

Chapter 6: Transition to Java

Programming with Alice and Java
First Edition

by
John Lewis
and
Peter DePasquale



Copyright © 2009 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

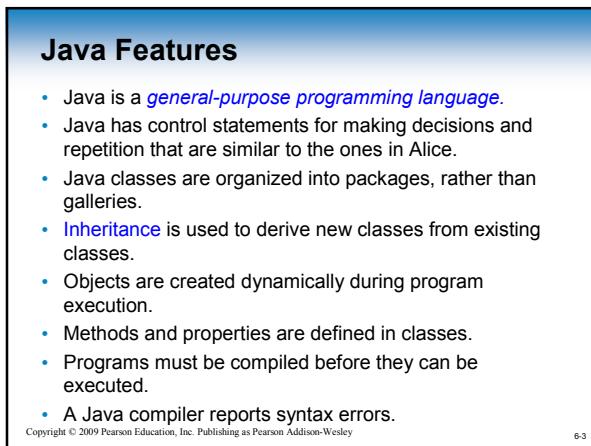
6-1

Objectives

- Compare the concepts you saw in Alice to their counterparts in Java
- Learn about program development environments for Java
- Access the Java API support library and its online documentation
- Explore several Java program examples
- Practice using various Java statements

Copyright © 2009 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

6-2

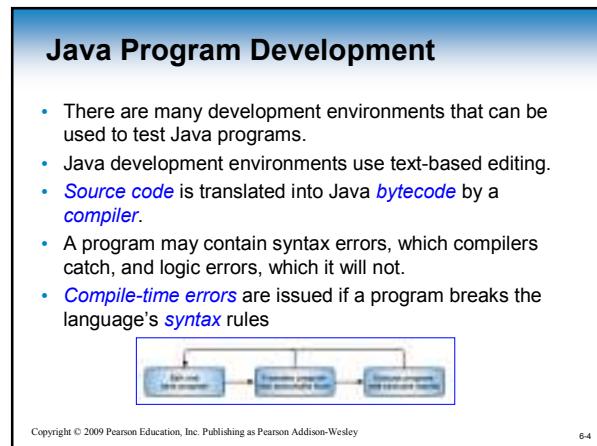


Java Features

- Java is a *general-purpose programming language*.
- Java has control statements for making decisions and repetition that are similar to the ones in Alice.
- Java classes are organized into packages, rather than galleries.
- *Inheritance* is used to derive new classes from existing classes.
- Objects are created dynamically during program execution.
- Methods and properties are defined in classes.
- Programs must be compiled before they can be executed.
- A Java compiler reports *syntax errors*.

Copyright © 2009 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

6-3



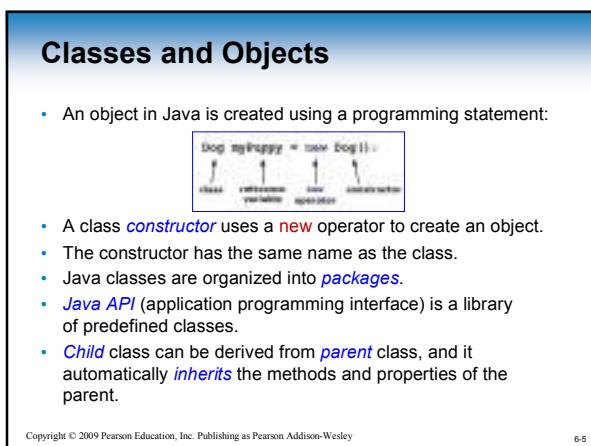
Java Program Development

- There are many development environments that can be used to test Java programs.
- Java development environments use *text-based editing*.
- *Source code* is translated into Java *bytecode* by a *compiler*.
- A program may contain *syntax errors*, which compilers catch, and *logic errors*, which it will not.
- *Compile-time errors* are issued if a program breaks the language's *syntax rules*



Copyright © 2009 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

6-4



- An object in Java is created using a programming statement:

```

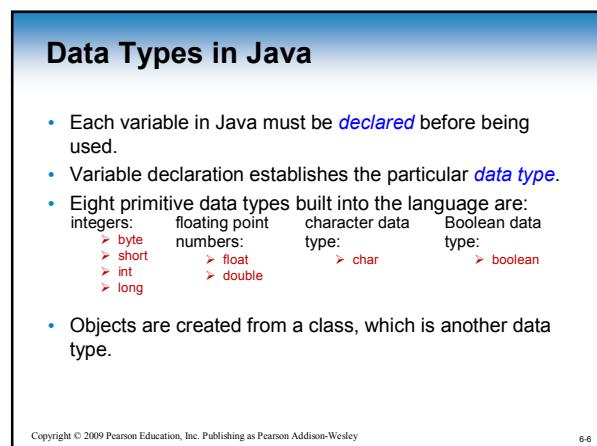
class Dog {
    ...
}

Dog myDoggy = new Dog();
    
```

