Use parameters in methods

COMP 102

Victoria University of Wellington

Even better design: parameters

- Every time we want a lollipop of a different size or in a different position, we have to modify the code.
- How come we don't have to do that with fillOval?
- fillOval has four parameters:

Definition of fillOval:

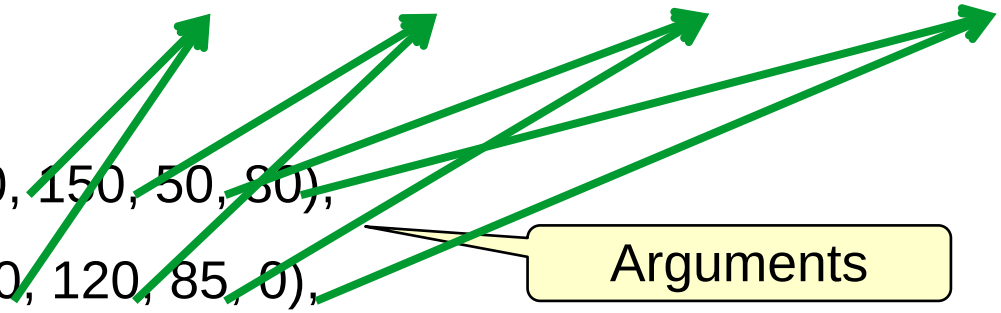
```
public void fillOval(double left, double top, double wd, double ht) {……}
```

Parameters

In the library files

Calling fillOval:

```
UI.fillOval(200, 150, 50, 80);
UI. fillOval(400, 120, 85, 0);
```



Arguments

In our program

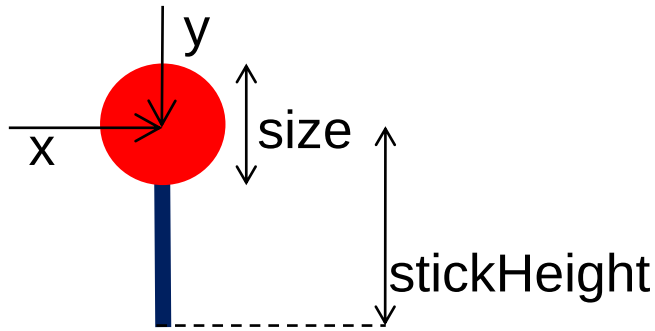
⇒ fillOval can make many different ovals.

Why can't we do that with lollipop?

Improving the program: using parameters

```
/** Draw a lollipop at  (300, 180), asking the user for its size */
public void doDrawLollipop() {
    double size = UI.askDouble("Diameter:");
    double stickHeight =  UI.askDouble("Stick height");
    this.drawLollipop(300, 180, size, stickHeight);
}

public void drawLollipop(double x, double y, double size, double stick) {
    double left = x – size/2.0;          // left of lollipop
    double top = y – size/2.0;           // top of lollipop
    double bot =  y + stick;             // bottom of stick
    UI.setLineWidth(size/8);
    UI.drawLine(x, y, x, bot);
    UI.setLineWidth(1);
    UI.setColor(Color.red);
    UI.fillOval(left, top, size, size);
}
```



