の図に peek メソッドの説明を追加してください  <div style="border: 1px solid black; padding: 10px;"> <p>キュー構造 (FIFO: First In, First Out)</p>  <p>最初に入れたものを最初に出すという構造</p> <p>offer -- 入れる poll --- 取り出す (キューから削除する) peek --- 取り出す (キューから削除しない)</p> </div>
406 ページ 練習 14-1 の 1	問題文の 1~2 行目 誤… 該当するインタフェースが複数ある場合は・・・ 正… 該当するインタフェースやクラスが複数ある場合は・・・
408 ページ 本文 中ほどの位置	誤… <u>ListArray</u> に Dog 型のオブジェクトを格納する・・・ 正… <u>ArrayList</u> に Dog 型のオブジェクトを格納する・・・
413 ページ コラム	ページ下部の注 誤… ※E, T は総称型を表します。また、反復子についてはコラムを参照してください。 正… ※E, T は総称型を表します。また、反復子については P.436 のコラムを参照してください。
415 ページ プログラムのコメント	誤… // Collection クラスのメソッドです 正… // Collections クラスのメソッドです
422 ページ 表	表の 1 行目    o (小文字) を O (大文字に) する 誤… Integer obj = Integer.valueof(10); 正… Integer obj = Integer.valueOf(10);

<p><b>425</b> ページ</p>	<p>上段のプログラムのコメントを移動する</p> <pre> s.add("お"); s.add("え"); s.add("う"); s.add("い"); s.add("あ"); <del>// 重複要素なので登録できないはず</del> for(String str : s ){     System.out.print(str+"¥t");// どんな順で表示されるか } System.out.println(""); // 改行 // // 重複要素を登録して結果を表示する System.out.print(s.add("あ")); // 重複要素なので登録できないはず } </pre> <p>移動する</p>
<p><b>427</b> ページ 本文</p>	<p>説明をわかりやすくするため6行目に挿入</p> <p>マップは要素とそのキーをセットで使用するデータ構造です。キーと要素は一対一に対応します。</p>
<p><b>434</b> ページ 通過テスト 3のSPD</p>	<p>SPDの最後にある配列変数名を小文字にする</p> <p>誤… Cars[i] 正… cars[i]</p>
<p><b>435</b> ページ 通過テスト 6</p>	<p>問題文を補足</p> <p>6. p.404の継承図を参照し、「ここに挿入」の部分に書くことのできるものはどれか答えなさい。 ただし、peekはキュー構造を表すリストから先頭の要素を取り出すメソッドです。</p>

- 1) コラム「GUI の画面はどうやって作るのか (その 1)」を P.522 へ移動し、(その 2) と同じページにおく
- 2) 以下のコラムを P. 436 に掲載する

## Iterator と拡張 for 文

**Iterator** を反復子といいます。List や Set など、要素を先頭から順にひとつずつ取りだしたいときに利用します。下の例は、Iterator を使ってリストから順に全ての要素を取り出して表示しています。



```
1 package column14_8;
2 import java.util.*;
3 public class IteratorTest {
4     public static void main(String[] args) {
5         List<String> ls = new ArrayList<String>();
6         ls.add("おはよう");
7         ls.add("こんにちは");
8         ls.add("こんばんは");
9         //
10        Iterator<String> it = ls.iterator();
11        while(it.hasNext()){
12            System.out.println(it.next());
13        }
14        //
15        for(String s : ls){
16            System.out.println(s);
17        }
18    }
19 }
```

コンソール

```
<終了> IteratorTest [Java アプリケーション] C:\Program Files\Java\jre6\bin\javaw.exe
おはよう
こんにちは
こんばんは
おはよう
こんにちは
こんばんは
```

`hasNext()` は次に取りだす要素がまだあるかどうか調べます。ある時は `true`、ない時は `false` を返します。11 行の `while` 文では `it.hasNext()` が `true` の間繰り返すことで、全ての要素を取り出します。これは `hasNext()` の典型的な使い方です。

実際に要素を取り出すのは `next()` です。12 行目で `it.next()` で要素を取り出して表示しています。

ところで、15 行目からの拡張 `for` 文は `iterator` を使う `while` 文と同じ働きをします。しかも、どうみてもこちらの方が `iterator` より簡明です。実は、拡張 `for` 文は `Iterator` に代わる仕組みとして Java1.5 から導入されたものなのです。そのため、今では `Iterator` の出番は極端に少なくなっています。