- A class *constructor* uses a *new* operator to create an object.
- The constructor has the same name as the class.
- Java classes are organized into *packages*.
- *Java API* (application programming interface) is a library of predefined classes.
- *Child* class can be derived from *parent* class, and it automatically *inherits* the methods and properties of the parent.

Copyright © 2009 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

6-5



- Each variable in Java must be *declared* before being used.
- Variable declaration establishes the particular *data type*.
- Eight primitive data types built into the language are:

integers:	floating point numbers:	character data type:	Boolean data type:
> byte	> short	> float	> char
> int	> double		> boolean
> long			
- Objects are created from a class, which is another data type.

Copyright © 2009 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

6-6

Operators in Java and Alice

	Alice	Java
Assignment	set value	=
Basic Arithmetic	$a_1 - a_2, /$	$a_1 - a_2, /$
Remainder	IEEEremainder	%
Equality	$a_1 == a_2$	$a_1 == a_2$
Relational	$a_1 < a_2, > a_2$	$a_1 < a_2, > a_2$
Logical NOT	not a	!a
Logical AND	both a and b	a & b
Logical OR	either a or b, or both	a b

Copyright © 2009 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

6-7

Java Statements

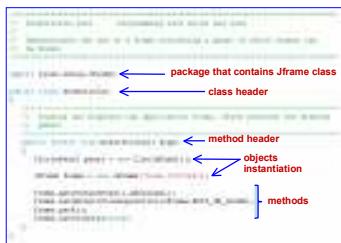
- The **assignment** statement: `total = total + 25;`
- The **print** statement uses two methods, **print** and **println**:
`System.out.println("The result is " + total);`
- The **if-else** statement:
`if (height > 69)
 height = height / 2;
else
 System.out.println("Current height: " + height);`
- A **main** method is the default starting point in every Java program.

Copyright © 2009 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

6-8

Java Classes and Objects

- Each circle is an object;
- The window **frame** is another object;
- Each object has properties: color, size, location;



- When the program is executed, a **main** method is executed first and drives everything else;
- The **DrawCircles** class is the program **driver**.