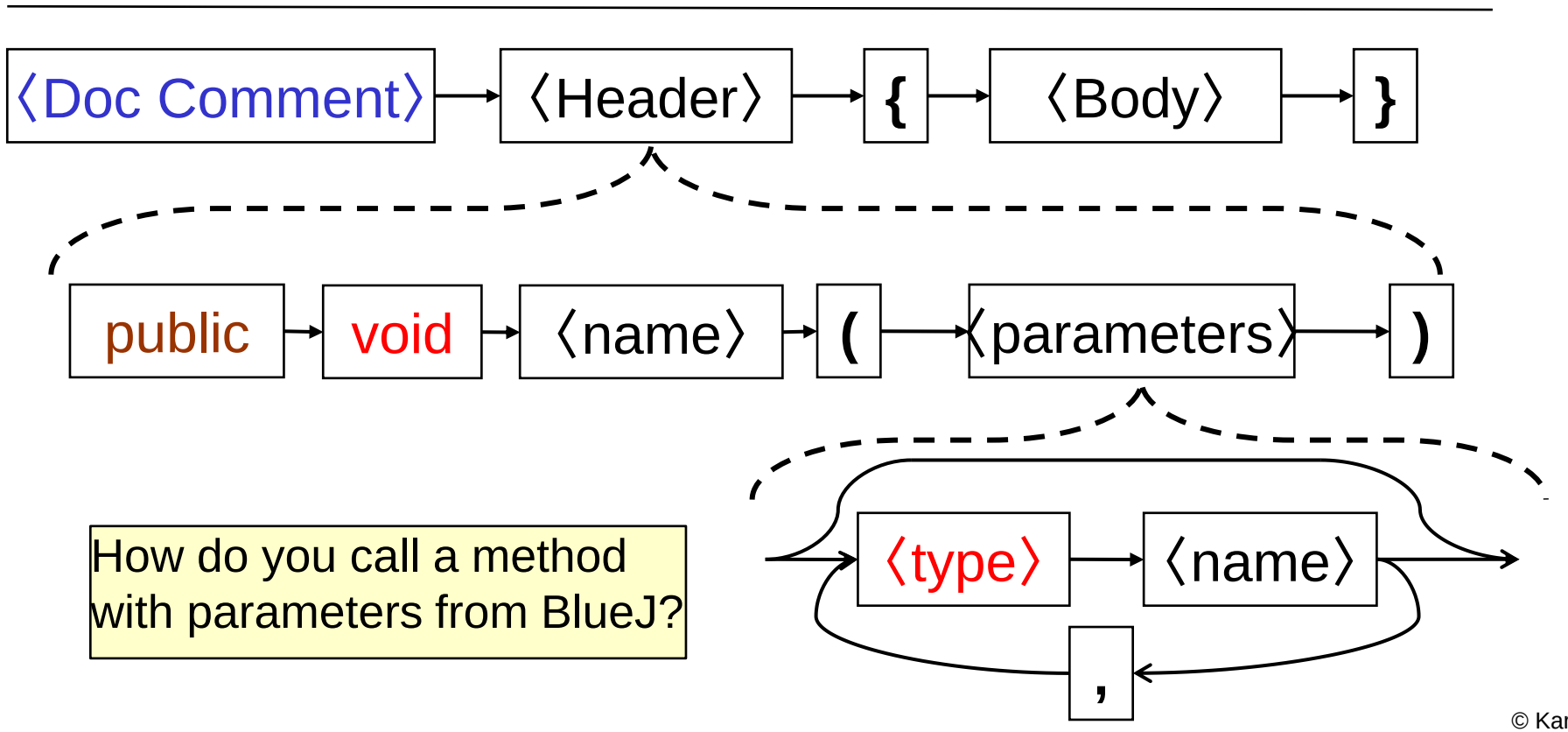
Parameters

Special variables which are given values each time the method is called.

Body of method can use the values in the parameters

Syntax: Method Definitions (v2)

