Class specifications

CSE 331 – Section 3
10/11/12

Slides by Kellen Donohue
with material from Krystal Yousoufian, Mike Ernst

Agenda

- hw3 due tonight
- hw4 out tomorrow
  - Due next Thurs
  - Rational numbers, polynomial, graphing calculator
- Javadoc
- JUnit
- Representation Invariants & Abstraction Functions

hw4

- Polynomial Addition

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

hw4

- Polynomial Addition

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

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

$+ \quad 3x^5 - 2x^3 + x - 5$

hw4

- Polynomial Addition

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

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

hw4

- Polynomial Addition

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

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

hw4

- Polynomial Subtraction

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

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

$- \quad 3x^5 - 2x^3 + x - 5$

hw4

- Polynomial Subtraction

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

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

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

hw4

- Polynomial Subtraction

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

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

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

$$-3x^5 + 5x^4 + 6x^3 - x^2 - x + 10$$

hw4

- Polynomial Multiplication

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

hw4

- Polynomial Multiplication

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

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

*

$x - 5$

hw4

- Polynomial Multiplication

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

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

* $x - 5$

$-20x^3 + 5x^2 \quad - 25$

hw4

- Polynomial Multiplication

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

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

* $x - 5$

$-20x^3 + 5x^2 - 25$

$4x^4 - x^3 + 5x$

hw4

- Polynomial Multiplication

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

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

* $x - 5$

$-20x^3 + 5x^2 - 25$

\+ $4x^4 - x^3 + 5x$

$4x^4 - 21x^3 + 5x^2 + 5x - 25$

Polynomial Division

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

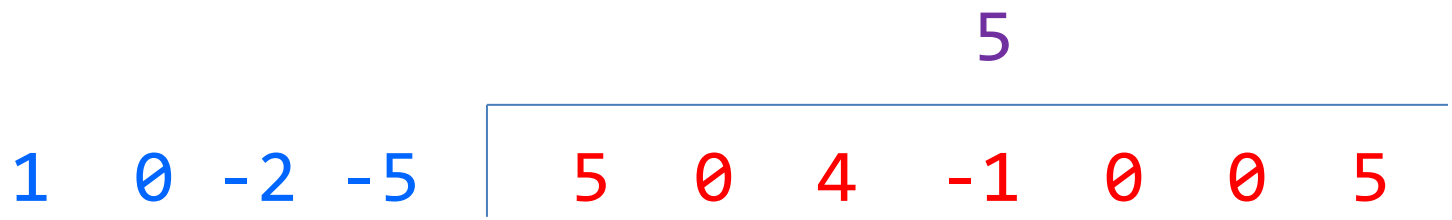
Polynomial Division

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

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

Polynomial Division

Polynomial Division



Polynomial Division

```
                                   5
                   ___________________________________
1   0  -2  -5  |    5    0    4    -1    0    0    5
                    5    0-10  -25
```

Polynomial Division

```
                    5
          ┌──────────────────────
1  0 -2 -5 |  5  0  4  -1  0  0  5
           |  5  0-10 -25
           ├──────────────────────
           |  0  0 14  24
```

Polynomial Division

```
                         5
            ----------------------------------
1  0 -2 -5 |  5  0  4  -1  0  0  5
              5  0-10 -25
              --------------
              0  0 14  24
                   14  24 0
```

Polynomial Division

```
                              5   0
                 _________________________________
1   0  -2  -5   |   5   0   4   -1   0   0   5
                    5   0-10  -25
                   ________________
                    0   0  14   24
                           14   24  0
```

Polynomial Division

```
                        5  0
           ______________________________
1  0 -2 -5 |  5  0  4  -1  0  0  5
              5  0-10 -25
              ---------------
              0  0 14  24
                   14  24 0
                   14  24 0  0
```

Polynomial Division

```
                       5  0 14
           ________________________________
1  0 -2 -5 |  5  0  4  -1  0  0  5
              5  0-10 -25
              _______________
              0  0 14  24
                   14  24 0
                   14  24 0  0
```

Polynomial Division

5 0 14

1 0 -2 -5 | 5 0 4 -1 0 0 5

5 0-10 -25

---

0 0 14 24

14 24 0

14 24 0 0

14 0 -28 -70

Polynomial Division

```
                    5   0 14
              ┌──────────────────────
1   0 -2 -5   |  5   0   4   -1   0   0   5
              |  5   0-10 -25
              |-------------------------
              |  0   0 14   24
              |        14   24 0
              |        14   24 0   0
              |        14   0 -28 -70
              |-------------------------
              |        0   24   28   70
```

Polynomial Division

```
                    5   0 14
              ┌──────────────────────
1   0 -2 -5   |  5   0   4   -1   0   0   5
              |  5   0-10 -25
              ├──────────────────────
              |  0   0 14   24
              |        14   24 0
              |        14   24 0   0
              |        14   0 -28 -70
              ├──────────────────────
              |  0   24   28   70
              |      24   28   70 5
```