Copyright © 2009 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

6-9

The CirclePanel Class

```
1 package com.pearson.ait.alice.alice;
2
3 import javax.swing.JFrame;
4 import javax.swing.JPanel;
5 import javax.swing.WindowConstants;
6
7 public class CirclePanel extends JPanel {
8
9     public void paintComponent(Graphics g) {
10        g.drawOval(100, 100, 100, 100);
11        g.drawOval(200, 200, 100, 100);
12        g.drawOval(300, 300, 100, 100);
13    }
14 }
15
16
17 package com.pearson.ait.alice.alice;
18
19 import javax.swing.JFrame;
20 import javax.swing.JPanel;
21
22 public class DrawCircles extends JFrame {
23
24     public static void main(String[] args) {
25         DrawCircles driver = new DrawCircles();
26     }
27 }
28
29
30 package com.pearson.ait.alice.alice;
31
32 import javax.swing.JPanel;
33
34 public class Circle {
35
36     int x, y, radius;
37
38     public Circle(int x, int y, int radius) {
39         this.x = x;
40         this.y = y;
41         this.radius = radius;
42     }
43 }
44
45
46 package com.pearson.ait.alice.alice;
47
48 import javax.swing.JPanel;
49
50 public class CirclePanel extends JPanel {
51
52     public void paintComponent(Graphics g) {
53         g.drawOval(100, 100, 100, 100);
54         g.drawOval(200, 200, 100, 100);
55         g.drawOval(300, 300, 100, 100);
56     }
57 }
58
59
60 package com.pearson.ait.alice.alice;
61
62 import javax.swing.JFrame;
63 import javax.swing.JPanel;
64
65 public class DrawCircles extends JFrame {
66
67     public static void main(String[] args) {
68         DrawCircles driver = new DrawCircles();
69     }
70 }
71
72
73 package com.pearson.ait.alice.alice;
74
75 import javax.swing.JPanel;
76
77 public class Circle {
78
79     int x, y, radius;
80
81     public Circle(int x, int y, int radius) {
82         this.x = x;
83         this.y = y;
84         this.radius = radius;
85     }
86 }
87
88
89 package com.pearson.ait.alice.alice;
90
91 import javax.swing.JPanel;
92
93 public class CirclePanel extends JPanel {
94
95     public void paintComponent(Graphics g) {
96         g.drawOval(100, 100, 100, 100);
97         g.drawOval(200, 200, 100, 100);
98         g.drawOval(300, 300, 100, 100);
99     }
100 }
101
102
103 package com.pearson.ait.alice.alice;
104
105 import javax.swing.JFrame;
106 import javax.swing.JPanel;
107
108 public class DrawCircles extends JFrame {
109
110     public static void main(String[] args) {
111         DrawCircles driver = new DrawCircles();
112     }
113 }
114
115
116 package com.pearson.ait.alice.alice;
117
118 import javax.swing.JPanel;
119
120 public class Circle {
121
122     int x, y, radius;
123
124     public Circle(int x, int y, int radius) {
125         this.x = x;
126         this.y = y;
127         this.radius = radius;
128     }
129 }
130
131
132 package com.pearson.ait.alice.alice;
133
134 import javax.swing.JPanel;
135
136 public class CirclePanel extends JPanel {
137
138     public void paintComponent(Graphics g) {
139         g.drawOval(100, 100, 100, 100);
140         g.drawOval(200, 200, 100, 100);
141         g.drawOval(300, 300, 100, 100);
142     }
143 }
144
145
146 package com.pearson.ait.alice.alice;
147
148 import javax.swing.JFrame;
149 import javax.swing.JPanel;
150
151 public class DrawCircles extends JFrame {
152
153     public static void main(String[] args) {
154         DrawCircles driver = new DrawCircles();
155     }
156 }
157
158
159 package com.pearson.ait.alice.alice;
160
161 import javax.swing.JPanel;
162
163 public class Circle {
164
165     int x, y, radius;
166
167     public Circle(int x, int y, int radius) {
168         this.x = x;
169         this.y = y;
170         this.radius = radius;
171     }
172 }
173
174
175 package com.pearson.ait.alice.alice;
176
177 import javax.swing.JPanel;
178
179 public class CirclePanel extends JPanel {
180
181     public void paintComponent(Graphics g) {
182         g.drawOval(100, 100, 100, 100);
183         g.drawOval(200, 200, 100, 100);
184         g.drawOval(300, 300, 100, 100);
185     }
186 }
187
188
189 package com.pearson.ait.alice.alice;
190
191 import javax.swing.JFrame;
192 import javax.swing.JPanel;
193
194 public class DrawCircles extends JFrame {
195
196     public static void main(String[] args) {
197         DrawCircles driver = new DrawCircles();
198     }
199 }
200
201
202 package com.pearson.ait.alice.alice;
203
204 import javax.swing.JPanel;
205
206 public class Circle {
207
208     int x, y, radius;
209
210     public Circle(int x, int y, int radius) {
211         this.x = x;
212         this.y = y;
213         this.radius = radius;
214     }
215 }
216
217
218 package com.pearson.ait.alice.alice;
219
220 import javax.swing.JPanel;
221
222 public class CirclePanel extends JPanel {
223
224     public void paintComponent(Graphics g) {
225         g.drawOval(100, 100, 100, 100);
