

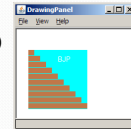
# Building Java Programs

Graphics (Part Deux)

Copyright 2010 by Pearson Education

## Java book figure

- Write a program that draws the following figure:
  - drawing panel is size 200x150
  - book is at (20, 35), size 100x100
  - cyan background
  - white "BJP" text at position (70, 55)
  - stairs are in color (red=191, green=118, blue=73)
  - each stair is 9px tall
    - 1st stair is 10px wide
    - 2nd stair is 20px wide ...
  - stairs are 10px apart (1 blank pixel between)



2

## Java book solution

```
// Draws a Building Java Programs textbook with DrawingPanel.
import java.awt.*;

public class Book {
    public static void main(String[] args) {
        DrawingPanel panel = new DrawingPanel(200, 150);
        panel.setBackground(Color.WHITE);
        Graphics g = panel.getGraphics();

        g.setColor(Color.CYAN);           // cyan background
        g.fillRect(20, 35, 100, 100);

        g.setColor(Color.WHITE);         // white "bjp" text
        g.drawString("BJP", 70, 55);

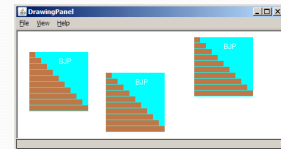
        g.setColor(new Color(191, 118, 73));
        for (int i = 0; i < 10; i++) {    // orange "bricks"
            g.fillRect(20, 35 + 10 * i, 10 + 10 * i, 9);
        }
    }
}
```

Copyright 2010 by Pearson Education

3

## Multiple Java books

- Modify the Java book program so that it can draw books at different *positions* as shown below.
  - book top/left positions: (20, 35), (150, 70), (300, 10)
  - drawing panel's new size: 450x180



Copyright 2010 by Pearson Education

4

## Multiple books solution

- To draw in a method, you must pass Graphics g to it.

```
// Draws many BJP textbooks using parameters.
import java.awt.*;

public class Book2 {
    public static void main(String[] args) {
        DrawingPanel panel = new DrawingPanel(450, 180);
        panel.setBackground(Color.WHITE);
        Graphics g = panel.getGraphics();

        // draw three books at different locations
        drawBook(g, 20, 35);
        drawBook(g, 150, 70);
        drawBook(g, 300, 10);
    }
    ...
}
```

Copyright 2010 by Pearson Education

5

## Multiple books, cont'd.

```
...
// Draws a BJP textbook at the given x/y position.
public static void drawBook(Graphics g, int x, int y) {
    g.setColor(Color.CYAN);           // cyan background
    g.fillRect(x, y, 100, 100);

    g.setColor(Color.WHITE);         // white "bjp" text
    g.drawString("BJP", x + 50, y + 20);

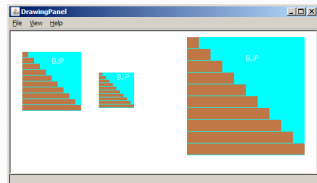
    g.setColor(new Color(191, 118, 73));
    for (int i = 0; i < 10; i++) {    // orange "bricks"
        g.fillRect(x, y + 10 * i, 10 * (i + 1), 9);
    }
}
}
```

Copyright 2010 by Pearson Education

6

## Resizable Java books

- Modify the Java book program so that it can draw books at different sizes as shown below.
  - book sizes: 100x100, 60x60, 200x200
  - drawing panel's new size: 520x240



Copyright 2010 by Pearson Education

7

## Resizable books solution

```
// Draws many sized BJP textbooks using parameters.
import java.awt.*;

public class Book3 {
    public static void main(String[] args) {
        DrawingPanel panel = new DrawingPanel(520, 240);
        panel.setBackground(Color.WHITE);
        Graphics g = panel.getGraphics();

        // draw three books at different locations/sizes
        drawBook(g, 20, 35, 100);
        drawBook(g, 150, 70, 60);
        drawBook(g, 300, 10, 200);
    }
    ...
}
```

Copyright 2010 by Pearson Education

8

## Resizable solution, cont'd.

```
...
// Draws a book of the given size at the given position.
public static void drawBook(Graphics g, int x, int y, int size) {
    g.setColor(Color.CYAN); // cyan background
    g.fillRect(x, y, size, size);

    g.setColor(Color.WHITE); // white "bjp" text
    g.drawString("BJP", x + size/2, y + size/5);

    g.setColor(new Color(191, 118, 73)); // orange "bricks"
    for (int i = 0; i < 10; i++) {
        g.fillRect(x, // x
                  y + size/10 * i, // y
                  size/10 * (i + 1), // width
                  size/10 - 1); // height
    }
}
}
```

Copyright 2010 by Pearson Education

9

## DrawingPanel methods

- **panel.clear()**;  
Erases any shapes that are drawn on the drawing panel.
- **panel.setWidth(width)** ;  
**panel.setHeight(height)** ;  
**panel.setSize(width, height)** ;  
Changes the drawing panel's size to the given value(s).
- **panel.save(filename)** ;  
Saves the image on the panel to the given file (String).
- **panel.sleep(ms)** ;  
Pauses the drawing for the given number of milliseconds.

