

---

# CSE 331

## Software Design & Implementation

Winter 2021

Section 3 – HW4, Abstract Data Types, and JUnit

---

# Agenda

---

- Overview of HW4
- Quick review of polynomial arithmetic
- Review abstract data types (ADTs) by example
- Unit testing with JUnit – an initial tour for HW4

# HW4 – Polynomial calculator

---

A homework in 6 parts:

0. Pseudocode algorithms for polynomial arithmetic
1. Conceptual questions about `RatNum`
2. Implement `RatTerm`
3. Implement `RatPoly`
4. Implement `RatPolyStack`
5. Try out your finished calculator!
6. Run your code against our tests to make sure it works!



# The RatThings

---

- **RatNum** ADT
  - A rational number
  - Also includes a NaN (“not a number”) value
- **RatTerm** ADT
  - A polynomial term (rational coefficient w/ integer degree)
- **RatPoly** ADT
  - A polynomial expression (sum of polynomial terms)
- **RatPolyStack** ADT
  - An ordered collection of polynomial expressions



# The RatThings

---

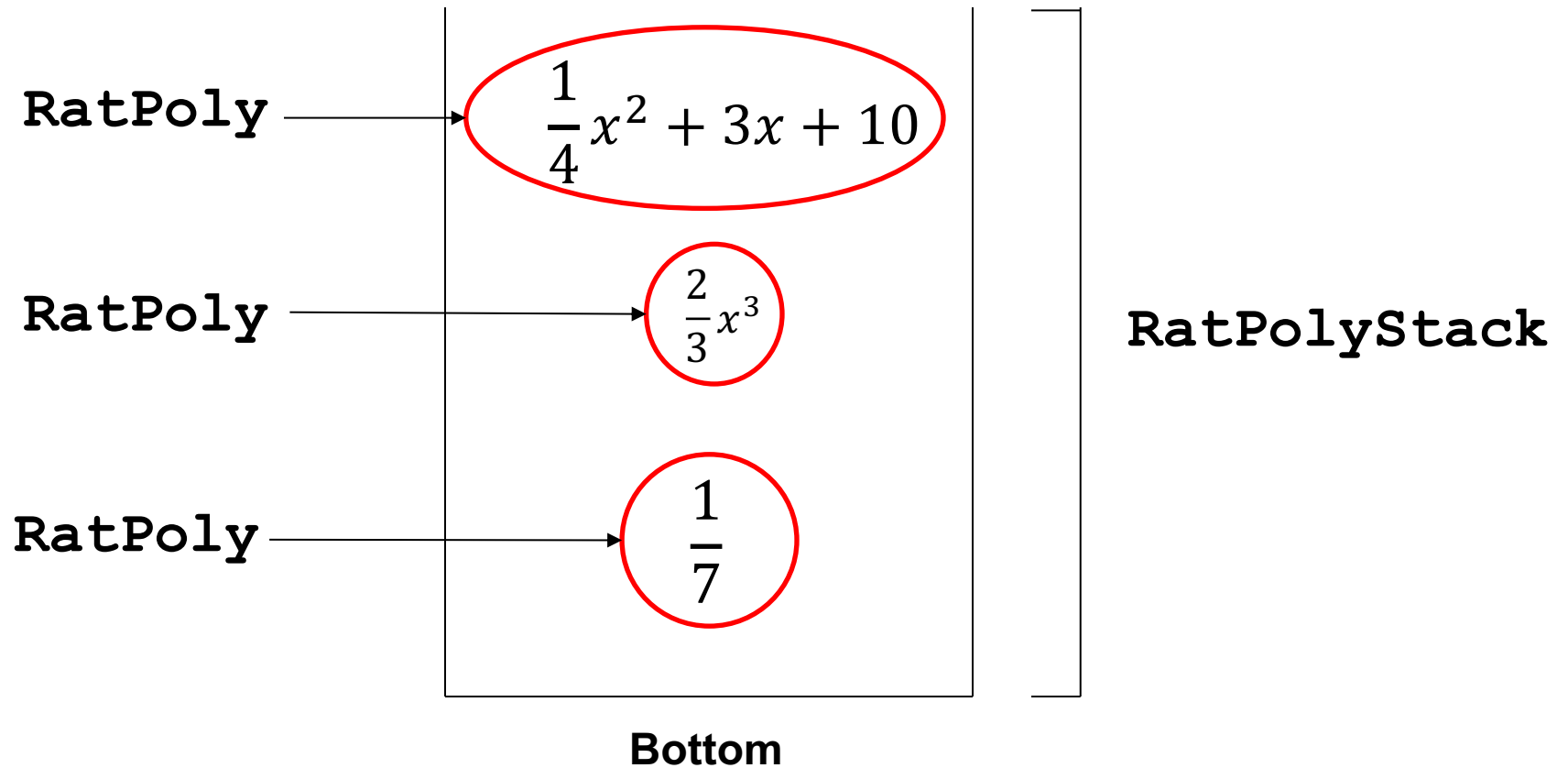
**RatPoly** →  $\frac{1}{4}x^2 + 3x + 10$