226         g.drawOval(200, 200, 100, 100);
227         g.drawOval(300, 300, 100, 100);
228     }
229 }
230
231
232 package com.pearson.ait.alice.alice;
233
234 import javax.swing.JFrame;
235 import javax.swing.JPanel;
236
237 public class DrawCircles extends JFrame {
238
239     public static void main(String[] args) {
240         DrawCircles driver = new DrawCircles();
241     }
242 }
243
244
245 package com.pearson.ait.alice.alice;
246
247 import javax.swing.JPanel;
248
249 public class Circle {
250
251     int x, y, radius;
252
253     public Circle(int x, int y, int radius) {
254         this.x = x;
255         this.y = y;
256         this.radius = radius;
257     }
258 }
259
260
261 package com.pearson.ait.alice.alice;
262
263 import javax.swing.JPanel;
264
265 public class CirclePanel extends JPanel {
266
267     public void paintComponent(Graphics g) {
268         g.drawOval(100, 100, 100, 100);
269         g.drawOval(200, 200, 100, 100);
270         g.drawOval(300, 300, 100, 100);
271     }
272 }
273
274
275 package com.pearson.ait.alice.alice;
276
277 import javax.swing.JFrame;
278 import javax.swing.JPanel;
279
280 public class DrawCircles extends JFrame {
281
282     public static void main(String[] args) {
283         DrawCircles driver = new DrawCircles();
284     }
285 }
286
287
288 package com.pearson.ait.alice.alice;
289
290 import javax.swing.JPanel;
291
292 public class Circle {
293
294     int x, y, radius;
295
296     public Circle(int x, int y, int radius) {
297         this.x = x;
298         this.y = y;
299         this.radius = radius;
300     }
301 }
302
303
304 package com.pearson.ait.alice.alice;
305
306 import javax.swing.JPanel;
307
308 public class CirclePanel extends JPanel {
309
310     public void paintComponent(Graphics g) {
311         g.drawOval(100, 100, 100, 100);
312         g.drawOval(200, 200, 100, 100);
313         g.drawOval(300, 300, 100, 100);
314     }
315 }
316
317
318 package com.pearson.ait.alice.alice;
319
320 import javax.swing.JFrame;
321 import javax.swing.JPanel;
322
323 public class DrawCircles extends JFrame {
324
325     public static void main(String[] args) {
326         DrawCircles driver = new DrawCircles();
327     }
328 }
329
330
331 package com.pearson.ait.alice.alice;
332
333 import javax.swing.JPanel;
334
335 public class Circle {
336
337     int x, y, radius;
338
339     public Circle(int x, int y, int radius) {
340         this.x = x;
341         this.y = y;
342         this.radius = radius;
343     }
344 }
345
346
347 package com.pearson.ait.alice.alice;
348
349 import javax.swing.JPanel;
350
351 public class CirclePanel extends JPanel {
352
353     public void paintComponent(Graphics g) {
354         g.drawOval(100, 100, 100, 100);
355         g.drawOval(200, 200, 100, 100);
356         g.drawOval(300, 300, 100, 100);
357     }
358 }
359
360
361 package com.pearson.ait.alice.alice;
362
363 import javax.swing.JFrame;
364 import javax.swing.JPanel;
365
366 public class DrawCircles extends JFrame {
367
368     public static void main(String[] args) {
369         DrawCircles driver = new DrawCircles();
370     }
371 }
372
373
374 package com.pearson.ait.alice.alice;
375
376 import javax.swing.JPanel;
377
378 public class Circle {
379
380     int x, y, radius;
381
382     public Circle(int x, int y, int radius) {
383         this.x = x;
384         this.y = y;
385         this.radius = radius;
386     }
387 }
388
389
390 package com.pearson.ait.alice.alice;
391
392 import javax.swing.JPanel;
393
394 public class CirclePanel extends JPanel {
395
396     public void paintComponent(Graphics g) {
397         g.drawOval(100, 100, 100, 100);
398         g.drawOval(200, 200, 100, 100);
399         g.drawOval(300, 300, 100, 100);
400     }
401 }
402
403
404 package com.pearson.ait.alice.alice;
405
406 import javax.swing.JFrame;
407 import javax.swing.JPanel;
408
409 public class DrawCircles extends JFrame {
410
411     public static void main(String[] args) {
412         DrawCircles driver = new DrawCircles();
413     }
414 }
415
416
417 package com.pearson.ait.alice.alice;
418
419 import javax.swing.JPanel;
420
421 public class Circle {
422
423     int x, y, radius;
424
425     public Circle(int x, int y, int radius) {
426         this.x = x;
427         this.y = y;
428         this.radius = radius;
429     }
430 }
431
432
433 package com.pearson.ait.alice.alice;
434
435 import javax.swing.JPanel;
436
437 public class CirclePanel extends JPanel {
438
439     public void paintComponent(Graphics g) {
440         g.drawOval(100, 100, 100, 100);
441         g.drawOval(200, 200, 100, 100);
442         g.drawOval(300, 300, 100, 100);
443     }
444 }
445
446
447 package com.pearson.ait.alice.alice;
448
449 import javax.swing.JFrame;
450 import javax.swing.JPanel;
451
452 public class DrawCircles extends JFrame {
453