Copyright 2010 by Pearson Education

10

## Animation with sleep

- DrawingPanel's sleep method pauses your program for a given number of milliseconds.

- You can use sleep to create simple animations.

```
DrawingPanel panel = new DrawingPanel(250, 200);
Graphics g = panel.getGraphics();

g.setColor(Color.BLUE);
for (int i = 1; i <= 10; i++) {
    g.fillOval(15 * i, 15 * i, 30, 30);
    panel.sleep(500);
}
}
```

- Try adding sleep commands to loops in past exercises in this chapter and watch the panel draw itself piece by piece.

Copyright 2010 by Pearson Education

11

## Building Java Programs

### Chapter 3

#### Lecture 3-2: Math class; Return values

**reading: 3.2**

Copyright 2010 by Pearson Education

### Java's Math class

Method name	Description
Math.abs( <i>value</i> )	absolute value
Math.ceil( <i>value</i> )	rounds up
Math.floor( <i>value</i> )	rounds down
Math.log10( <i>value</i> )	logarithm, base 10
Math.max( <i>value1</i> , <i>value2</i> )	larger of two values
Math.min( <i>value1</i> , <i>value2</i> )	smaller of two values
Math.pow( <i>base</i> , <i>exp</i> )	base to the <i>exp</i> power
Math.random()	random double between 0 and 1
Math.round( <i>value</i> )	nearest whole number
Math.sqrt( <i>value</i> )	square root
Math.sin( <i>value</i> )	sine/cosine/tangent of an angle in radians
Math.cos( <i>value</i> )	
Math.tan( <i>value</i> )	
Math.toDegrees( <i>value</i> )	convert degrees to radians and back
Math.toRadians( <i>value</i> )	

Constant	Description
Math.E	2.7182818...
Math.PI	3.1415926...

13

### Math questions

- Evaluate the following expressions:
  - Math.abs(-1.23)
  - Math.pow(3, 2)
  - Math.pow(10, -2)
  - Math.sqrt(121.0) - Math.sqrt(256.0)
  - Math.round(Math.PI) + Math.round(Math.E)
  - Math.ceil(6.022) + Math.floor(15.9994)
  - Math.abs(Math.min(-3, -5))
- Math.max and Math.min can be used to bound numbers. Consider an int variable named age.
  - What statement would replace negative ages with 0?
  - What statement would cap the maximum age to 40?

14

### Calling Math methods

Math.**methodName** (**parameters**)

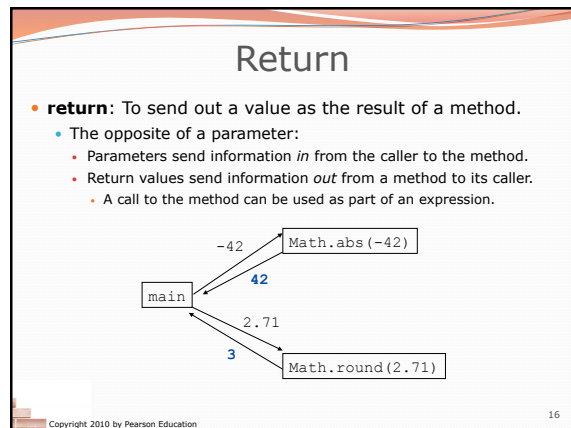
- Examples:
 

```
double squareRoot = Math.sqrt(121.0);
System.out.println(squareRoot); // 11.0

int absoluteValue = Math.abs(-50);
System.out.println(absoluteValue); // 50

System.out.println(Math.min(3, 7) + 2); // 5
```
- The Math methods do not print to the console.
  - Each method produces ("returns") a numeric result.
  - The results are used as expressions (printed, stored, etc.).