[15 章]

455 ページ 本文	本文上から 3 行目 誤… 次 <u>ぎ</u> の表を見てください。 正… 次の表を見てください。
456 ページ 練習 15-4	本文上から 5 行目 誤… ・この節の解説で説明した setName メソッドを使ってスレッドに名前を付けなさい 正… ・この節の解説で説明した setName メソッドを使ってスレッドに名前としてそれぞれ A, B, C を付けなさい
459 ページ 練習 15-6 の 2	SPD の 2 行目 誤… スレッド名と i の値を表示する 正… スレッド名と i+1 の値を表示する
464 ページ 本文	問題文 問 3 の 3 行目 誤… ……と優先順位を取得して表示しなさい。 正… ……と優先順位を取得して下図の実行結果のように表示しなさい。
475 ページ 例示プログラム	欄外の補足の記述を削除する 誤… ※10B を実行するには、10A と同じパッケージに入れてください 正… <del>削除</del>
476 ページ 例示プログラム	プログラムの 3 行目はスーパークラスと同じ変数を定義している。このような同名の変数を定義した場合の影響については p238 「隠蔽と再利用」で説明している。ここでは不要なので削除する 誤… int dish = 0; // 何番目の料理かを示す値 正… <del>削除</del>
479 ページ まとめ	「複数のスレッドを起動する」の 2 行目 誤… Thread.current() は現在実行中のスレッドオブジェクトの参照を返す 正… Thread.currentThread() は現在実行中のスレッドオブジェクトの参照を返す

[16 章]

484 ページ インタフェース宣言	誤… int SMALL        80; int MEDIUM    80; int LARGE     80;  誤… int SMALL        = 80; int MEDIUM    = 80; int LARGE     = 80;
486 ページ 例題	例題 1 中ほどのコメント 誤… …… // 列強定数初期化用の… 正… …… // 列挙定数初期化用の…
486 ページ 本文	【解説】の 1 行目 誤… …… 破線で囲った部分が… 正… …… 青いアミカケの部分が…

<p><b>488</b> ページ 本文</p>	<p>本文上から 1 行目 誤… ……一般のクラスと同様に <code>class</code> ファイルが作成される… 正… ……一般のクラスと同様に <code>class</code> ファイルが作成される…</p>
<p><b>491</b> ページ 例題 4</p>	<p>誤… <code>Result.SUCCESS, Result.SUCCESS, }</code> 正… <code>Result.SUCCESS, Result.SUCCESS }</code></p>
<p><b>492</b> ページ 例題 4</p>	<p>492 ページ 1 行目の <code>break;</code> は不要なので削除 誤… <code>break;</code> 正… <b>-削除-</b></p>
<p><b>492</b> ページ 練習 16-2 の問題文</p>	<p>誤… このとき、<code>Card</code> 型の配列 <code>cards</code> を作成し、次の 8 枚のカードを要素として持つようにしなさい。 正… このとき、<b>main メソッドを持つ <code>Exec</code> クラスを作成しなさい。</b><b><code>Exec</code> クラスの main メソッドでは、</b><code>Card</code> 型の配列 <code>cards</code> を作成し、次の 8 枚のカードを要素として持つようにしなさい。</p>
<p><b>504</b> ページ 例題 7 の実行結果</p>	<p>実行結果 2 行目 誤… 静的初期化ブロック <u>1</u> 正… 静的初期化ブロック <u>2</u></p>