454     public static void main(String[] args) {
455         DrawCircles driver = new DrawCircles();
456     }
457 }
458
459
460 package com.pearson.ait.alice.alice;
461
462 import javax.swing.JPanel;
463
464 public class Circle {
465
466     int x, y, radius;
467
468     public Circle(int x, int y, int radius) {
469         this.x = x;
470         this.y = y;
471         this.radius = radius;
472     }
473 }
474
475
476 package com.pearson.ait.alice.alice;
477
478 import javax.swing.JPanel;
479
480 public class CirclePanel extends JPanel {
481
482     public void paintComponent(Graphics g) {
483         g.drawOval(100, 100, 100, 100);
484         g.drawOval(200, 200, 100, 100);
485         g.drawOval(300, 300, 100, 100);
486     }
487 }
488
489
490 package com.pearson.ait.alice.alice;
491
492 import javax.swing.JFrame;
493 import javax.swing.JPanel;
494
495 public class DrawCircles extends JFrame {
496
497     public static void main(String[] args) {
498         DrawCircles driver = new DrawCircles();
499     }
500 }
501
502
503 package com.pearson.ait.alice.alice;
504
505 import javax.swing.JPanel;
506
507 public class Circle {
508
509     int x, y, radius;
510
511     public Circle(int x, int y, int radius) {
512         this.x = x;
513         this.y = y;
514         this.radius = radius;
515     }
516 }
517
518
519 package com.pearson.ait.alice.alice;
520
521 import javax.swing.JPanel;
522
523 public class CirclePanel extends JPanel {
524
525     public void paintComponent(Graphics g) {
526         g.drawOval(100, 100, 100, 100);
527         g.drawOval(200, 200, 100, 100);
528         g.drawOval(300, 300, 100, 100);
529     }
530 }
531
532
533 package com.pearson.ait.alice.alice;
534
535 import javax.swing.JFrame;
536 import javax.swing.JPanel;
537
538 public class DrawCircles extends JFrame {
539
540     public static void main(String[] args) {
541         DrawCircles driver = new DrawCircles();
542     }
543 }
544
545
546 package com.pearson.ait.alice.alice;
547
548 import javax.swing.JPanel;
549
550 public class Circle {
551
552     int x, y, radius;
553
554     public Circle(int x, int y, int radius) {
555         this.x = x;
556         this.y = y;
557         this.radius = radius;
558     }
559 }
560
561
562 package com.pearson.ait.alice.alice;
563
564 import javax.swing.JPanel;
565
566 public class CirclePanel extends JPanel {
567
568     public void paintComponent(Graphics g) {
569         g.drawOval(100, 100, 100, 100);
570         g.drawOval(200, 200, 100, 100);
571         g.drawOval(300, 300, 100, 100);
572     }
573 }
574
575
576 package com.pearson.ait.alice.alice;
577
578 import javax.swing.JFrame;
579 import javax.swing.JPanel;
580
581 public class DrawCircles extends JFrame {
582
583     public static void main(String[] args) {
584         DrawCircles driver = new DrawCircles();
585     }
586 }
587
588
589 package com.pearson.ait.alice.alice;
590
591 import javax.swing.JPanel;
592
593 public class Circle {
594
595     int x, y, radius;
596
597     public Circle(int x, int y, int radius) {
598         this.x = x;
599         this.y = y;
600         this.radius = radius;
601     }
602 }
603
604
605 package com.pearson.ait.alice.alice;
606
607 import javax.swing.JPanel;
608
609 public class CirclePanel extends JPanel {
610
611     public void paintComponent(Graphics g) {
612         g.drawOval(100, 100, 100, 100);
613         g.drawOval(200, 200, 100, 100);
614         g.drawOval(300, 300, 100, 100);
615     }
616 }
617
618
619 package com.pearson.ait.alice.alice;
620
621 import javax.swing.JFrame;
622 import javax.swing.JPanel;
623
624 public class DrawCircles extends JFrame {
625
626     public static void main(String[] args) {
627         DrawCircles driver = new DrawCircles();
628     }
629 }
630
631
632 package com.pearson.ait.alice.alice;
633
634 import javax.swing.JPanel;
635
636 public class Circle {
637
638     int x, y, radius;
639
640     public Circle(int x, int y, int radius) {
641         this.x = x;
642         this.y = y;
643         this.radius = radius;
644     }
645 }
646
647
648 package com.pearson.ait.alice.alice;
649
650 import javax.swing.JPanel;
651
652 public class CirclePanel extends JPanel {
653
654     public void paintComponent(Graphics g) {
655         g.drawOval(100, 100, 100, 100);
656         g.drawOval(200, 200, 100, 100);
657         g.drawOval(300, 300, 100, 100);
658     }
659 }
660
661
662 package com.pearson.ait.alice.alice;
663
664 import javax.swing.JFrame;
665 import javax.swing.JPanel;
666
667 public class DrawCircles extends JFrame {
668
669     public static void main(String[] args) {
670         DrawCircles driver = new DrawCircles();
671     }
672 }
673
674
675 package com.pearson.ait.alice.alice;
676
677 import javax.swing.JPanel;
678
679 public class Circle {
680
681     int x, y, radius;
682