Polynomial Division

```
                        5   0 14 24
            ___________________________
1  0 -2 -5 |  5   0   4   -1   0   0   5
              5   0-10 -25
              ________________
              0   0 14   24
                    14   24 0
                    14   24 0   0
                    14   0 -28 -70
                   ____________________
                    0   24   28   70
                        24   28   70 5
                        24   0   -48 120
```

Polynomial Division

5 0 14 24

1 0 -2 -5 | 5 0 4 -1 0 0 5

5 0-10 -25

0 0 14 24

14 24 0

14 24 0 0

14 0 -28 -70

0 24 28 70

24 28 70 5

24 0 -48 120

0 28 118 125

Polynomial Division

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

$5x^3 + 14x + 24$

Polynomial Division

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

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

RatNum & RatTerm

- RatNum
  - ADT for a Rational Number
  - Immutable
  - Has NaN
  - Implement add/sub/mul div
- RatTerm
  - Single polynomial term
  - Coefficient (RatNum) & degree
  - Basic math operations

RatPoly & RatPolyStack

- RatPoly
  - Sum of RatTerms
  - Implement add/sub/div/mul (as demonstrated)

- RatPolyStack
  - Push/pop terms
  - add/sub/div/mul top two

CalculatorFrame

- Quick demo

Javadoc

- Javadoc comments are a special kind of Java comment

```
/**
 *  This is a javadoc comment.
 */
```

```
// This is just a regular comment.
```

- Javadoc comments are used to communicate your API to the outside world.
- Automatic documentation generation, Eclipse hovercards

Recall: Class Specifications

- Describe **abstract value:** what the class represents at an abstract level
  - What the client sees
  - What data the ADT holds
- Brief summary of the ADT
- Doesn’t describe implementation details

Recall: Representation Invariant

- Constrains an object's internal state
- Defines what must be true for abstraction function to hold
- If representation invariant is violated:
  - Object is "broken" – doesn't map to any abstract value

Verifying RI

- Representation invariant should hold before and after every public method

- Write & Use checkRep()
  - Call before and after public methods
  - OK that it adds extra code
    - Asserts won't be included on release builds
    - Important for finding bugs

Aside on optimization

- "More computing sins are committed in the name of efficiency (without necessarily achieving it) than for any other single reason — including blind stupidity." — W.A. Wulf

- “…[A]bout 97% of the time: premature optimization is the root of all evil. A good programmer … will be wise to look carefully at the critical code; but only after that code has been identified" — Donald Knuth

- "The First Rule of Program Optimization: Don't do it. The Second Rule of Program Optimization (for experts only!): Don't do it yet." — Michael A. Jackson

Source: http://en.wikipedia.org/wiki/Program_optimization#Quotes

Recall: Abstraction Function

- Abstraction function: a **mapping** from **internal state** to **abstract value**

- Abstract fields may not map directly to representation fields
  - Circle has **radius** but not necessarily

```
private int radius;
```

- Internal representation can be anything as long as it somehow encodes the abstract value

ADT Example: NonNullStringList

```
public class NonNullStringList  {
        // AF = ???
        // RI = ???

        public void add(String s) { ... }
        public boolean remove(String s) { ... }
        public String get(int i) { ... }
}
```

ADT Example: NonNullStringList

```
public class NonNullStringList  {
        // AF = ???
        // RI = ???

        public void add(String s) { ... }
        public boolean remove(String s) { ... }
        public String get(int i) { ... }
}
```

Internal representation with an array

ADT Example: NonNullStringList

```
public class NonNullStringList  {
        // AF = Index i in arr contains the ith element in the list
        // RI = [0,count-1] != null

        private String[] arr;
        private int count;


        public void add(String s) { ... }
        public boolean remove(String s) { ... }
        public String get(int i) { ... }
}
```

ADT Example: NonNullStringList

```
public class NonNullStringList  {
        // AF = Index i in arr contains the ith element in the list
        // RI = [0,count-1] != null

        private String[] arr;
        private int count;


        public void add(String s) { ... }
        public boolean remove(String s) { ... }
        public String get(int i) { ... }
}
```

How to implement remove?

ADT Example: NonNullStringList

```
public class NonNullStringList  {
        // AF = ???
        // RI = ???

        public void add(String s) { ... }
        public boolean remove(String s) { ... }
        public String get(int i) { ... }
}
```

Internal representation with a LinkedList

ADT Example: NonNullStringList

```
public class NonNullStringList  {
        // AF = Value in the nth node after head contains the nth item
        //      in the list
        // RI = Head has size nodes after it, each whose value is non-
        //      null, no cycle in ListNodes

        public int size;
        public ListNode head;



        public void add(String s) { ... }
        public boolean remove(String s) { ... }
        public String get(int i) { ... }
}
```

ADT Example: ComplexNumber

```
public class ComplexNumber  {

        // AF = ???
        // RI = ???

        public double getReal() { }
        public double getImag() { }
        public double getAbs() { }
}
```

- Two possible implementations
  - x, y
  - r, theta

ADT Example: Circle

```
public class Circle {

        // AF = ???
        // RI = ???

        public boolean isPointOnBorder(Point p) { }
        public boolean containPoint(Point p) { }
}
```

- Two possible implementations
  - Center, radius
  - Center, edge point