CS 112 Introduction to Programming

Java Graphics

Yang (Richard) Yang
Computer Science Department
Yale University
208A Watson, Phone: 432-6400
Email: yry@cs.yale.edu
Outline

- Admin and recap
- Java graphics (StdDraw)
- Parameterized graphics methods and loops
Coding style review 6:30 pm Wednesday
PS2 due on Tuesday
PS3 to be posted (please start early on PS3)
  • Walk-through session?
  • Potentially combine w/ coding style?
Recap: DrawX

1. Bound
   • \( \text{== , } 2S - 4 \text{ spaces, ==} \)

2. for line = 1 to \( S -1 \)
   line spaces
   \]
   \2 S - 2 - 2 \* line spaces
   \]

3. for line = 1 to \( S - 1 \)
   S - line spaces
   \]
   (line - 1) \* 2 spaces
   \]

4. Bound
Recap: Identify Subtasks (Methods)

1. Bound
   • ==, 2S - 4 spaces, ==

2. for line = 1 to S - 1
   line spaces
   2 S - 2 - 2 * line spaces

3. for line = 1 to S - 1
   S - line spaces
   (line - 1) * 2 spaces

4. Bound

Drawing spaces is a reusable function, but need parameter to be reusable.
Method w/ Parameters

- Why: Redundancy removal/abstraction through generalization
- General: a method can accept multiple parameters. (separate by , )
  - When calling it, you must pass values for each parameter.
- Declaration:
  
  ```java
  public static void <name>(<type> <name>, ..., <type> <name>) {
    <statement>(s);
  }
  ```
- Call:
  
  ```java
  <name>(<exp>, <exp>, ..., <exp>);
  ```
public static void main(String[] args) {
    printNumber(4, 9);
    printNumber(17, 6);
    printNumber(8, 0);
    printNumber(0, 8);
}

public static void printNumber(int number, int count) {
    for (int i = 1; i <= count; i++) {
        System.out.print(number);
    }
    System.out.println();
}

Output:
44444444
1717171717
00000000
Multiple Parameter Invocation

- Corresponding actual argument in the invocation is copied into the corresponding formal argument

```java
public static void printNumber(int number, int count)
{
    // equiv: number = 4; count = 2;
    for (int i = 1; i <= count; i++) {
        System.out.print(number);
    } // equiv: number = 4; count = 2;
    System.out.println();
}

printNumber(1+3, 1+1);
```
Exercise

- Define displayChars to display a char ch count times
- Implement spaces by calling displayChars
Backup Slides of Last Friday

- It is another way to motivate the need of parameterized methods
Foundational Programming Concepts

any program you might want to write

- objects
- methods and classes
- graphics, sound, and image I/O
- arrays
- conditionals and loops
- math
- text I/O
- primitive data types
- assignment statements
Java Graphics

- Java provides a large number of methods for graphics.

- A graphical method may need to use a large number of parameters:
  - E.g., draw a line: line color, stroke width, stroke pattern, init pos, end pos.

- To avoid specifying a large of parameters in each method call, Java native graphics uses an object-oriented design: state parameters are contained (encapsulated) in entities called objects:
  - We will cover objects in more details later.
Java Graphics Wrapper Classes

To simplify the usage of Java native graphics, wrapper graphics classes are provided

- Back to Basics Textbook: defines class `DrawingPanel`, which is still an object-oriented design
- Sedgewick & Wayne book: avoids objects, by defining a class called `StdDraw`, which contains many static (non-object-oriented) graphics methods:
  - To access a static method defined in a class, use `<class-name>.<method-name>(...)`, for example,
    - `StdDraw.line (0, 0, 10, 10);`

- `DrawingPanel` and `StdDraw` are good examples that method designers may differ substantially in their designs, although for very similar functions
Coordinate system

- Each (x, y) position is *pixel* ("picture element") position
- Position (0, 0) is at the window's top-left corner.
  - x increases rightward and the y increases *downward*.

Example method rectangle:
- DrawingPanel: rect (int x0, int y0, int width, int height),
  - e.g., rect(0, 0, 200, 100)
StdDraw (The One we will use)

- **Coordinate system**
  - You still set canvas size using numbers of pixels:
    ```
    setCanvasSize(int w, int h)
    ```
  - But \((x, y)\) position is the coordinate in a **normalized** coordinate system
    - \([0, 1]\) as default
    - `setXScale(double x0, double x1)` to set x range
    - `setYScale(double y0, double y1)`
  - Position \((0, 0)\) is at the window's **lower-left** corner.
    - `x` increases rightward and `y` increases **upward**.

- **Example method rectangle:**
  - `rectangle(double cx, double cy, double halfWidth, double halfHeight);`

```java
StdDraw.setCanvasSize(500, 500);
StdDraw.setXScale(0, 500);
StdDraw.setYScale(0, 500);
StdDraw.rectangle(100, 450, 100, 50);
```
### Example StdDraw Methods

```java
void line(double x0, double y0, double x1, double y1)
void point(double x, double y)
void text(double x, double y, String s)
void circle(double x, double y, double r)
void filledCircle(double x, double y, double r)
void square(double x, double y, double r)
void filledSquare(double x, double y, double r)
void polygon(double[] x, double[] y)
void filledPolygon(double[] x, double[] y)
void setXscale(double x0, double x1)    // reset x range to (x₀, x₁)
void setYscale(double y0, double y1)    // reset y range to (y₀, y₁)
void setPenRadius(double r)            // set pen radius to r
void setPenColor(Color c)              // set pen color to c
void setFont(Font f)                   // set text font to f
void setCanvasSize(int w, int h)       // set canvas to w-by-h window
void clear(Color c)                     // clear the canvas; color it c
void show(int dt)                       // show all; pause dt milliseconds
void save(String filename)              // save to a .jpg or .png file
```

Example: StdDraw X

- Implement SimpleStdDrawX.java to draw an X using StdDraw
  - Try two approaches:
    1. Use the normalized coordinate system (0,0) - (1,1)
    2. Set the Xscale and Yscale
Example: StdDraw Loops

- You may use loop (nested loop) to produce more “complex” patterns.

- See SimpleStdDrawLoop.java and predict its display
Example: Draw using Loops
Example: Modify SimpleStdDrawLoop to Label a Number for each Cell

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
</tr>
<tr>
<td>48</td>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
</tr>
<tr>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
</tr>
<tr>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
</tr>
<tr>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>