

『Java によるアルゴリズムとデータ構造の基礎』（初版 1 刷）正誤表 2020 年 9 月 3 日現在

No.	頁 行	誤	正
1	p.12 図 1.3	StringValueOf(i) StringValueOf(f)	String.valueOf(i) String.valueOf(f)
2	p.14 表 1.3 11 行目	System.out.printf("%02d", i2);	System.out.printf("%05d", i2);
	15 行目	System.out.printf("-8.2%f", d2);	System.out.printf("%-8.2%f", d1);
	20 行目	System.out.printf("-8.2f%", d2);	System.out.printf("%-8.2f", d2);
3	P.14 プログラム 1.7	2 行目 import java.util.*;	削除
4	p.20 下から 2 行目	text.bat	test.bat
5	p.23 (9)①	した処理	したい処理
6	p.24 プログラム 1.18	(PreodicApp.java)	(PeriodicApp.java)
7	p.26 下から 7 行目	基本処理技術者試験	基本情報処理技術者試験
8	p.28 表 2.2 左	=, <>	==, !=
9	p.29 表 2.3	AND, OR, NOT	&&, , !
	図 2.3	b: 条件式 c: 変化式	b: 変化式 c: 条件式
10	p.32 上から 12 行目	while(true);	} while(true);
11	p.43 プログラム 3.3 10 行目	add(3,28) :	add(3,29) :
	実行結果	After add(3,28) :	After add(3,29) :
12	p.44 プログラム 3.4 9 行目	System.out.print(" { "s[i].getNo()+	System.out.print(" { "+s[i].getNo()+
13	p.45 上から 6 行目	Student クラスには・・・	この文削除
14	P.58 上から 10 行目	n > m > 0	n > m >= 0
	上から 11 行目	gcd(n,m) = m	gcd(n,m) = n
15	p.59 プログラム 4.2 11 行目	division of "+n+" =	divisor of "+n+" and "+m+" = =

	下から 8 行目	中間軸 (inner)	中間軸 (inter)
	下から 4 行目	"Disk(n-1)from(o)to(i)"	"Disk(n)from(o)to(d)"
16	p.61 プログラム 4.3 3 行目	char orign	char origin
	4 行目	orign "+from+	from "+origin+
	6~8 行目	orign	origin
17	p.71 プログラム 5.2 23 行目	Node tmp = head;	Node<E> tmp = head;
18	p.76 プログラム 5.5 64 行目	public void remove(Node p){	public void remove(Node<E> p){
19	P.78 図 5.8	追記	<p style="text-align: right;">[0],[1],[2]: weight[]のインデックス</p> <p style="text-align: right;">Bias bias[0] I-0 input[0] I-1 input[1] H-0 hidden[0] H-1 hidden[1] O-0 output[0]</p>
20	p.79 プログラム 5.7 5 行目	ArrayList link;	ArrayList<Neuron> link;
	15 行目	= new ArrayList();	= new ArrayList<Neuron> ();
21	p.80 プログラム 5.8 14 行目	Xor()	XorApp()
	40 行目	" %6.3f %6.3f %6.3f %6.3f %6.3f %6.3f ¥n",	" %6.3f %6.3f %6.3f ¥n",
	44 行目	削除	空白行のまま
	52 行目	" %6.3f %6.3f %6.3f %6.3f %6.3f %6.3f ¥n",	" %6.3f %6.3f %6.3f ¥n",
	55 行目	削除	空白行のまま
	168 行目	Xor()	XorApp()

22	p.83 実行結果	本に記載の部分削除	<pre> 0 : err= 1.0250 w: 0.411 0.076 0.098 0.239 -0.061 -0.242 -0.204 0.471 -0.484 100 : err= 0.4955 w: -4.221 6.964 0.075 -5.030 0.700 0.977 -7.907 5.215 -3.364 200 : err= 0.0039 w: -7.972 8.361 3.720 -5.277 5.265 2.570 -8.115 6.110 -3.333 300 : err= 0.0020 w: -8.656 8.910 4.091 -5.559 5.553 2.747 -8.140 6.349 -3.352 400 : err= 0.0013 w: -9.053 9.257 4.303 -5.719 5.713 2.844 -8.159 6.481 -3.377 492 : err= 0.0010 w: -9.311 9.492 4.439 -5.821 5.815 2.905 -8.172 6.565 -3.398 </pre>
23	p.88 プログラム 6.1 22 行目	if(top <= 0)	if(top < 0)
24	p.104 上から 2 行目	<p>プログラム 7.1 に 2 分探索木のプログラムを、プログラム 7.2 にテストプログラムと実行結果を示す。ここでは、Student クラス・・・</p> <p>追記する。</p> <p>行数が追加となるため、図 7.8 の最後の図を省略し、スペースをかせぐ。</p>	<p>プログラム 7.1 に 2 分探索木のプログラムを、プログラム 7.2 にテストプログラムと実行結果を示す。メンバ変数 comparator は、大小比較をするためのコンパレータである。ここで、「Comparator<? super K> comparator = null;」は、K もしくはその先祖クラスに対する comparator を null とする意味である。また、コンストラクタ BinTree() は、引数のないものと、Comparator のオブジェクトを引数とする 2 種類を定義している。キー key1 と key2 の大小比較は、コンパレータの指定の有無により以下のメソッドを利用している。</p> <ul style="list-style-type: none"> • Comparator の指定が無の場合 : <pre>key1.compareTo(key2)</pre> <ul style="list-style-type: none"> • Comparator の指定が有の場合 :