**RatTerm** →  $\frac{2}{3}x^3$

**RatNum** →  $\frac{1}{7}$

# The RatThings

---



# Polynomial arithmetic

---

Review arithmetic operations over polynomial expressions:

1. Addition
2. Subtraction
3. Multiplication
4. Division

# Polynomial addition

---

$$(5x^4 + 4x^3 - x^2 + 5) + (3x^5 - 2x^3 + x - 5)$$

# Polynomial addition

---

$$(5x^4 + 4x^3 - x^2 + 5) + (3x^5 - 2x^3 + x - 5)$$

$$\begin{array}{r} 5x^4 + 4x^3 - 1x^2 + 5 \\ + \quad 3x^5 - 2x^3 + 1x - 5 \\ \hline \end{array}$$

# Polynomial addition

---

$$(5x^4 + 4x^3 - x^2 + 5) + (3x^5 - 2x^3 + x - 5)$$

$$\begin{array}{r} 0x^5 + 5x^4 + 4x^3 - 1x^2 + 0x + 5 \\ + 3x^5 + 0x^4 - 2x^3 + 0x^2 + 1x - 5 \\ \hline \end{array}$$

# Polynomial addition

---

$$(5x^4 + 4x^3 - x^2 + 5) + (3x^5 - 2x^3 + x - 5)$$

$$\begin{array}{r} 0x^5 + 5x^4 + 4x^3 - 1x^2 + 0x + 5 \\ + 3x^5 + 0x^4 - 2x^3 + 0x^2 + 1x - 5 \\ \hline 3x^5 + 5x^4 + 2x^3 - 1x^2 + 1x + 0 \end{array}$$

# Polynomial subtraction

---

$$(5x^4 + 4x^3 - x^2 + 5) - (3x^5 - 2x^3 + x - 5)$$

# Polynomial subtraction

---

$$(5x^4 + 4x^3 - x^2 + 5) - (3x^5 - 2x^3 + x - 5)$$

$$\begin{array}{r} 5x^4 + 4x^3 - 1x^2 + 5 \\ - 3x^5 - 2x^3 + 1x - 5 \\ \hline \end{array}$$

# Polynomial subtraction

---

$$(5x^4 + 4x^3 - x^2 + 5) - (3x^5 - 2x^3 + x - 5)$$

$$\begin{array}{r} 0x^5 + 5x^4 + 4x^3 - 1x^2 + 0x + 5 \\ - 3x^5 + 0x^4 - 2x^3 + 0x^2 + 1x - 5 \\ \hline \end{array}$$

# Polynomial subtraction

---

$$(5x^4 + 4x^3 - x^2 + 5) - (3x^5 - 2x^3 + x - 5)$$

$$\begin{array}{r} 0x^5 + 5x^4 + 4x^3 - 1x^2 + 0x + 5 \\ - 3x^5 + 0x^4 - 2x^3 + 0x^2 + 1x - 5 \\ \hline -3x^5 + 5x^4 + 6x^3 - 1x^2 - 1x + 10 \end{array}$$

# Polynomial multiplication

---

$$(4x^3 - x^2 + 5) \times (x - 5)$$

# Polynomial multiplication

---

$$(4x^3 - x^2 + 5) \times (x - 5)$$

$$\begin{array}{r} 4x^3 - x^2 + 5 \\ \times \phantom{4x^3 - x^2} 1x - 5 \\ \hline \end{array}$$

# Polynomial multiplication

---

$$(4x^3 - x^2 + 5) \times (x - 5)$$

$$\begin{array}{r} \phantom{\times} \phantom{4x^3} - \phantom{1x^2} \phantom{+} \phantom{5} \\ \times \phantom{4x^3} - \phantom{1x^2} \phantom{+} \phantom{5} \\ \hline \phantom{\times} -20x^3 + 5x^2 \phantom{+} \phantom{5} \\ \phantom{\times} \phantom{4x^3} - \phantom{1x^2} \phantom{+} \phantom{5} \end{array}$$

