Introduction to Buggle World

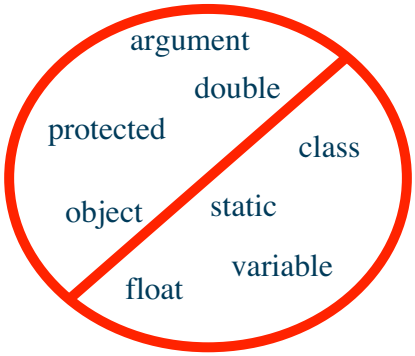
Reminders...

- FirstClass course conferences are your friend!
- Reading assignments are on the web site.

Outline

- Buggles and BuggleWorld
- Objects and Classes
- Methods and Variables
- Contracts
- Expressions and Statements

Unlearn what you have learned

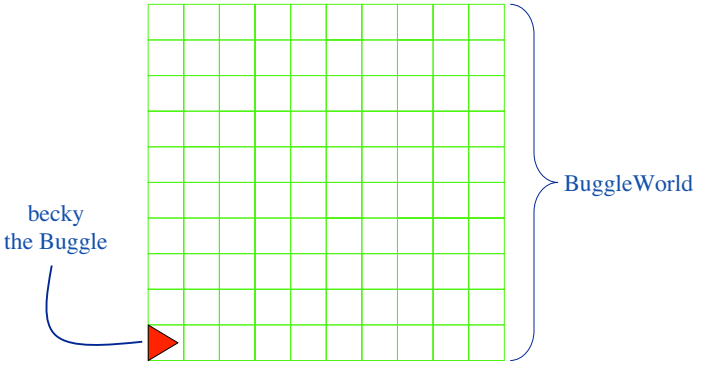


"I don't think that word means what you think it means"

The Buggle

- Buggles are represented on the computer screen as triangles; BuggleWorld is represented as a grid of squares
- What can Buggles do?
  - Buggles can move around in BuggleWorld
  - Buggles can paint in BuggleWorld
  - Buggles can change direction
  - Buggles can drop and eat bagels
- Buggles and BuggleWorld are not part of the Java language, they are part of a program written by Lyn Turbak

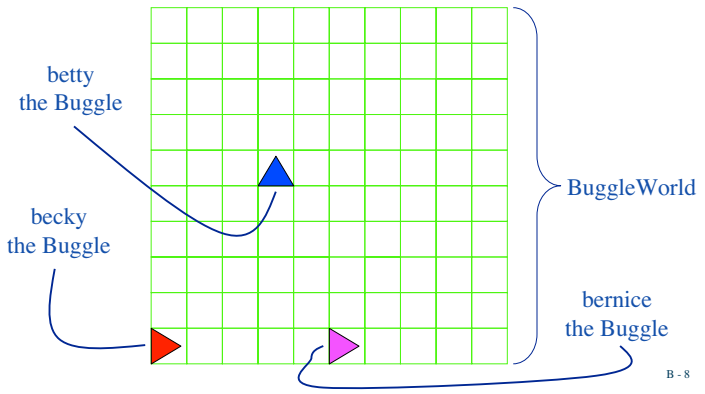
BuggleWorld on the Screen



Buggle properties

- Buggles have four properties...
  - position: The location of the Buggle in BuggleWorld - specified by an (X,Y) coordinate
  - heading: The compass direction the Buggle is facing (NORTH, SOUTH, EAST, or WEST)
  - color: The color of the Buggle (and its painted trail)
  - brushDown: Whether the Buggle's brush is down (the Buggle is leaving a trail behind it)

More Buggles



Buggles as Objects

- Now we have 3 Buggles. Buggles are *objects*.
- We say that these 3 Buggles (objects) belong to the Buggle category - or rather, these 3 objects belong to the Buggle *class*.
- A class can be thought of as a template, or a mold, for creating an object.
- Every object is a member of a class
- Objects are also called instances - each object is a specific instance of its general class

What is a Class?

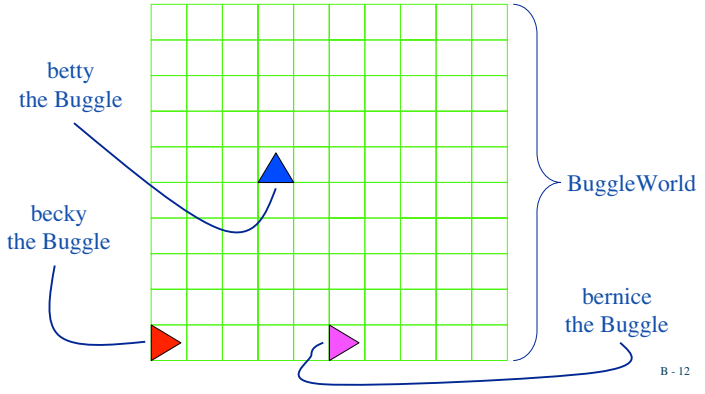
A class is described by:

- instance variables - Instance variables describe the properties of each class instance (each object of the class).

