CSE 331
Software Design & Implementation

Section 3 – HW4, Abstract Data Types, and JUnit

Administrivia

- HW3 due tonight at 11 PM!
- HW2 due Monday at 5 PM!
- Any questions?

Agenda

- ADTs!
- Overview of HW4
- Quick review of polynomial arithmetic
- Abstraction functions
- Unit testing with Junit – an initial tour for HW4

Abstract Data Types (ADTs)

- Abstraction representing some set of data
  - Meant to express the meaning/concept behind some Java class
- Different from implementation/Java fields!
  - Same ADT can have many different implementations
- Any questions?

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!

Start early, and use your knowledge of invariants to unblock yourself.

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

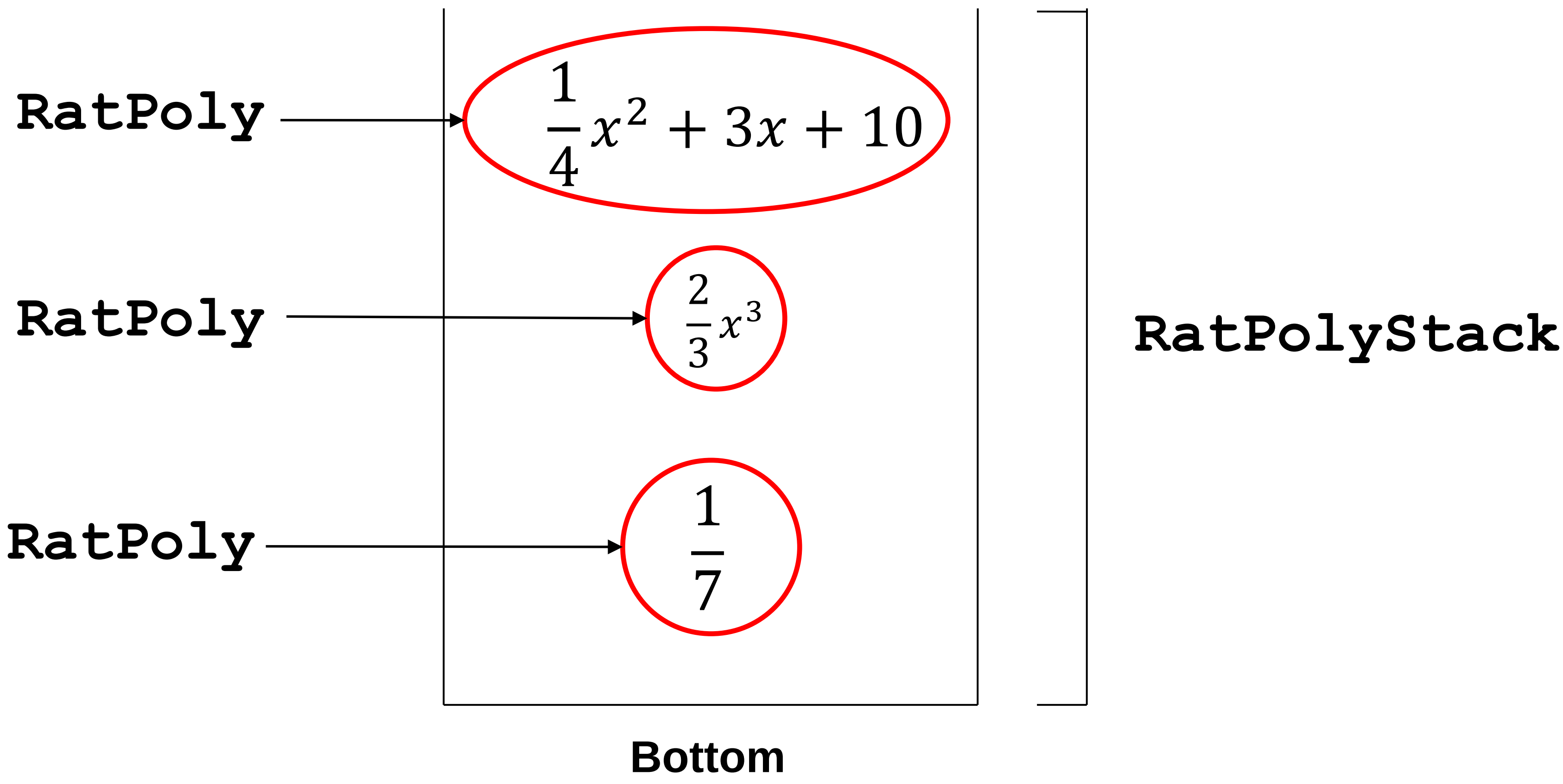
The RatThings

`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

Defining and following invariants is critical to making sure that these operations are implemented correctly.