# Polynomial multiplication

---

$$(4x^3 - x^2 + 5) \times (x - 5)$$

$$\begin{array}{r} \phantom{\times} \phantom{4x^3} - \phantom{1x^2} \phantom{+} \phantom{5} \\ \times \phantom{4x^3} - \phantom{1x^2} \phantom{+} \phantom{5} \\ \hline \phantom{4x^4} - 20x^3 + 5x^2 \phantom{+} \phantom{5x} - 25 \\ 4x^4 - 1x^3 \phantom{+} \phantom{5x} \phantom{-} \phantom{25} \\ \hline \phantom{4x^4} - 20x^3 + 5x^2 + 5x - 25 \end{array}$$



# Polynomial division

---

$$(5x^6 + 4x^4 - x^3 + 5) / (x^3 - 2x - 5)$$

# Polynomial division

---

$$(5x^6 + 4x^4 - x^3 + 5) / (x^3 - 2x - 5)$$

$$\begin{array}{r|l} 1x^3 & 5x^6 \\ -2x & +4x^4 \\ -5 & -1x^3 \\ \hline & +5 \end{array}$$

# Polynomial division

---

$$(5x^6 + 4x^4 - x^3 + 5) / (x^3 - 2x - 5)$$

$$1x^3 + 0x^2 - 2x - 5 \overline{) 5x^6 + 0x^5 + 4x^4 - 1x^3 + 0x^2 + 0x + 5}$$

# Polynomial division

---

$$(5x^6 + 4x^4 - x^3 + 5) / (x^3 - 2x - 5)$$

$$1x^3 + 0x^2 - 2x - 5 \overline{) 5x^6 + 0x^5 + 4x^4 - 1x^3 + 0x^2 + 0x + 5}$$

$5x^3$

# Polynomial division

---

$$(5x^6 + 4x^4 - x^3 + 5) / (x^3 - 2x - 5)$$

$$\begin{array}{r|rrrrrrr} & & & & 5x^3 & & & \\ 1x^3 + 0x^2 - 2x - 5 & 5x^6 & +0x^5 & +4x^4 & -1x^3 & +0x^2 & +0x & +5 \\ & 5x^6 & +0x^5 & -10x^4 & -25x^3 & & & \end{array}$$







# Polynomial division

---

$$(5x^6 + 4x^4 - x^3 + 5) / (x^3 - 2x - 5)$$

				$5x^3$	$+0x^2$		
$1x^3 + 0x^2 - 2x - 5$	$5x^6$	$+0x^5$	$+4x^4$	$-1x^3$	$+0x^2$	$+0x$	$+5$
	$- 5x^6$	$+0x^5$	$-10x^4$	$-25x^3$			
	$0x^6$	$+0x^5$	$+14x^4$	$+24x^3$	$+0x^2$		

# Polynomial division

---

$$(5x^6 + 4x^4 - x^3 + 5) / (x^3 - 2x - 5)$$

				$5x^3$	$+0x^2$	
$1x^3 + 0x^2 - 2x - 5$	$5x^6$	$+0x^5$	$+4x^4$	$-1x^3$	$+0x^2$	$+0x$
	$- 5x^6$	$+0x^5$	$-10x^4$	$-25x^3$		
	$0x^6$	$+0x^5$	$+14x^4$	$+24x^3$	$+0x^2$	$+0x$
						$+5$

# Polynomial division

---

$$(5x^6 + 4x^4 - x^3 + 5) / (x^3 - 2x - 5)$$

				$5x^3$	$+0x^2$	$+14x$	
$1x^3 + 0x^2 - 2x - 5$	$5x^6$	$+0x^5$	$+4x^4$	$-1x^3$	$+0x^2$	$+0x$	$+5$
	$- 5x^6$	$+0x^5$	$-10x^4$	$-25x^3$			
	$0x^6$	$+0x^5$	$+14x^4$	$+24x^3$	$+0x^2$	$+0x$	