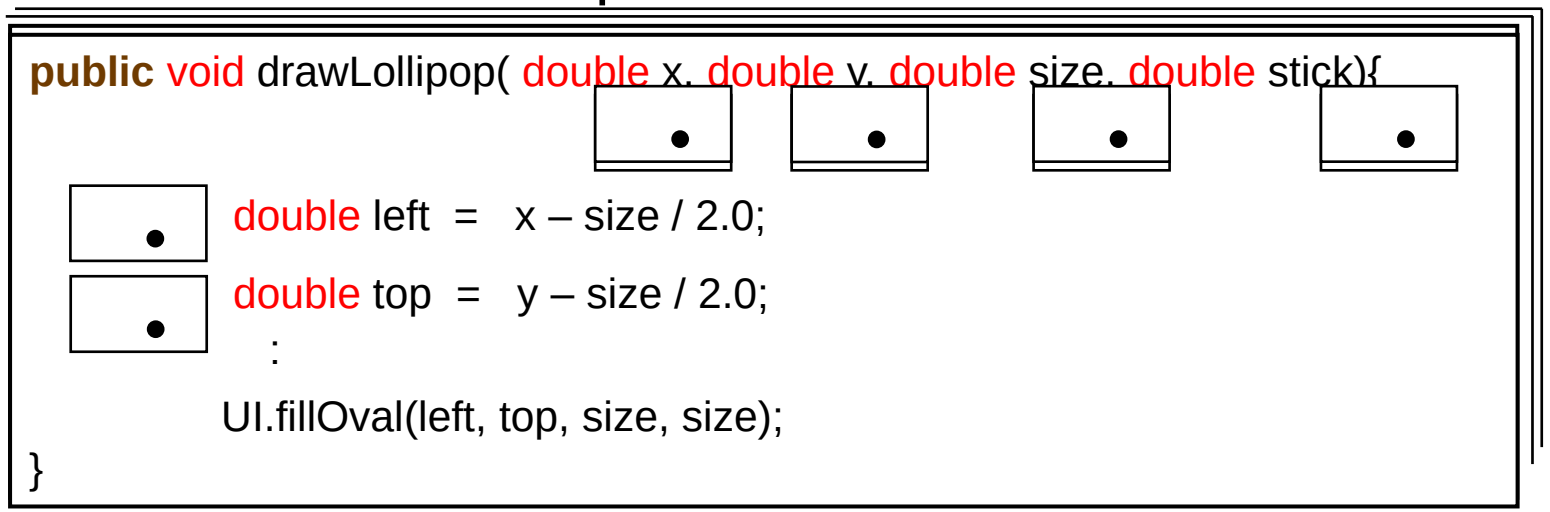
```
/** Draw a lollipop on a stick */
public void drawLollipop(double x, double y, double size, double stick ){
    double left = x – size/ 2.0;
            :
```



How do you call a method with parameters from BlueJ?

Method Calls with parameters

Method Definition: Like a pad of worksheets



Calling a Method:

this.drawLollipop(300, 100, 75, 95);

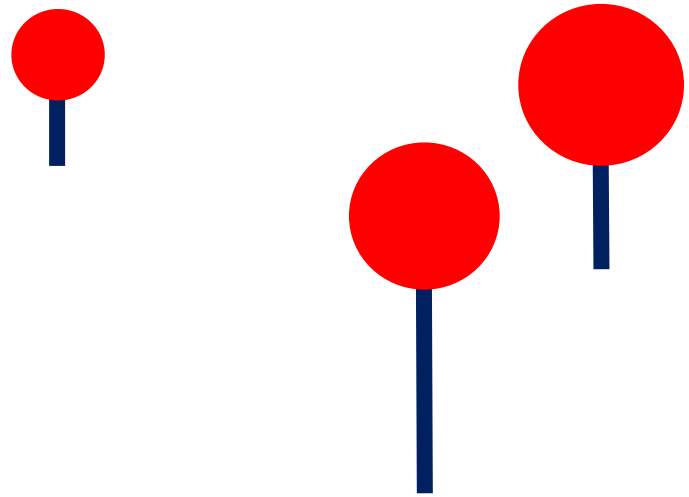
⇒ get a “copy” of the method worksheet

⇒ copy the arguments to the parameter places

⇒ perform each action in the body

⇒ throw the worksheet away (losing all the information on it)

Calling drawLollipop



```
public class  Drawer {
    public void drawThreeLollipops() {
        double diam = UI.askDouble("diameter:");
        this.drawLollipop(300, 180, diam, 200);
        this.drawLollipop(50, 60, diam/2.0, 90);
        this.drawLollipop(400, 100, diam, 70);
    }
    /** Draw a lollipop */
    public void drawLollipop(double x, double y, double size, double stick) {
        double left = x – size/2.0;          // left of lollipop
        double top = y – size/2.0;           // top of lollipop
        double bot =  y + stick;             // bottom of stick
        UI.setLineWidth(size/8);
        UI.drawLine(x, y, x, bot);
        UI.setLineWidth(1);
        UI.setColor(Color.red);
        UI.fillOval(left, top, size, size);
    }
```

Principle of good design

- Parameterising a method makes it more flexible and general
  - Allows us to call the same method with different arguments to do the same thing in different ways
  - Allows us to reuse the same bit of code