Polynomial addition

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

Polynomial addition

5)x4 +^{4} x3) + (5 +^{2} x −^{3} x2 −^{5} x(5 − x +^{3}

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

Polynomial addition

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

$$0x^5\ 5\ +x^4\ 4\ +x^3\ 1\ -x^2 + 0x + 5$$

$$+ \quad 3x^5 + 0x^4 - 2x^3 + 0x^2 + 1x - 5$$

Polynomial addition

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

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

Polynomial subtraction

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

Polynomial subtraction

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

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

Polynomial subtraction

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

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

Polynomial subtraction

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

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

Polynomial multiplication

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

Polynomial multiplication

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

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

Polynomial multiplication

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

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

Polynomial multiplication

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

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

Polynomial multiplication

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

$$\begin{array}{rrrrrr}
 & & 4x^3 & -\ 1x^2 & & +\ 5 \\
\times & & & & 1x & -\ 5 \\
\hline
 & & -20x^3 & +\ 5x^2 & & -\ 25 \\
+ & 4x^4 & -\ 1x^3 & & +\ 5x & \\
\hline
 & 4x^4 & -21x^3 & +\ 5x^2 & +\ 5x & -\ 25
\end{array}$$

Polynomial division

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

Polynomial division

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

$$1x^3 \quad -2x \quad -5 \;\Big|\; 5x^6 \quad +4x^4 \quad -1x^3 \quad +5$$

Polynomial division

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

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

Polynomial division

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

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

Polynomial division

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

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

Polynomial division

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

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

Notice (quotient * divisor) + remainder is always equal to $(5x^6 + 4x^4 - x^3 + 5)$.

We can use this fact to produce an invariant.

Polynomial division

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

$$
\begin{array}{r|rrrrrrr}
 & & & & 5x^3 & & & \\
\hline
1x^3 + 0x^2 - 2x - 5 & 5x^6 & +0x^5 & +4x^4 & -1x^3 & +0x^2 & +0x & +5 \\
- & 5x^6 & +0x^5 & -10x^4 & -25x^3 & & & \\
\cline{2-5}
 & 0x^6 & +0x^5 & +14x^4 & +24x^3 & & &
\end{array}
$$

Polynomial division

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

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

Polynomial division

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

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

Polynomial division

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

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

Polynomial division

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

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

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

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

$$
\begin{array}{r|rrrrrrr}
 & & & & 5x^3 & +0x^2 & +14x & \\
\hline
1x^3 +0x^2 -2x -5 & 5x^6 & +0x^5 & +4x^4 & -1x^3 & +0x^2 & +0x & +5 \\
- & 5x^6 & +0x^5 & -10x^4 & -25x^3 & & & \\
\cline{2-5}
 & 0x^6 & +0x^5 & +14x^4 & +24x^3 & +0x^2 & +0x & \\
 & & - & 14x^4 & +0x^3 & -28x^2 & -70x & \\
\cline{4-7}
 & & & 0x^4 & +24x^3 & +28x^2 & +70x & \\
\end{array}
$$

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

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

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

Polynomial division

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

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

Polynomial division

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

				$5x^3$	$+0x^2$	$+14x$	$+24$	
$1x^3 +0x^2 -2x -5$	$5x^6$	$+0x^5$	$+4x^4$	$-1x^3$	$+0x^2$	$+0x$	$+5$	
$-$	$5x^6$	$+0x^5$	$-10x^4$	$-25x^3$				
	$0x^6$	$+0x^5$	$+14x^4$	$+24x^3$	$+0x^2$	$+0x$		
		$-$	$14x^4$	$+0x^3$	$-28x^2$	$-70x$		
			$0x^4$	$+24x^3$	$+28x^2$	$+70x$	$+5$	
			$-$	$24x^3$	$+0x^2$	$-48x$	-120	
				$0x^3$	$+28x^2$	$+118x$	$+125$	