# Polynomial division

---

$$(5x^6 + 4x^4 - x^3 + 5) / (x^3 - 2x - 5)$$

				$5x^3$	$+0x^2$	$+14x$	
$1x^3 + 0x^2 - 2x - 5$	$5x^6$	$+0x^5$	$+4x^4$	$-1x^3$	$+0x^2$	$+0x$	$+5$
	$- 5x^6$	$+0x^5$	$-10x^4$	$-25x^3$			
	$0x^6$	$+0x^5$	$+14x^4$	$+24x^3$	$+0x^2$	$+0x$	
			$14x^4$	$+0x^3$	$-28x^2$	$-70x$	

# Polynomial division

---

$$(5x^6 + 4x^4 - x^3 + 5) / (x^3 - 2x - 5)$$

				$5x^3$	$+0x^2$	$+14x$	
$1x^3 + 0x^2 - 2x - 5$	$5x^6$	$+0x^5$	$+4x^4$	$-1x^3$	$+0x^2$	$+0x$	$+5$
	$- 5x^6$	$+0x^5$	$-10x^4$	$-25x^3$			
	$0x^6$	$+0x^5$	$+14x^4$	$+24x^3$	$+0x^2$	$+0x$	
		$- 14x^4$	$+0x^3$	$-28x^2$	$-70x$		
		$0x^4$	$+24x^3$	$+28x^2$	$+70x$		

# Polynomial division

---

$$(5x^6 + 4x^4 - x^3 + 5) / (x^3 - 2x - 5)$$

				$5x^3$	$+0x^2$	$+14x$	
$1x^3 + 0x^2 - 2x - 5$	$5x^6$	$+0x^5$	$+4x^4$	$-1x^3$	$+0x^2$	$+0x$	$+5$
	$- 5x^6$	$+0x^5$	$-10x^4$	$-25x^3$			
	$0x^6$	$+0x^5$	$+14x^4$	$+24x^3$	$+0x^2$	$+0x$	
			$- 14x^4$	$+0x^3$	$-28x^2$	$-70x$	
			$0x^4$	$+24x^3$	$+28x^2$	$+70x$	

# Polynomial division

---

$$(5x^6 + 4x^4 - x^3 + 5) / (x^3 - 2x - 5)$$

				$5x^3$	$+0x^2$	$+14x$	
$1x^3 + 0x^2 - 2x - 5$	$5x^6$	$+0x^5$	$+4x^4$	$-1x^3$	$+0x^2$	$+0x$	$+5$
	$- 5x^6$	$+0x^5$	$-10x^4$	$-25x^3$			
	$0x^6$	$+0x^5$	$+14x^4$	$+24x^3$	$+0x^2$	$+0x$	
			$- 14x^4$	$+0x^3$	$-28x^2$	$-70x$	
			$0x^4$	$+24x^3$	$+28x^2$	$+70x$	$+5$

# Polynomial division

---

$$(5x^6 + 4x^4 - x^3 + 5) / (x^3 - 2x - 5)$$

				$5x^3$	$+0x^2$	$+14x$	$+24$		
$1x^3 + 0x^2 - 2x - 5$	$\overline{)$	$5x^6$	$+0x^5$	$+4x^4$	$-1x^3$	$+0x^2$	$+0x$	$+5$	
	$-$	$5x^6$	$+0x^5$	$-10x^4$	$-25x^3$				
		$0x^6$	$+0x^5$	$+14x^4$	$+24x^3$	$+0x^2$	$+0x$		
				$-$	$14x^4$	$+0x^3$	$-28x^2$	$-70x$	
					$0x^4$	$+24x^3$	$+28x^2$	$+70x$	$+5$

# Polynomial division

---

$$(5x^6 + 4x^4 - x^3 + 5) / (x^3 - 2x - 5)$$

				$5x^3$	$+0x^2$	$+14x$	$+24$		
$1x^3 + 0x^2 - 2x - 5$	$\overline{)$	$5x^6$	$+0x^5$	$+4x^4$	$-1x^3$	$+0x^2$	$+0x$	$+5$	
	$-$	$5x^6$	$+0x^5$	$-10x^4$	$-25x^3$				
		$0x^6$	$+0x^5$	$+14x^4$	$+24x^3$	$+0x^2$	$+0x$		
				$-$	$14x^4$	$+0x^3$	$-28x^2$	$-70x$	
					$0x^4$	$+24x^3$	$+28x^2$	$+70x$	$+5$
					$24x^3$	$+0x^2$	$-48x$	$-120$	