**[17 章]**

<p><b>531</b> ページ 例題 2</p>	<p>プログラム 下から 3 行目 誤… インデントの誤り <code>}</code> が行の左端にある 正… 次のように直す</p> <pre> public class Exec2 {     public static void main(String[] args) {         String reg = Input.getString("正規表現");         String str = Input.getString("文字列");         Pattern p = Pattern.compile(reg);         Matcher m = p.matcher(str);         while (m.find()) {             System.out.println(m.start() + "-" + m.end() + " ");         }     } } </pre>
<p><b>532</b> ページ 本文 例示プログラム</p>	<p>3 つ目の網掛けの例示プログラム バックスラッシュを <code>¥</code> に訂正 誤… <code>System.out.println(m.start()+"-"+m.end()+"\t" + m.group());</code> 正… <code>System.out.println(m.start()+"-"+m.end()+"¥t" + m.group());</code></p>
<p><b>536</b> ページ 練習 17-3</p>	<p>網掛け部分 <code>t</code> が抜けている 誤… <code>public String replaceFirs(String regx, String rep)</code> 正… <code>public String replaceFirst(String regx, String rep)</code></p>
<p><b>544</b> ページ 通過テスト SPD</p>	<p>SPD の 2 行目 「ひとつ以上の」を挿入 誤… <code>useDelimiter</code> メソッドで区切り文字をひとつ以上の空白かコンマに設定する 正… <code>useDelimiter</code> メソッドで区切り文字をひとつ以上の空白か<b>ひとつ以上の</b>コンマに設定する</p>

**[プログラミング演習]**



<p><b>554</b> ページ</p> <p>本文</p>	<p>最後の行</p> <p>誤… <code>maim</code></p> <p>正… <code>main</code></p>
<p><b>576</b> ページ</p> <p>本文</p>	<p>中段の行 <code>()</code>が抜けている</p> <p>誤… 残りカード枚数は <code>deck.size</code> で求められます</p> <p>正… 残りカード枚数は <code>deck.size()</code> で求められます</p>
<p><b>580</b> ページ</p> <p>本文</p>	<p>中段の行 <code>return</code> を <code>Enter</code> に訂正</p> <p>誤… 手札を引く場合は <u>Return</u> をタイプします。<u>Return</u> をタイプすると・・・</p> <p>正… 手札を引く場合は <u>Enter</u> をタイプします。<u>Enter</u> をタイプすると・・・</p>

**[補足資料]**

<p><b>586</b> ページ</p> <p>表内の 4,5,6,7 行目</p>	<p>誤… <code>publid</code></p> <p>正… <code>public</code></p>
---	---

[練習問題の解答の訂正]

<p><b>595</b> ページ 練習 2-1</p>	<p>解答プログラムを以下のように訂正</p> <pre>package ex1; public class Player {     String handle; // 呼び名     int[] hand; // 手札配列      int <del>tips</del> chips; // チップの残り数     public Player(String <del>tips</del> _handle, int[] <del>tips</del> _hand, int <del>tips</del> <u>_chips</u>){         handle = _handle;         hand = _hand;          <del>tips</del> chips = <u>_chips</u>;     } }</pre>
<p><b>595</b> ページ 練習 2-2</p>	<p>解答プログラムを以下のように訂正</p> <pre>public class ExecPlayer {     public static void main(String[] args) {         Player dealer             = new Player("comp", new int[]{5, 16, 22, 31, 33}, 150);         Player player1             = new Player("Tom", new int[]{2, 13, 26, 13, 48}, 50);          //dealer         System.out.print(dealer.handle + ", {");         for(int m : dealer.hand){             System.out.print(m+ " ");         }         System.out.println("}, " + dealer.<del>tips</del> <u>chips</u>);         // Tom         System.out.print(player1.handle + ", {");         for(int m : player1.hand){             System.out.print(m+ " ");         }         System.out.println("}, " + player1.<del>tips</del> <u>chips</u>);     } }</pre>
<p><b>598</b> ページ 練習 2-4 の 2</p>	<p>【解説】 誤… クラス変数は参照変数なので、参照以外を代入することはできません。… 正… <b>クラス型</b>変数は参照変数なので、参照以外を代入することはできません。…</p>
<p><b>600</b> ページ <b>3章</b> 練習 3-4</p>	<p>setSeconds の末尾の s が抜けている</p> <p>誤… 3) void setSecond(int s) ⇒ public void setSecond(int s) 正… 3) void setSeconds(int s) ⇒ public void setSeconds(int s)</p>
<p><b>603</b> ページ <b>5章</b> 練習 5-2</p>	<p>2. の【解説】の 2行目</p> <p>誤… C, D は正しいコンストラクタです。 正… <b>C</b> はコンストラクタと同名のメソッドですが、文法的には許されます。また、<b>D</b> は正しいコンストラクタです。</p>