Polynomial division

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

quotient

					$5x^3$	$+0x^2$	$+14x$	$+24$
$1x^3 +0x^2 -2x -5$		$5x^6$	$+0x^5$	$+4x^4$	$-1x^3$	$+0x^2$	$+0x$	$+5$
	$-$	$5x^6$	$+0x^5$	$-10x^4$	$-25x^3$			
		$0x^6$	$+0x^5$	$+14x^4$	$+24x^3$	$+0x^2$	$+0x$	
			$-$	$14x^4$	$+0x^3$	$-28x^2$	$-70x$	
				$0x^4$	$+24x^3$	$+28x^2$	$+70x$	$+5$
				$-$	$24x^3$	$+0x^2$	$-48x$	-120
					$0x^3$	$+28x^2$	$+118x$	$+125$

remainder

Polynomial division

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

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

Notice that the loop invariant, q*y + r = x and 0 <= r where q is the quotient, y is the divisor, r is the remainder and x is the polynomial that is being divided is always correct after each subtraction step.

Abstraction Functions (AFs)

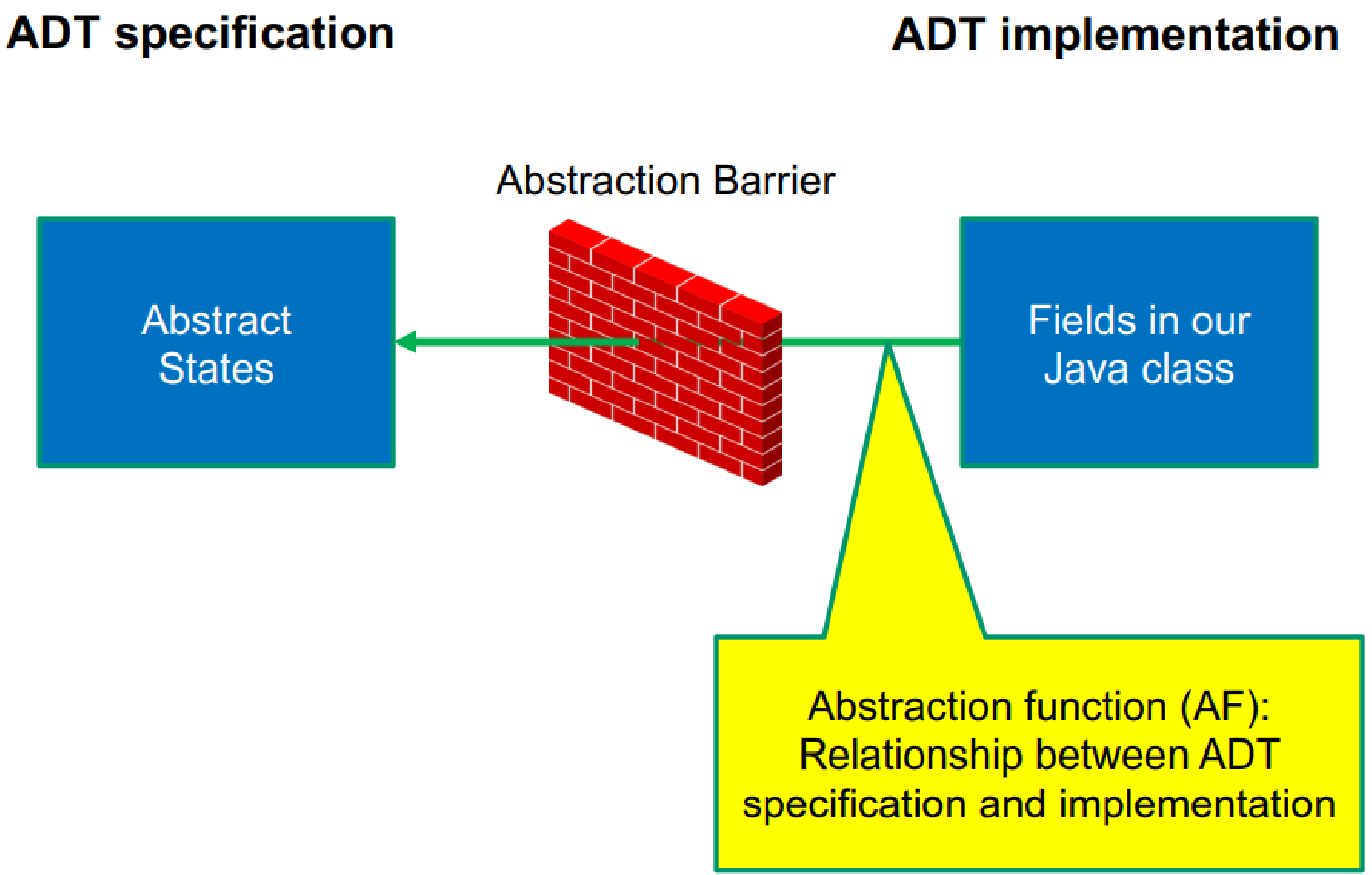
- Let’s say we have an ADT
  - And we choose some way to implement it