# Polynomial division

---

$$(5x^6 + 4x^4 - x^3 + 5) / (x^3 - 2x - 5)$$

				$5x^3$	$+0x^2$	$+14x$	$+24$
$1x^3 + 0x^2 - 2x - 5$	$5x^6$	$+0x^5$	$+4x^4$	$-1x^3$	$+0x^2$	$+0x$	$+5$
	$-$	$5x^6$	$+0x^5$	$-10x^4$	$-25x^3$		
		$0x^6$	$+0x^5$	$+14x^4$	$+24x^3$	$+0x^2$	$+0x$
			$-$	$14x^4$	$+0x^3$	$-28x^2$	$-70x$
				$0x^4$	$+24x^3$	$+28x^2$	$+70x$
				$-$	$24x^3$	$+0x^2$	$-48x$
					$0x^3$	$+28x^2$	$+118x$
						$+125$	



# Polynomial division

---

$$(5x^6 + 4x^4 - x^3 + 5) / (x^3 - 2x - 5)$$

$$5x^3 + 14x + 24 + \frac{28x^2 + 118x + 125}{x^3 - 2x - 5}$$

# Abstract data types by example

---

Review ADT concepts through two examples:

- A **Line** ADT
- A **Circle** ADT

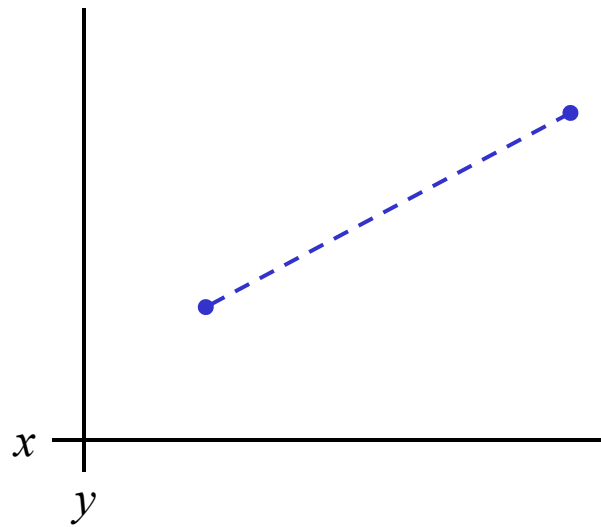
On the course website, see “Resources” → “Class and Method Specifications” for a handy guide with full details.

We won't cover abstraction functions today (see upcoming lecture).

# Line ADT

---

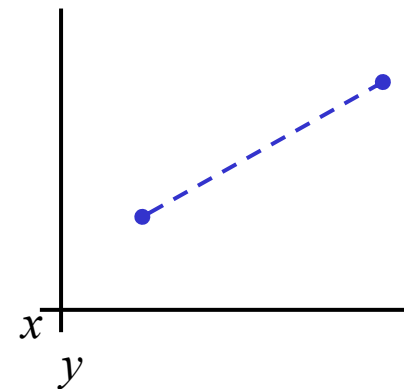
Concept: A line segment in the Cartesian co-ordinate plane



# Line ADT: Class specification

---

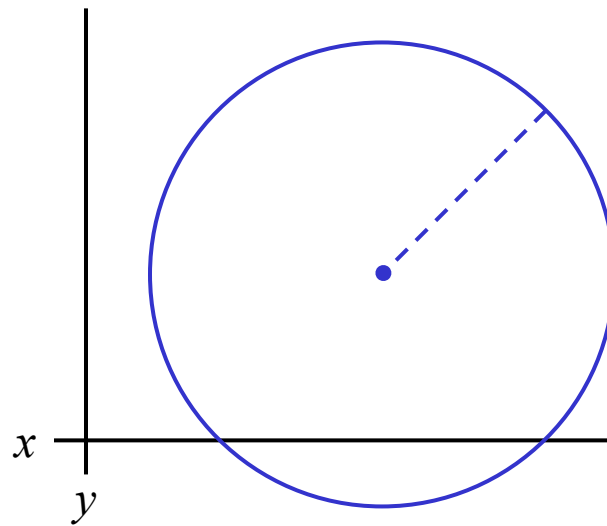
```
/**
 * A Line is a mutable 2D line segment with endpoints
 * p1 and p2.
 */
public class Line {
    ... // rep invariant, fields, methods, etc.
}
```