			<code>comparator.compare(key1,key2)</code> 戻り値は、 <code>key1<key2</code> : 負の整数, <code>key1=key2</code> : 0, <code>key1>key2</code> : 正の整数である。 ここでは, Student クラス・・・
25	p.110 プログラム 7.4 8 行目	<code>hs.print_data(a);w</code>	<code>hs.print_data(a);</code>
26	p.111 文章 2 行目	2,1 と変更しながら	2,1,0 と変更しながら
27	P.112 図 7.8 最下位図	矢印	矢印のそばに交換を追記
28	P.113 図 7.8 最上位図	矢印	矢印のそばに交換を追記
	heapify(a,5,0) 図	図	削除 (スペースをかせぐため)
	heapify(a,5,0) :	heapify(a,5,0) :	heapify(a,5,0) : 図省略 {31, 29, 3, 28, 0, 32, 61, 64, 97} 網掛け : ソート済
29	p.129 下から 23 行目	追記	プログラム 8.7 にそのテスト用クラスと実行結果を示す。プログラム 8.7 の 19 行目のコメントを外すと, "DELETED" の状態を確認することができる。
30	p.139 図 9.2	<code>a[i-1] < a[i]</code> , <code>a[i-1]</code> と <code>a[i]</code> の交換	<code>a[j-1] > a[j]</code> , <code>a[j-1]</code> と <code>a[j]</code> の交換
	プログラム 9.1 6 行目	<code>i<n-2</code>	<code>i<n-1</code>

31	p.141 図 9.5	本文と不一致	<p>The diagram illustrates the steps of a sorting algorithm on an array [64, 28, 61, 32, 29]. It shows comparisons and swaps between elements at indices i and \min. Red boxes highlight the elements being compared or swapped. Red arrows indicate the movement of elements.</p>
32	p.142 プログラム 9.2 6 行目	$i < n - 2$	$i < n - 1$
33	p.143 図 9.6	$a[i] = tmp$	$a[j] = tmp$
	図 9.7	$i = 3, 4$ の図	途中の過程の図を削除
34	p.145 上から 6 行目	追記	表 9.2 に示す。プログラム 9.4 に <code>HashMap</code> を用いたソートテストクラスの例を示す。このプログラムは、 <code>Set</code> と <code>Iterator</code> インタフェースを用いている。 <code>Set</code> は重複のない要素の集合であるコレクションである。 <code>Java</code> におけるコレクションは要素の集合の総称であり、 <code>Set</code> の他に <code>List</code> や <code>Deque</code> なども含まれ、それぞれ有用なメソッドが準備されている。また、 <code>Iterator</code> はコレクションの要素を順番に処理する場合に利用され、こちらにも有用なメソッドが準備されている。
35	p.145 表 9.2	<code>map.remove()</code>	<code>map.remove(key)</code>
36	p.146 プログラム 9.4 14 行目 15 行目	<code>System.out.println(me.getKey() + ": ");</code> <code>System.out.println(me.getValue());</code>	<code>System.out.println ("(key,value)= ("+</code> <code>me.getKey()+", "+me.getValue()+ ")");</code>

37	p.152 図 10.4	<pre> pivot = a[(1 + r)/2] r++ </pre>	<pre> pivot = a[(0 + r)/2] r-- </pre> <p>判断 (0 <= r) を追加して、図を修正</p>
38	p.153 下から 7 行目	<pre> pivot = a[(1 + r)/2] </pre>	<pre> pivot = a[(0 + r)/2] </pre>
39	p.156 図 10.6	<pre> mrge(a, left, m, right) </pre>	<pre> marge(a, left, m, right) </pre> <p>図を修正</p>
40	p.159 プログラム 10.3 16 行目 17 行目	<pre> ++i ++j </pre>	<pre> i++ j++ </pre>
41	p.160 表 10.1	<pre> static<T> sort(T[] a, comparator<? Super T> c) </pre>	<pre> static<T> sort(T[] a, Comparator<? super T> c) </pre>
42	p.161 下から 9 行目	<pre> comparator<? Super T> c </pre>	<pre> Comparator<? super T> c </pre>
43	p.164 下から 14 行目 12 行目	<pre> Comparator<Student>... comparators </pre> <p>ピリオドを 3 つ (…)</p>	<pre> Comparator<Student>... comparators </pre> <p>ピリオドを 3 つ (...)</p>
44	p.165 プログラム 10.11 19 行目	<pre> for(Student s : listStudent) </pre>	<pre> for(Student s : ls) </pre>
45	p.174 プログラム 11.3 12 行目 13 行目	<pre> getBranchs() setBranchs(</pre>	<pre> getBranches() setBranches(</pre>
	プログラム 11.4 15 行目 16 行目	<pre> getBranchs().add(branches[i]); getBranchs().add(branches[i]); </pre>	<pre> getBranches().add(branches[i]); getBranches().add(branches[i]); </pre>
46	p.175 プログラム 11.4 19 行目	<pre> getBranchs() </pre>	<pre> getBranches() </pre>
47	p.181 実行結果 1 行目		<pre> #### Start = A Goal = E </pre>
48	p.184 下から 7 行目, 6 行目	<pre> :\$HOME/jara/bbb.jar </pre>	<pre> :\$HOME/jar/bbb.jar </pre>