

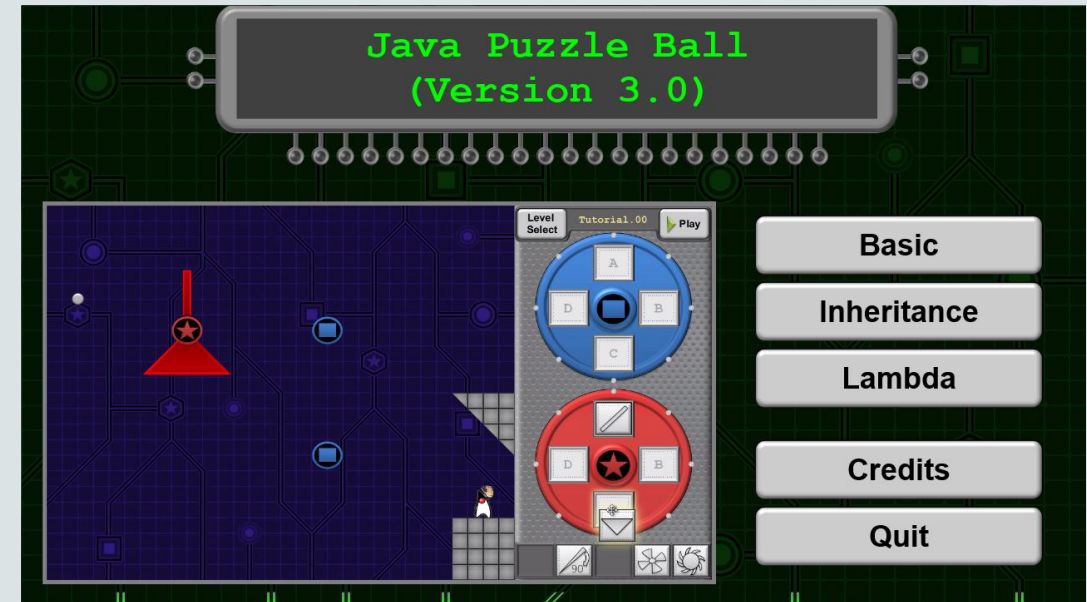


Java Puzzle Ball

Nick Ristuccia

Lesson 2-2


Static vs Instance Variables



One Quick Note

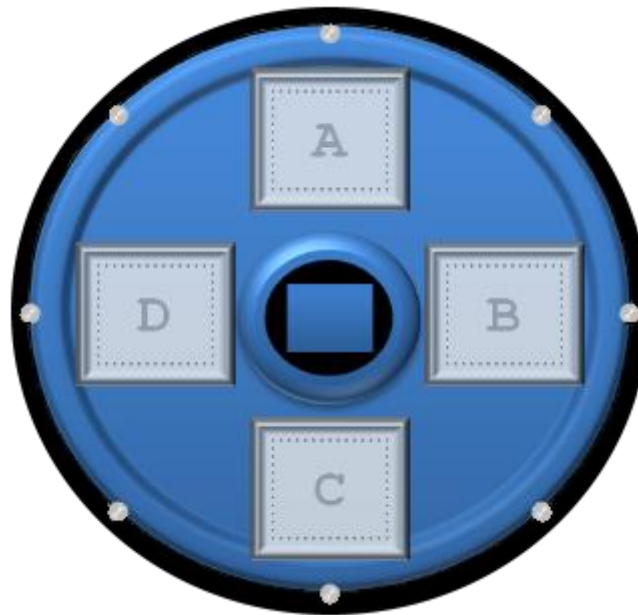
- In this lesson, we'll be using the term **variable**.
- Like a variable in mathematics, a Java variable represents a value.
- Fields utilize variables:
 - In Lab 1, the variable `balance` represents the amount of money in an account.
 - The value of `balance` may change.
 - There are different ways fields can utilize variables. We'll explore this in this lesson.

```
public class SavingsAccount {  
    //Fields  
    private String accountType;  
    private String accountOwner;  
    private double balance;  
    private double interestRate;  
    ...  
}
```



Exercise 2

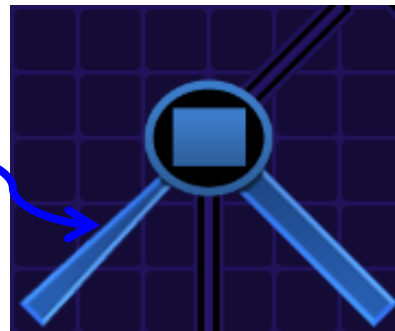
- Play **Basic Puzzles 8 through 11.**
- Consider the following:
 - What happens when you rotate the BlueWheel?
 - How else can you affect the rotation of bumpers?