# Circle ADT

---

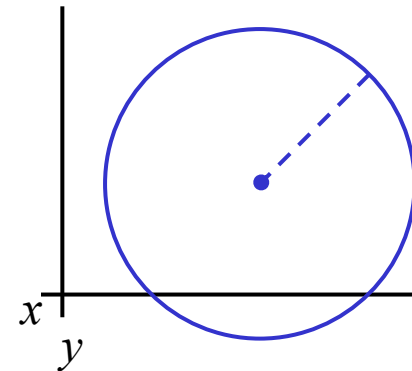
Concept: A circle in the Cartesian co-ordinate plane



# Circle ADT: Class specification

---

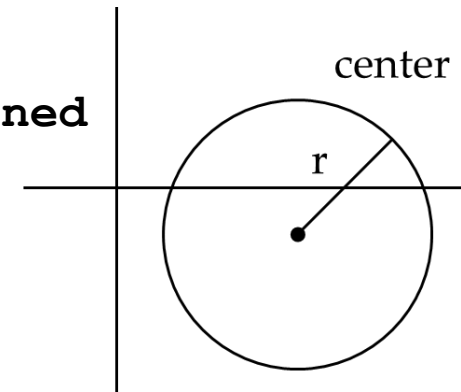
```
/**  
 * A Circle is a mutable 2D circle, defined by a  
 * center point p and radius r.  
 */  
public class Circle {  
    ... // fields, rep invariant, methods, etc.  
}
```



# Circle ADT: Representation #1

---

```
/**  
 * A Circle is a mutable 2D circle, defined  
 * by a center point p and radius r.  
 */  
public class Circle {  
    private Point center;  
    private double radius;  
  
    ...  
}
```



# Interlude: Representation invariant

---

An ADT implementation has a representation invariant:

- Restricts concrete representation of the ADT
- Maps each object's internal state to a boolean for validity

If the representation invariant is violated by (*i.e.*, false for) some object, that object is “broken.”

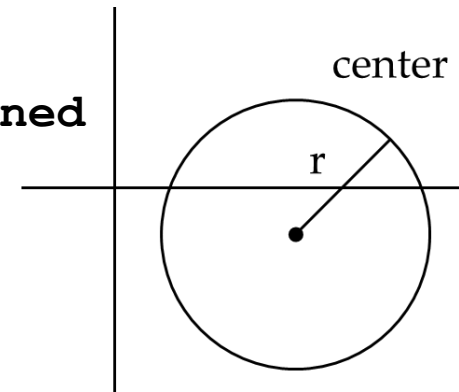
- The object doesn't map to any abstract value
- Indicates a bug in the ADT implementation!

# Circle ADT: Representation #1

---

```
/**
 * A Circle is a mutable 2D circle, defined
 * by a center point p and radius r.
 */
public class Circle {
    private Point center;
    private double radius;

    // Representation Invariant:
    //   center != null && radius > 0
    ...
}
```

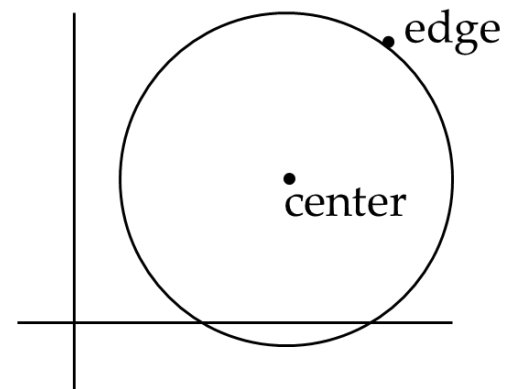


# Circle ADT: Representation #2

---

```
/**
 * A Circle is a mutable 2D circle,
 * defined by a center point p and
 * radius r.
 */
public class Circle {
    private Point center;
    private Point edge;

    // Representation Invariant:
    //   center != null &&
    //   edge != null &&
    //   !center.equals(edge)
}
```

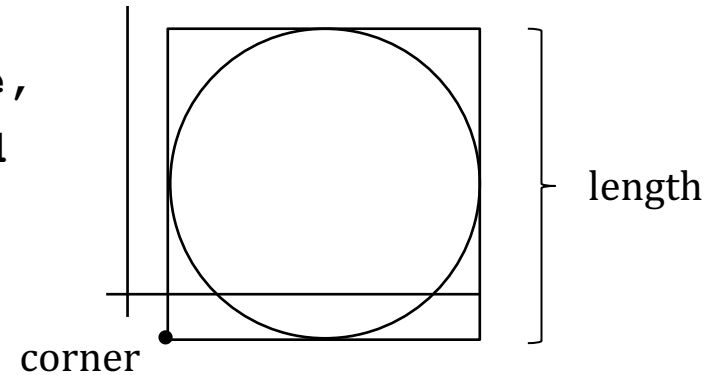


# Circle ADT: Representation #3

---

```
/**  
 * A Circle is a mutable 2D circle,  
 * defined by a center point p and  
 * radius r.  
 */
```

```
public class Circle {  
    private Point corner;  
    private double length;  
  
    // Representation Invariant:  
    //   corner != null &&  
    //   length > 0  
}
```