683     public Circle(int x, int y, int radius) {
684         this.x = x;
685         this.y = y;
686         this.radius = radius;
687     }
688 }
689
690
691 package com.pearson.ait.alice.alice;
692
693 import javax.swing.JPanel;
694
695 public class CirclePanel extends JPanel {
696
697     public void paintComponent(Graphics g) {
698         g.drawOval(100, 100, 100, 100);
699         g.drawOval(200, 200, 100, 100);
700         g.drawOval(300, 300, 100, 100);
701     }
702 }
703
704
705 package com.pearson.ait.alice.alice;
706
707 import javax.swing.JFrame;
708 import javax.swing.JPanel;
709
710 public class DrawCircles extends JFrame {
711
712     public static void main(String[] args) {
713         DrawCircles driver = new DrawCircles();
714     }
715 }
716
717
718 package com.pearson.ait.alice.alice;
719
720 import javax.swing.JPanel;
721
722 public class Circle {
723
724     int x, y, radius;
725
726     public Circle(int x, int y, int radius) {
727         this.x = x;
728         this.y = y;
729         this.radius = radius;
730     }
731 }
732
733
734 package com.pearson.ait.alice.alice;
735
736 import javax.swing.JPanel;
737
738 public class CirclePanel extends JPanel {
739
740     public void paintComponent(Graphics g) {
741         g.drawOval(100, 100, 100, 100);
742         g.drawOval(200, 200, 100, 100);
743         g.drawOval(300, 300, 100, 100);
744     }
745 }
746
747
748 package com.pearson.ait.alice.alice;
749
750 import javax.swing.JFrame;
751 import javax.swing.JPanel;
752
753 public class DrawCircles extends JFrame {
754
755     public static void main(String[] args) {
756         DrawCircles driver = new DrawCircles();
757     }
758 }
759
760
761 package com.pearson.ait.alice.alice;
762
763 import javax.swing.JPanel;
764
765 public class Circle {
766
767     int x, y, radius;
768
769     public Circle(int x, int y, int radius) {
770         this.x = x;
771         this.y = y;
772         this.radius = radius;
773     }
774 }
775
776
777 package com.pearson.ait.alice.alice;
778
779 import javax.swing.JPanel;
780
781 public class CirclePanel extends JPanel {
782
783     public void paintComponent(Graphics g) {
784         g.drawOval(100, 100, 100, 100);
785         g.drawOval(200, 200, 100, 100);
786         g.drawOval(300, 300, 100, 100);
787     }
788 }
789
790
791 package com.pearson.ait.alice.alice;
792
793 import javax.swing.JFrame;
794 import javax.swing.JPanel;
795
796 public class DrawCircles extends JFrame {
797
798     public static void main(String[] args) {
799         DrawCircles driver = new DrawCircles();
800     }
801 }
802
803
804 package com.pearson.ait.alice.alice;
805
806 import javax.swing.JPanel;
807
808 public class Circle {
809
810     int x, y, radius;
811
812     public Circle(int x, int y, int radius) {
813         this.x = x;
814         this.y = y;
815         this.radius = radius;
816     }
817 }
818
819
820 package com.pearson.ait.alice.alice;
821
822 import javax.swing.JPanel;
823
824 public class CirclePanel extends JPanel {
825
826     public void paintComponent(Graphics g) {
827         g.drawOval(100, 100, 100, 100);
828         g.drawOval(200, 200, 100, 100);
829         g.drawOval(300, 300, 100, 100);
830     }
831 }
832
833
834 package com.pearson.ait.alice.alice;
835
836 import javax.swing.JFrame;
837 import javax.swing.JPanel;
838
839 public class DrawCircles extends JFrame {
840
841     public static void main(String[] args) {
842         DrawCircles driver = new DrawCircles();
843     }
844 }
845
846
847 package com.pearson.ait.alice.alice;
848
849 import javax.swing.JPanel;
850
851 public class Circle {
852
853     int x, y, radius;
854
855     public Circle(int x, int y, int radius) {
856         this.x = x;
857         this.y = y;
858         this.radius = radius;
859     }
860 }
861
862
863 package com.pearson.ait.alice.alice;
864
865 import javax.swing.JPanel;
866
867 public class CirclePanel extends JPanel {
868
869     public void paintComponent(Graphics g) {
870         g.drawOval(100, 100, 100, 100);
871         g.drawOval(200, 200, 100, 100);
872         g.drawOval(300, 300, 100, 100);
873     }
874 }
875
876
877 package com.pearson.ait.alice.alice;
878
879 import javax.swing.JFrame;
880 import javax.swing.JPanel;
881
882 public class DrawCircles extends JFrame {
883
884     public static void main(String[] args) {
885         DrawCircles driver = new DrawCircles();
886     }
887 }
888
889
890 package com.pearson.ait.alice.alice;
891
892 import javax.swing.JPanel;
893
894 public class Circle {
895
896     int x, y, radius;
897
898     public Circle(int x, int y, int radius) {
899         this.x = x;
900         this.y = y;
901         this.radius = radius;
902     }
903 }
904
905
906 package com.pearson.ait.alice.alice;
907
908 import javax.swing.JPanel;
909
910 public class CirclePanel extends JPanel {
911