15



### Quirks of real numbers

- Some Math methods return double or other non-int types.
 

```
int x = Math.pow(10, 3); // ERROR: incompat. types
```
- Some double values print poorly (too many digits).
 

```
double result = 1.0 / 3.0;
System.out.println(result); // 0.33333333333333
```
- The computer represents doubles in an imprecise way.
 

```
System.out.println(0.1 + 0.2);
```

  - Instead of 0.3, the output is 0.30000000000000004

17

### Type casting

- type cast:** A conversion from one type to another.
  - To promote an int into a double to get exact division from /
  - To truncate a double from a real number to an integer
- Syntax:
 

```
(type) expression
```
- Examples:
 

```
double result = (double) 19 / 5; // 3.8
int result2 = (int) result; // 3
int x = (int) Math.pow(10, 3); // 1000
```

18

## More about type casting

- Type casting has high precedence and only casts the item immediately next to it.
  - `double x = (double) 1 + 1 / 2; // 1`
  - `double y = 1 + (double) 1 / 2; // 1.5`
- You can use parentheses to force evaluation order.
  - `double average = (double) (a + b + c) / 3;`
- A conversion to `double` can be achieved in other ways.
  - `double average = 1.0 * (a + b + c) / 3;`

Copyright 2010 by Pearson Education

19

## Exercise

- If you drop three balls, which will hit the ground first?
  - Ball 1: height of 600m, initial velocity = 25 m/sec downward
  - Ball 2: height of 400m, initial velocity = 0
  - Ball 3: height of 500m, initial velocity = 15 m/sec downward
- Write a program that draws each ball falling.
- Total time is based on the force of gravity on each ball.
  - Acceleration due to gravity  $\cong 9.81 \text{ m/s}^2$ , downward
  - Displacement =  $v_0 t + \frac{1}{2} a t^2$

Copyright 2010 by Pearson Education

20

## Returning a value

```
public static type name(parameters) {
    statements;
    ...
    return expression;
}
```

- Example:
  - `// Returns the slope of the line between the given points.`
  - `public static double slope(int x1, int y1, int x2, int y2) {`
  - `double dy = y2 - y1;`
  - `double dx = x2 - x1;`
  - `return dy / dx;`
  - `}`
  - `slope(1, 3, 5, 11)` returns 2.0

Copyright 2010 by Pearson Education

21

## Return examples

```
// Converts degrees Fahrenheit to Celsius.
public static double fToC(double degreesF) {
    double degreesC = 5.0 / 9.0 * (degreesF - 32);
    return degreesC;
}

// Computes triangle hypotenuse length given its side lengths.
public static double hypotenuse(int a, int b) {
    double c = Math.sqrt(a * a + b * b);
    return c;
}
```

- You can shorten the examples by returning an expression:

```
public static double fToC(double degreesF) {
    return 5.0 / 9.0 * (degreesF - 32);
}
```

Copyright 2010 by Pearson Education

22

## Common error: Not storing

- Many students incorrectly think that a `return` statement sends a variable's name back to the calling method.

```
public static void main(String[] args) {
    slope(0, 0, 6, 3);
    System.out.println("The slope is " + result); // ERROR:
} // result not defined

public static double slope(int x1, int x2, int y1, int y2) {
    double dy = y2 - y1;
    double dx = x2 - x1;
    double result = dy / dx;
    return result;
}
```

Copyright 2010 by Pearson Education

23

## Fixing the common error

- Instead, returning sends the variable's *value* back.
  - The returned value must be stored into a variable or used in an expression to be useful to the caller.

```
public static void main(String[] args) {
    double s = slope(0, 0, 6, 3);
    System.out.println("The slope is " + s);
}

public static double slope(int x1, int x2, int y1, int y2) {
    double dy = y2 - y1;
    double dx = x2 - x1;
    double result = dy / dx;
    return result;
}
```

Copyright 2010 by Pearson Education

24

hi

### Ball solution

```
// Simulates the dropping of three balls from various heights.
import java.awt.*;

public class Balls {
    public static void main(String[] args) {
        DrawingPanel panel = new DrawingPanel(600, 600);
        panel.setBackground(Color.CYAN);
        Graphics g = panel.getGraphics();

        // draw the balls at each time increment
        for (double t = 0; t <= 10.0; t = t + 0.1) {
            drawBall(g, 100, 0, 25, t); // initial velocity of 25
            drawBall(g, 200, 0, 0, t); // initial velocity of 0
            drawBall(g, 300, 10, 15, t); // initial velocity of 15
            panel.sleep(50); // pause for 50 ms
        }
    }
    ...
}
```

Copyright 2010 by Pearson Education 25

### Ball solution, cont'd.

```
...

// Draws the given ball point with the given initial velocity
// after the given amount of time has elapsed.
public static void drawBall(Graphics g, int x, int y, double v0, double t) {
    double disp = displacement(v0, t, 9.81);
    g.fillOval(x, y + (int) disp, 10, 10);
}

// Computes the displacement of a moving ball
// with the given initial velocity, acceleration, and time.
// displacement = v0 t + 1/2 a t^2
public static double displacement(double v0, double t, double a) {
    double d = v0 * t + 0.5 * a * Math.pow(t, 2);
    return d;
}
}
```

Copyright 2010 by Pearson Education 26

bye