Java Puzzle Ball Debriefing

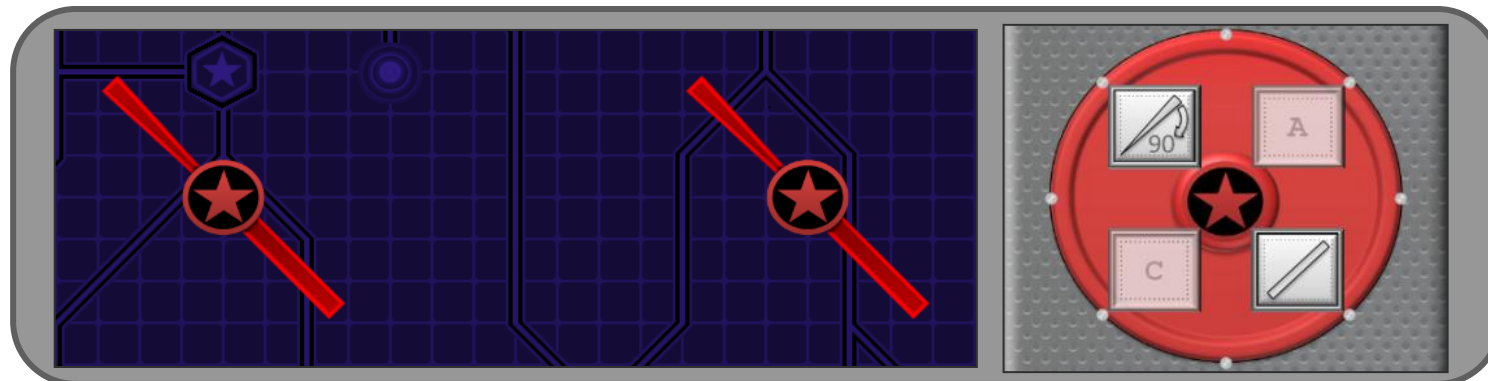
- What happens when you rotate the BlueWheel?
 - The orientation of **all** BlueBumpers change.
 - **All** BlueBumpers share the orientation property.
 - Orientation can be represented by a **static variable**.
- How else can you affect the rotation of bumpers?
 - After the ball strikes a rotation wall, the rotation of an **individual** bumper changes.
 - Rotation can be represented by an **instance variable**.

Rotation wall



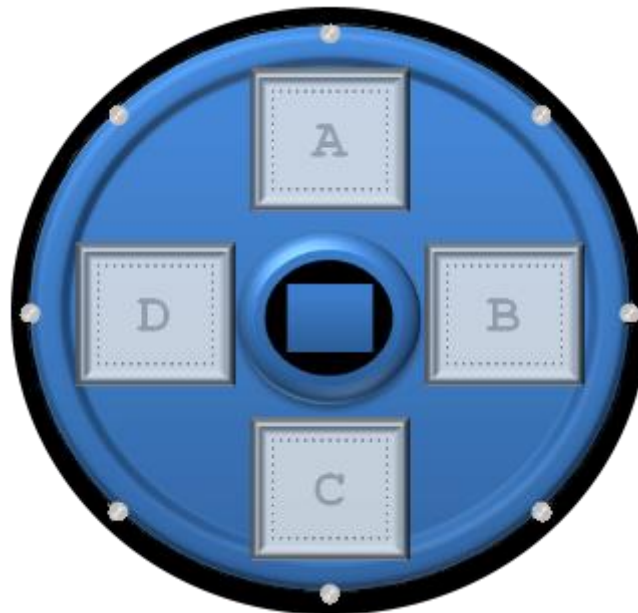
Static Variable: Orientation

- This static variable is shared by all instances.
- Static variables apply to the class, not to any individual instance.
- Therefore, a static variable needs to be changed only once for every instance to be affected.
- In Basic Puzzle 11, rotating the RedWheel changes the orientation of all RedBumper objects.