912     public void paintComponent(Graphics g) {
913         g.drawOval(100, 100, 100, 100);
914         g.drawOval(200, 200, 100, 100);
915         g.drawOval(300, 300, 100, 100);
916     }
917 }
918
919
920 package com.pearson.ait.alice.alice;
921
922 import javax.swing.JFrame;
923 import javax.swing.JPanel;
924
925 public class DrawCircles extends JFrame {
926
927     public static void main(String[] args) {
928         DrawCircles driver = new DrawCircles();
929     }
930 }
931
932
933 package com.pearson.ait.alice.alice;
934
935 import javax.swing.JPanel;
936
937 public class Circle {
938
939     int x, y, radius;
940
941     public Circle(int x, int y, int radius) {
942         this.x = x;
943         this.y = y;
944         this.radius = radius;
945     }
946 }
947
948
949 package com.pearson.ait.alice.alice;
950
951 import javax.swing.JPanel;
952
953 public class CirclePanel extends JPanel {
954
955     public void paintComponent(Graphics g) {
956         g.drawOval(100, 100, 100, 100);
957         g.drawOval(200, 200, 100, 100);
958         g.drawOval(300, 300, 100, 100);
959     }
960 }
961
962
963 package com.pearson.ait.alice.alice;
964
965 import javax.swing.JFrame;
966 import javax.swing.JPanel;
967
968 public class DrawCircles extends JFrame {
969
970     public static void main(String[] args) {
971         DrawCircles driver = new DrawCircles();
972     }
973 }
974
975
976 package com.pearson.ait.alice.alice;
977
978 import javax.swing.JPanel;
979
980 public class Circle {
981
982     int x, y, radius;
983
984     public Circle(int x, int y, int radius) {
985         this.x = x;
986         this.y = y;
987         this.radius = radius;
988     }
989 }
990
991
992 package com.pearson.ait.alice.alice;
993
994 import javax.swing.JPanel;
995
996 public class CirclePanel extends JPanel {
997
998     public void paintComponent(Graphics g) {
999         g.drawOval(100, 100, 100, 100);
1000         g.drawOval(200, 200, 100, 100);
1001         g.drawOval(300, 300, 100, 100);
1002     }
1003 }
1004
1005
1006 package com.pearson.ait.alice.alice;
1007
1008 import javax.swing.JFrame;
1009 import javax.swing.JPanel;
1010
1011 public class DrawCircles extends JFrame {
1012
1013     public static void main(String[] args) {
1014         DrawCircles driver = new DrawCircles();
1015     }
1016 }
1017
1018
1019 package com.pearson.ait.alice.alice;
1020
1021 import javax.swing.JPanel;
1022
1023 public class Circle {
1024
1025     int x, y, radius;
1026
1027     public Circle(int x, int y, int radius) {
1028         this.x = x;
1029         this.y = y;
1030         this.radius = radius;
1031     }
1032 }
1033
1034
1035 package com.pearson.ait.alice.alice;
1036
1037 import javax.swing.JPanel;
1038
1039 public class CirclePanel extends JPanel {
1040
1041     public void paintComponent(Graphics g) {
1042         g.drawOval(100, 100, 100, 100);
1043         g.drawOval(200, 200, 100, 100);
1044         g.drawOval(300, 300, 100, 100);
1045     }
1046 }
1047
1048
1049 package com.pearson.ait.alice.alice;
1050
1051 import javax.swing.JFrame;
1052 import javax.swing.JPanel;
1053
1054 public class DrawCircles extends JFrame {
1055
1056     public static void main(String[] args) {
1057         DrawCircles driver = new DrawCircles();
1058     }
1059 }
1060
1061
1062 package com.pearson.ait.alice.alice;
1063
1064 import javax.swing.JPanel;
1065
1066 public class Circle {
1067
1068     int x, y, radius;
1069
1070     public Circle(int x, int y, int radius) {
1071         this.x = x;
1072         this.y = y;
1073         this.radius = radius;
1074     }
1075 }
1076
1077
1078 package com.pearson.ait.alice.alice;
1079
1080 import javax.swing.JPanel;
1081
1082 public class CirclePanel extends JPanel {
1083
1084     public void paintComponent(Graphics g) {
1085         g.drawOval(100, 100, 100, 100);
1086         g.drawOval(200, 200, 100, 100);
1087         g.drawOval(300, 300, 100, 100);
1088     }
1089 }
1090
1091
1092 package com.pearson.ait.alice.alice;
1093
1094 import javax.swing.JFrame;
1095 import javax.swing.JPanel;
1096
1097 public class DrawCircles extends JFrame {
1098
1099     public static void main(String[] args) {
1100         DrawCircles driver = new DrawCircles();
1101     }
1102 }
1103
1104
1105 package com.pearson.ait.alice.alice;
1106
1107 import javax.swing.JPanel;
1108
1109 public class Circle {
1110
1111     int x, y, radius;
1112
1113     public Circle(int x, int y, int radius) {
1114         this.x = x;
1115         this.y = y;
1116         this.radius = radius;
1117     }
1118 }
1119
1120
1121 package com.pearson.ait.alice.alice;
1122
1123 import javax.swing.JPanel;
1124
1125 public class CirclePanel extends JPanel {
1126
1127     public void paintComponent(Graphics g) {
1128         g.drawOval(100, 100, 100, 100);
1129         g.drawOval(200, 200, 100, 100);
1130         g.drawOval(300, 300
```