- How does the concrete implementation relate to our ADT?

- This is an **abstraction function**
  - Maps object implementation (our Java fields) to the abstract state
  - Ex: “How does a Triangle object from Triangle.java represent a Triangle ADT?”
  - Note: specific to implementation

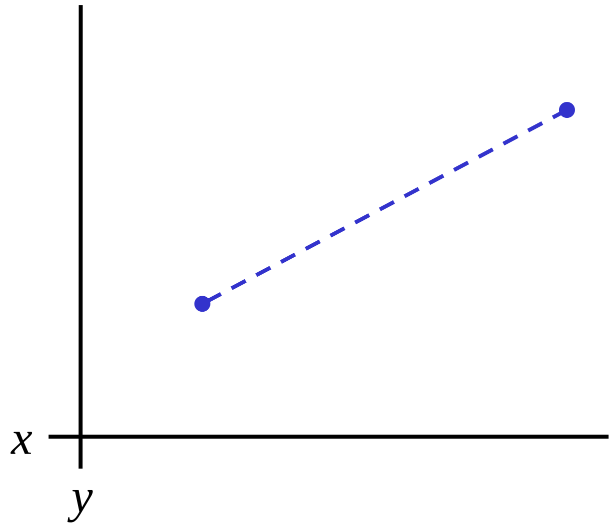
- On the course website, see “Resources” → “Class and Method Specifications” for a handy guide with full details.

Diagram


ADT specification
ADT implementation
Abstraction Barrier
Abstract States
Fields in our Java class
Abstraction function (AF): Relationship between ADT specification and implementation

Line ADT

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

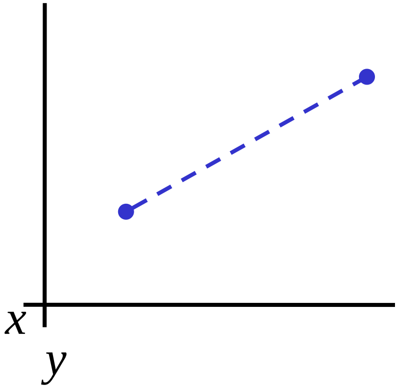


How might we implement this?

Line ADT: Representation #1

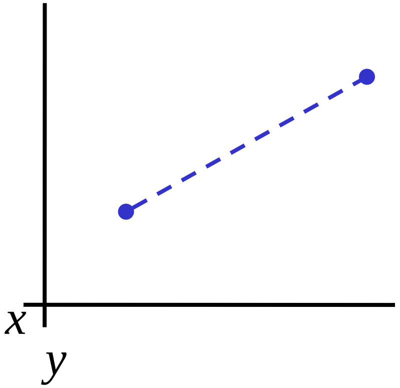
```
/**
 * A Line is a mutable 2D line segment with endpoints
 * p1 and p2.
 */
public class Line {


  private int x1, x2;
  private int y1, y2;
}
```

What is our abstraction function?