Buggle Objects

- In the Buggle class, every Buggle object has 4 properties (instance variables)
- So becky, betty, and bernice each have the same 4 properties, but they may have different states, i.e., the value of the properties may differ for each of our Buggles
  - becky: (1,1); EAST; red; true
  - betty: (4,6); NORTH; blue; true
  - bernice: (6,1); EAST; magenta; false

More Buggles



Changing Instance Variables

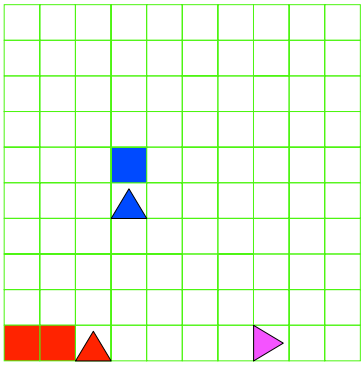
- How do we change the state of an object (such as a Buggle)? In other words, how do we change the value of an instance variable?

  - We send the object a message

- Messages you can send Buggles include `forward`, `backward`, `left`, and `right`

```
becky.forward();
becky.left();
```


Go, becky, Go!



```
becky.forward();
becky.forward();
becky.left();

betty.backward();

bernice.forward();
bernice.forward();
```


What is a Class?

A class is described by:

- instance variables - Instance variables desribe the properties of each class instance (each object of the class).

What is a Class?

A class is described by:

- instance variables - Instance variables desribe the properties of each class instance (each object of the class).
- instance methods - Instance methods are the messages an instance of the class can respond to.

Instance Methods

```
becky.forward();
becky.left();
```

The statements above send messages to becky. When an object receives a message, it executes a set of instructions called a method.

The general form of invoking a method is

```
object.method();
```


Methods with Arguments (aren't we sassy?)

- Some methods require additional information when they are invoked
- The additional information you provide is called an *argument*

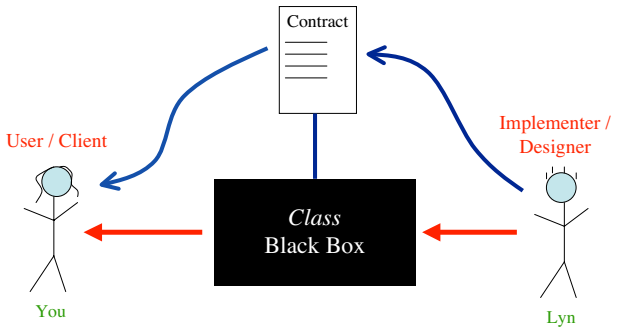
Examples:

```
becky.setColor(Color.yellow);
becky.forward(5);
becky.backward(3);
```

Contracts

- Every class has a *contract* that specifies the behavior of its methods, i.e., how instances of the class respond to messages
- Any user of a class can expect that objects will behave as described
- Any implementer of the class must ensure that objects fulfill the contract
- Another term for a contract is an Application Programming Interface (API). An API is the documentation that allows programmers to use class instances as black-box abstractions

A Class as a Black Box



It's *deja vu* all over again

- Remember *abstraction* and *interfaces*?
- Recall: Once you learn how to drive, you can drive any car - they all have common features (a common interface such as steering, gas, and breaking). Once you learn these, you can use any car.

Let's review...

- becky is an object. What kind of object? A Buggle. She is a member of the Buggle class.
- She (as a Buggle) has certain properties (instance variables): `position`, `heading`, `color`, and `brushDown`.
- To change her properties (instance variables), we send her a message (we invoke a method)...

```
becky.backward(7);
```

- The set of messages we can send her is specified in the contract.

What else is in the Buggle contract?

Constructor

- Every class has a special method (or methods) called a *constructor*
- A constructor helps create new instances of the class
- The constructor is invoked by using the *new* operator and the name of the class

```
Buggle becky = new Buggle();
```

Void vs. Fruitful Methods

- When an object receives a message, it executes a set of instructions called a method
- Some methods do NOT produce a value - these are called void methods
- Some methods produce a value - these are called fruitful methods

```
becky.getColor();
becky.isOverBagel();
```


FirstBuggleExample

```
import java.awt.*;

public class FirstBuggleExample extends BuggleWorld {

    public void init() {
      super.init();
      setDimensions(9,9);
    }

    public void run () {
      // Create new Buggle and move her around
      Buggle becky = new Buggle();
      becky.forward();
      becky.forward();
      becky.left();
    }
}
```