The **for** Loop Example



```
1// Insert the following code.
2public class CoinFlip {
3    public static void main(String[] args) {
4        Scanner input = new Scanner(System.in);
5        int n = 0, p = 0, diameter = 1000_0000;
6        String[] coinResults = new String[n];
7        for (int i = 0; i < n; i++) {
8            coinResults[i] = flip(diameter);
9        }
10    }
11    public String flip(int diameter) {
12        int r = (int) (Math.random() * (diameter - 1) + 0.5);
13        if (r < diameter / 2) {
14            return "Heads";
15        } else {
16            return "Tails";
17        }
18    }
19}
```

Copyright © 2009 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

6-13

Program Example

- The **CoinFlip** program flips a virtual coin.

- Sample run:

```
Run: main CoinFlip (1-1234) 5
Run 1: coin, flip many Coinflip (1-1234) 1-1234
Run 2: coin, flip many Coinflip (1-1234) 2-1234
Run 3: coin, flip many Coinflip (1-1234) 3-1234
Run 4: coin, flip many Coinflip (1-1234) 4-1234
Run 5: coin, flip many Coinflip (1-1234) 5-1234
Initial inventory of coins: 1000
Header 200 - Tails: 242
```

- The **CoinFlip** class contains the main method of the program.
- The **Scanner** class allows you to read input from the keyboard.
- The **Coin** class represents a single coin that can be flipped.

Copyright © 2009 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

6-14

Program Example (continued)

```
1// Insert the following code.
2public class CoinFlip {
3    public static void main(String[] args) {
4        Scanner input = new Scanner(System.in);
5        System.out.println("How many times would you like to flip the coin? ");
6        int n = input.nextInt();
7        if (n < 0) {
8            System.out.println("Please enter a positive integer.");
9            System.out.println("Please enter a positive integer.");
10        }
11        for (int i = 0; i < n; i++) {
12            flip();
13        }
14    }
15    public void flip() {
16        int r = (int) (Math.random() * 1000_0000);
17        if (r < 5000000) {
18            System.out.println("Heads");
19        } else {
20            System.out.println("Tails");
21        }
22    }
23}
```

Copyright © 2009 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

6-15

The while loop performs input validation

The for loop flips the coin as many times as was specified

Summary

- You can use one of several development environments to create Java programs.
- Java code is compiled and translated into bytecode before it is executed.
- A program may contain syntax errors, which will be caught by compiler.
- Logic errors cannot be caught by a compiler.
- A constructor is used to set up a newly created object.
- The Java API is a library of classes that we can use in any Java program.
- Encapsulation: the concept that each object should manage its own data and prevent explicit external modifications.
- The **toString** method is called automatically when an object is printed.

Copyright © 2009 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

6-16