Line ADT: Representation #1

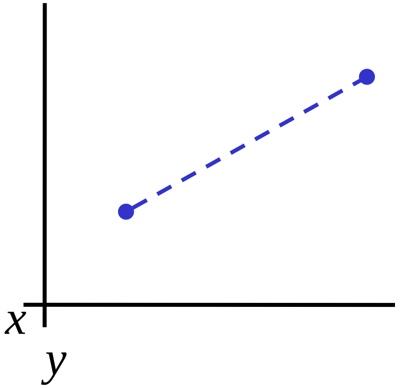
```
/**
 * A Line is a mutable 2D line segment with endpoints
 * p1 and p2.
 */
public class Line {
  // Abstract state is line with endpoints (x1, y1) and
  //                                       (x2, y2)
  private int x1, x2;
  private int y1, y2;
}
```



Line ADT: Representation #2

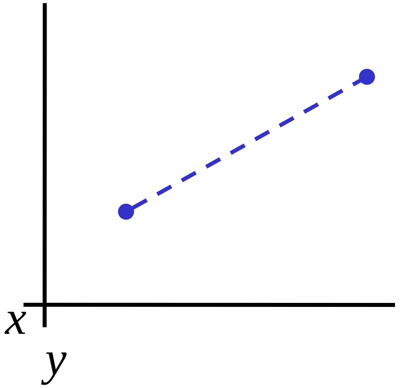
```
/**
 * A Line is a mutable 2D line segment with endpoints
 * p1 and p2.
 */
public class Line {

  private Point pointA, pointB;
}
```

What is our abstraction function?



Line ADT: Representation #2

```
/**
 * A Line is a mutable 2D line segment with endpoints
 * p1 and p2.
 */
public class Line {
  // Abstract state is line with endpoints p1 and p2
  private Point pointA, pointB;
}
```

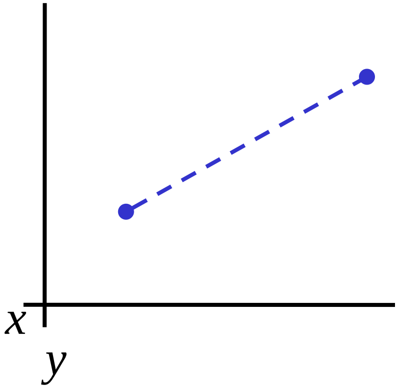


Does this representation have any advantages?

Line ADT: Representation #3

```
/**
 * A Line is a mutable 2D line segment with endpoints
 * p1 and p2.
 */
public class Line {


  private int x1, y1;
  private double angle;
  private double len;
}
```

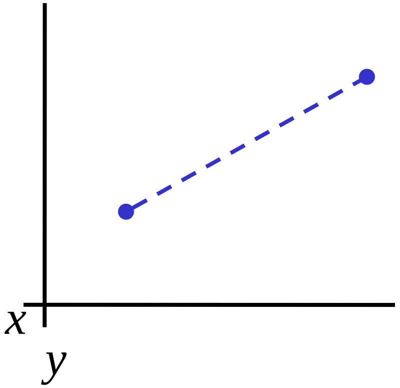
What is our abstraction function?



Line ADT: Representation #3

```
/**
 * A Line is a mutable 2D line segment with endpoints
 * p1 and p2.
 */
public class Line {
  // Abstract state is line with endpoints (x1, y1) and
  //  (x1 + len * cos(angle), y1 + len * sin(angle))
  private int x1, y1;
  private double angle;
  private double len;
}
```

Does this representation have any advantages?



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 abstraction 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 (yet)!

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**
`assertTrue(`*`test`*`)`	*`test`* `== false`
`assertFalse(`*`test`*`)`	*`test`* `== true`
`assertEquals(`*`expected, actual`*`)`	*`expected`* and *`actual`* are not equal
`assertSame(`*`expected, actual`*`)`	*`expected`* `!=` *`actual`*
`assertNotSame(`*`expected, actual`*`)`	*`expected`* `==` *`actual`*
`assertNull(`*`value`*`)`	*`value`* `!= null`
`assertNotNull(`*`value`*`)`	*`value`* `== null`

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() { ... }
```