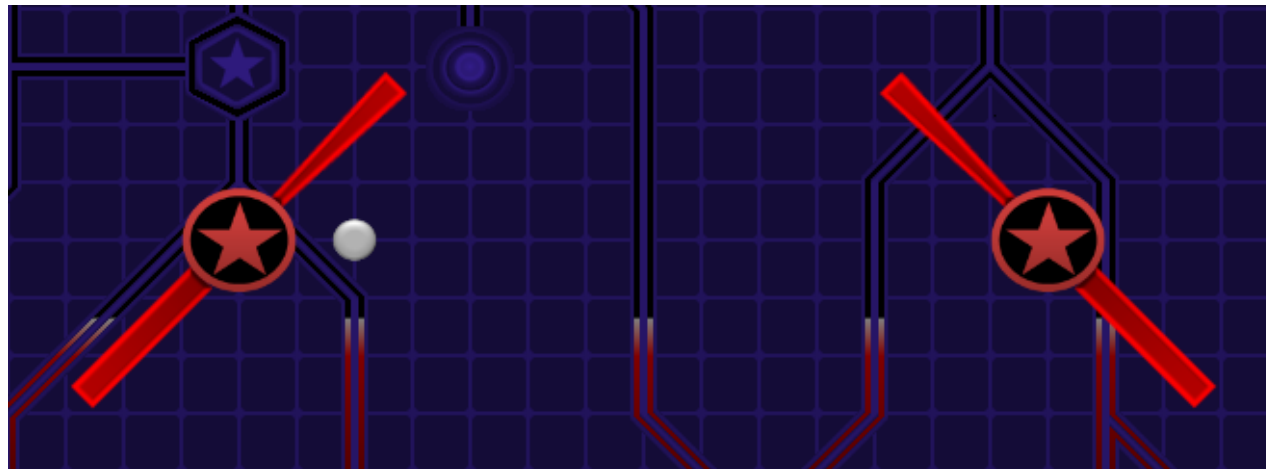
Static Variables with No Instances

- Static variables can be accessed, even if no objects have been instantiated.
- In Basic Puzzle 11, the BlueWheel can be rotated to change the orientation property of all BlueBumpers.
 - There just aren't any BlueBumpers to show the effects of this change.



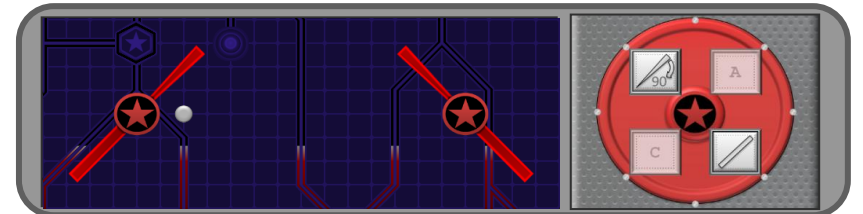
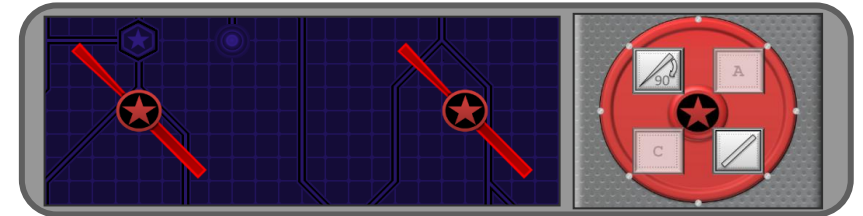
Instance Variables: Rotation

- Unique instance variables exist for every instance of an object.
- Therefore, instance variables need to be changed for each individual object.
- In Basic Puzzle 11, an individual RedBumper's rotation changes after being struck by the ball.



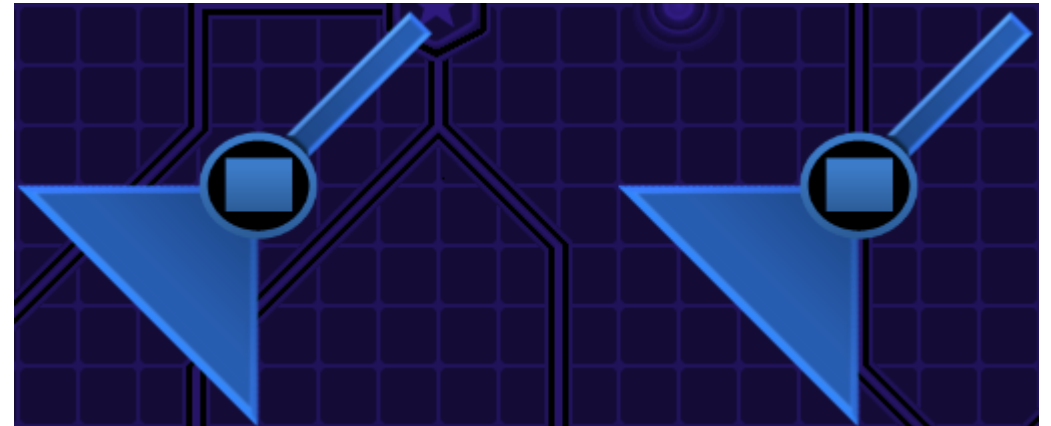
Static Vs Instance

- A Static Variable...
 - Applies to the entire class.
 - Exists once.
 - Needs to be changed once to affect all instance.
 - Example: The orientation of all Bumpers
- An Instance Variables...
 - Applies to a particular instance.
 - Exists for every instance.
 - Is changeable one-at-a-time for every instance.
 - Example: The additional rotation applied to an individual Bumper.



Static or Instance?

- Which BlueBumper fields could be represented by static variables? Which could be represented by instance variables?
 - Color (???)
 - Shape (???)
 - Orientation (static)
 - Rotation (instance)
 - x-position (???)
 - y-position (???)



Translating into Java Syntax

- A variable is used
- To make a field static, simply include the `static` keyword when the variable is declared.

```
1 public class BlueBumper {  
2     private static Color color = Color.BLUE;  
3     private static Shape shape = Shape.RECT;  
4     private static double orientation = 0;  
5     private double rotation;  
4     private int xPosition;  
5     private int yPosition;  
6  
7     ...  
}
```

