Lecture 7: Classes

CS 51G

Spring 2018

Kim Bruce

Announcements

- Discuss Exercise 6.8.4
- Demo light balloon lab

Example Class

- DragAShirt
    - T-shirt class doesn't have to be in same file.
        - Just import it!
    - http://www.cs.pomona.edu/classes/cs051G/demos/Tshirt/DragAShirt.grace
    - http://www.cs.pomona.edu/classes/cs051G/demos/Tshirt/Tshirt.grace
- Add methods to Basketball/Tshirt class
    - sendToFront, reset, color, color:=,

Making it Harder

- We can drag two shirts, but now logic gets harder.
- Simpler if name them selectedShirt & otherShirt.
  - http://www.cs.pomona.edu/classes/cs051G/demos/Tshirt/DragTshirts.grace
- Swap names when select new one
  - If have doubts, try to write logic with redShirt, blueShirt.

Parameters & Arguments

- Methods have formal parameters:
  - method moveBy(dx: Number, dy: Number)
- When you want method to execute, provide arguments for it to use:
  - basketBall.moveBy(4,5)
- Pairs argument with parameter by position:
  - dx -> 4, dy -> 5
  - Like a def that lasts until method ends.

Parameters and Defs

- Defs & vars at the top level (*instance variables*) contain information that lives as long as the object
- Defs and vars declared inside blocks (*local* variables) last only as long as the block is being executed.
  - Same for method bodies (which are blocks)
  - Perfect when need to hold value temporarily
    - Random number used to choose color in part 1 of laundry
    - Sum of colors in part 2 of laundry

Parameters

- Parameters are used to transmit information to a method, typically from another object.
    - Go out of existence when method body completes unless saved to instance variable.

Match statement

- Multiway if-then-elseif-elseif-else can be hard to read and inefficient if lots of comparisons.
- Match statement can use single value to determine which statements to execute.
- Example:

```
if (colorNumber == 1) then {
    lineColor := colorGen.red
} elseif {colorNumber == 2} then {
    lineColor := colorGen.black
} else {
    lineColor := colorGen.white
}
```

Match statement

- Instead:

```
match(colorNumber)
  case {1 -> lineColor := colorGen.red}
  case {2 -> lineColor := colorGen.white}
  case {3 -> lineColor := colorGen.blue}
```

- Only executes case corresponding to number

Questions?