<p><b>609</b> ページ <b>7章</b> 練習 7-5</p>	<p>1. の問題で⑤を削除したので、解答でも⑤を削除する (7章の正誤表を参照) 誤… ①E ②G ③A ④D <u>⑤F</u> ⑥B ⑦H ⑧C ⑨C 正… ①E ②G ③A ④D ⑥B ⑦H ⑧C ⑨C</p>
<p><b>609</b> ページ <b>8章</b> 練習 8-4</p>	<p>1. の【解説】の最後の行 誤… <u>web</u> なら自動型変換されます 正… <u>Myweb</u> なら自動型変換されます</p>
<p><b>611</b> ページ 練習 9-2</p>	<p>解説 1行目右端 誤… デフォルトアクセスを private 正… デフォルトアクセスを <u>protected</u></p>
<p><b>617</b> ページ <b>11章</b> 練習 11-3</p>	<p>1. の解答 誤… A、C、G、H、<u>I</u>、K、<u>L</u>、<u>N</u> 正… <u>A</u>、<u>C</u>、<u>G</u>、<u>H</u>、<u>J</u>、<u>K</u>、<u>M</u></p>
<p><b>620</b> ページ <b>12章</b> 練習 12-5</p>	<p>プログラム1行目の import 文 誤… import java.io.<u>FileNotFoundException</u>; 正… import java.io.<u>IOException</u>;</p>
<p><b>623</b> ページ <b>12章</b> 練習 12-9</p>	<p>1. の【解説】最後から2行目 誤… 最後に a は直列化され、・・・ 正… 最後に <u>X</u> は直列化され・・・</p> <p>「2. D」とあるのは 12.10 の 1 の解答です。見出しの練習番号を挿入し、問題番号を訂正します。</p> <p>誤… 2. D 正… <u>《12-10》</u> 1. D</p>
<p><b>624</b> ページ <b>13章</b> 練習 13-3</p>	<p>解答プログラムに文字数表示が抜けていた。文字数の表示(下波線の行)を追加する。</p> <pre>public class Exec {     public static void main(String[] args) {         String s = " c:/data/java/samples ";         s = s.replace("/", "¥¥").trim().toUpperCase();         System.out.println(s);         <u>System.out.println("文字数="+ s.length);</u>     } }</pre>

<p>625～626 ページ 13章 練習 13-7</p>	<p>1.の解答。625 ページの下から 3 行目 誤… System.out.printf(“%4d %4d¥n”, 150, 551); 正… System.out.printf(“%8d %8d¥n”, 150, 551);</p> <p>2. の解答の訂正と【解説】の追加 誤… “%3d 円のパンが%1d つで%4d 円です¥n” 正… “%3d 円のパンが%1d つで%3d 円です¥n” 【解説】 “%d 円のパンが%d つで%d 円です¥n” のように桁幅指定を省略してもよい</p>
<p>629 ページ 14章 練習 14-4</p>	<p>1.の【解説】の末尾に追加 クラスメソッドを使うなら、Double.valueOf(123.4)、Integer.valueOf(1234)、としてもよい</p>
<p>631 ページ</p>	<p>問 1 と問 2 の配置が逆になっているので入れ替える</p> <pre> public class Exec { // 問 2     public static void main(String[] args){         Counter c = new Counter();         Thread t = new Thread(c);         t.start();         System.out.println("メインスレッド終了");     } }  class Counter implements Runnable{ // 問 1     public void run(){         for(int i=0; i&lt;5; i++){             System.out.println(i+1 + "回目です");         }     } } </pre> <p style="text-align: right;">入れ替える</p>
<p>631 ページ 15章 練習 15-1</p>	<p>2.の【解説】の記述で MyRunnable が Mythread となっていた 誤… MyThread m = new MyThread(); 正… MyRunnable m = new MyRunnable();</p>
<p>632 ページ 15章 練習 15-2</p>	<p>2. の【解説】を訂正 誤… t.start()で起動しているのは Thread クラスの start メソッド。MyThread クラスの start メソッドを実行しているのではない。 正… t.start()で起動しているのは t の start メソッド。m の start メソッドを実行しているのではない。</p>
<p>632 ページ 15章 練習 15-3</p>	<p>【解説】 誤… MyThread クラスは… 正… 無名クラスは…</p>