Breakout YEAH hours

Brahm Capoor

Some slides edited by Nidhi Manoj
Road Map

- Lecture Review
- Using the debugger
- Assignment Overview
- Q&A!
Primitive variables

```java
int x = 7; // declare and initialize a variable
x = 9; // change the value of x
x = x + 1; // increment (add 1 to) x. A.K.A. x++
x = x + 2; // add 2 to x. A.K.A. x += 2
x /= 2 // divide x by 2, and truncate result

double d = 3.5;

boolean isThisTrue = true;
isThisTrue = !isThisTrue; // flip isThisTrue
```
Graphics

GRect rect = new GRect(50, 50, 200, 200);
rect.setFilled(true);
rect.setColor(Color.BLUE);

GOval oval = new GOval(0, 0, getWidth(), getHeight());
oval.setFilled(false);
oval.setColor(Color.GREEN);

GLabel text = new GLabel("banter", 200, 10);
add(text);
add(rect);
add(oval);

Things to remember

- Coordinates are **doubles**
- Coordinates are measured from the **top left** of the screen
- Coordinates of a shape are coordinates of its **top left corner**
- Coordinates of a label are coordinates of its **bottom left corner**
- Remember to **add** objects to the screen!
- Use the [online documentation](#)
- These are **class variables**!
Methods, parameters and variables
Parameters and a return value are both optional!

private returnType methodName(type parameter1, type parameter2,...)

private int returnsInt() {...}
private void drawsRect(int width, int length) {...} //void is no type
public boolean frontIsClear() {...} //look familiar?
Example: Methods and Parameters

```java
public void run() {
    println("Choose 2 numbers!");
    int n1 = readInt("Enter n1"); //5
    int n2 = readInt("Enter n2"); //7

    int total = addNumbers(n1, n2);
    println("The total is "+ total);
}

disprivate int addNumbers(int num1, int num2) {
    int sum = num1 + num2;  //12
    return sum;
}
```

GET n1 AND n2

run()

addNumbers(n1, n2)

num1 = 5, num2 = 7

sum = 12

PRINT RESULT

total = 12

Example: Methods and Parameters
Variable scope

Variables live inside the block in which they’re declared

```java
for (int i = 0; i < 5; i++) {
    int y = i * 4;
}
i = 3; // Error!
y = 2; // Error!
... // in some code far, far away
int y = 0;
for (int i = 0; i < 5; i++) {
    y = i * 4;
}
y = 2; // Ayy!
```
Instance variables

---

private int x; // belongs to the instance of the program

public void run() {
    x = 2;
    addTwo();
    println(x); // prints 4
}

private void addTwo() {
    x += 2;
}

Should you use an instance variable?

YES
- You access & change the variable everywhere
- You use it in mouseListener methods
- You have literally no other choice

NO
- It makes information flow more annoying to visualize (parameters are easier)
- Poor style to build up unnecessary instance variables

The opposite of an instance variable is a local variable
Returning in different places

private int multipleReturns(int x) {
    if (x == 5) {
        return 0;
    }
    return 1;  // this only happens if x != 5
    return 5;  // never gets to this line
}

// note: every path through the method ends with a single return statement

// note: a function ends immediately after it returns

---
Mouse Movement

---

addMouseListeners(); // this needs to happen before the program can respond to the mouse!

public void mouseMoved(MouseEvent e) {
    // remember to make this public!
    double mouseX = e.getX(); // get the x-coordinate of where the mouse moves to
    double mouseY = e.getY(); // get the x-coordinate of where the mouse moves to
    ...
}

Things to remember:

- Other things you can do with the mouse: mouseClicked(MouseEvent e), mouseDragged(MouseEvent e)
  - Check the textbook and the online documentation for more!
- mouseListeners are called parallel to your code, they happen as soon as you move the mouse
  - as long as you’ve called addMouseListeners() already!
Live demo: Using the debugger
Mouse Reporter

(A Sandcastle)

Tips and tricks

- The starter code stores the label as an instance variable
- `getElementAt` might be useful here!
Breakout!
Due Wednesday, February 7th
What we’re making
What you’re given

- These are **constants**
- Use `getWidth()` and `getHeight()` for dimensions of window, not the ones in the constants!
- You might need to add more instance variables…
MILESTONE 1: BRICKS

- Similar to pyramid!
- Drawing multiple rows
  - Figure out how to draw one row first
  - Bricks should be centered horizontally
- Reasonable coloring for any number of rows
MILESTONE 2: PADDLE

- How do you make the mouse control the paddle?
- Chapter 9: GObject Methods
- Chapter 10: Event Driven Programs (responding to mouse events)
- Things to consider:
  - Paddle only needs to move in the x direction
  - Paddle can’t move off the screen
Milestone 3: Play Ball!

- How do we move the ball?
- How do you choose the direction of the ball?
- What information do we need in the GOval constructor?
while (executing condition) {
    // update graphics
    obj.move(dx, dy);
    pause(PAUSE_TIME_MILLISEC);
}
Moving the ball

double vx;
double vy;
...

while(existing condition) {
    // update graphics
    ball.move(vx, vy);
    pause(PAUSE_TIME_MILLISEC);
}
Choosing the direction of the ball

---

// make a random generator instance variable
private RandomGenerator rgen = RandomGenerator.getInstance();

// give the ball an initial direction
vx = rgen.nextDouble(1.0, 3.0); // choose speed
if (rgen.nextBoolean(0.5)) vx = -vx; // choose left or right

// wait until player clicks the screen
waitForClick();
MILESTONE 4: COLLISIONS

Main idea: Check if there’s anything at each of the 4 corners and return one GObject

Useful method: public GObject getElementAt(double x, double y);
private GObject getCollidingObject() {
    // sick code
    // return a GObject
}

... 

GObject collider = getCollidingObject();
// only need to bounce vertically for collisions with brick, top wall and paddle
// only need to bounce horizontally for collisions with side walls

Things to think about: what direction needs to be flipped when?

This is just like the bouncing ball example in lecture!
Ending the game

- Remove the ball when it goes off the screen
  - remove(obj);

- Winning and losing
  - How? Bricks!
The sticky paddle 🤷‍♀️
Testing your program

- Check if it deals with changed constants
- Mega paddle
- Sticky paddle
- Crazy random player
Wrapping up

- Read the spec!
- Extensions!
- Commenting!
- Ask for help!
- Incorporate IG feedback!