# Checking the representation invariant

---

The rep. invariant must hold before and after each public method.

Write and use a **checkRep ()** method:

- Call at entry and exit of each public method
  - Only call at the exit of constructors
- Bug-finding value well worth the little extra code
- If slow to check, add code to conditionally do expensive checks when desired and omit when appropriate (more later with hw5, hw6, etc.)

```
public void m(...) {  
    checkRep ();  
    ...  
    checkRep ();  
}
```

## Try it yourself!

---

Write your own specification of a Rectangle ADT on the handout.

Then give two different possible representations for your Rectangle ADT and write checkRep functions for them

# Testing: A quick introduction

---

- For HW 4, you'll be running our test suite to verify your RatThings work.
- Let's do a quick walkthrough of our test suite
  - Just know how it works; don't need to know how to write tests!

# JUnit

---

- Industry-standard Java toolkit for unit testing
  - We're using JUnit 4
- A unit test is a test for one “component” by itself
  - “Component” typically a class or a method
- Each unit test written as a method
  - We'll see the particulars in a moment...
- Closely related unit tests should be grouped into a class
  - For example, all unit tests for the same ADT implementation

# Writing tests with JUnit

---

Annotate a method with `@Test` to flag it as a JUnit test

```
import org.junit.*;
import static org.junit.Assert.*;

/** Unit tests for my Foo ADT implementation */
public class FooTests {
    @Test
    public void testBar() {
        ... /* use JUnit assertions in here */
    }
}
```

# Using JUnit assertions

---

- JUnit assertions establish success or failure of the test method
  - *Note: JUnit assertions are different from Java's `assert` statement*
- Use to check that an actual result matches the expected value
  - Example: `assertEquals(42, meaningOfLife());`
  - Example: `assertTrue(list.isEmpty());`
- A test method stops immediately after the first assertion failure
  - If no assertion fails, then the test method passes
  - Other test methods still run either way
- JUnit results show details of any test failures

# Common JUnit assertions

---

JUnit's documentation has a full list, but these are the most common assertions.

Assertion	Failure condition
<code>assertTrue(<i>test</i>)</code>	<code><i>test</i> == false</code>
<code>assertFalse(<i>test</i>)</code>	<code><i>test</i> == true</code>
<code>assertEquals(<i>expected</i>, <i>actual</i>)</code>	<code><i>expected</i> and <i>actual</i> are not equal</code>
<code>assertSame(<i>expected</i>, <i>actual</i>)</code>	<code><i>expected</i> != <i>actual</i></code>
<code>assertNotSame(<i>expected</i>, <i>actual</i>)</code>	<code><i>expected</i> == <i>actual</i></code>
<code>assertNull(<i>value</i>)</code>	<code><i>value</i> != null</code>
<code>assertNotNull(<i>value</i>)</code>	<code><i>value</i> == null</code>

Any JUnit assertion can also take a string to show in case of failure, e.g., `assertEquals("helpful message", expected, actual)`.

# Checking for a thrown exception

---

- Should test that your code throws exceptions as specified
- This kind of test method fails if its body does *not* throw an exception of the named class
  - May not need any JUnit assertions inside the test method unlike our previous guideline

```
@Test(expected=IndexOutOfBoundsException.class)
public void testGetEmptyList() {
    List<String> list = new ArrayList<String>();
    list.get(0);
}
```

# Test ordering, setup, clean-up

---

JUnit does not promise to run tests in any particular order.

However, JUnit can run helper methods for common setup/cleanup

- Run before/after *each* test method in the class:

```
@Before
```

```
public void m() { ... }
```

```
@After
```

```
public void m() { ... }
```

- Run once before/after running *all* test methods in the class:

```
@BeforeClass
```

```
public static void m() { ... }
```

```
@AfterClass
```

```
public static void m() { ... }
```