

Topic 5: Developing Code

Using Java Math Class

Lecture Contents

- Java Class Library
- The Math Class
 - Math Constants
 - Math Methods

Java Class Library

- Java Class Library (JCL) is part of the Java Development Kit (JDK)
 - A comprehensive collection of pre-built classes and methods
 - Provides essential functionality for Java applications
 - Includes functionality in classes, for example:
 - `Math` (which we'll discuss in this slide show)
 - `Array` (sorting, etc.)
 - `File` (for file I/O)
 - `ArrayList`
 - `HashMap`

Math Constants

- The `Math` class contains constants for π and e .
 - Constants in Java are, by convention, in `UPPER_CASE`

```
public class PlayWithMath {  
    public static void main(String args[]) {  
        System.out.println("The value of pi is: " + Math.PI);  
        System.out.println("The value of e is: " + Math.E);  
    }  
}
```

```
The value of pi is 3.141592653589793.  
The value of e is 2.718281828459045.
```

Math Constants

- Use the constant `Math.PI` to write method `calculateCircumference`

```
public static void main(String args[]) {  
    double c = calculateCircumference(3.0);  
    System.out.println("Circumference: " + c);  
}
```

Circumference: 18.84955592153876

Math Methods

- **Math.abs(-5);**
- **Math.sqrt(2.0);**
- **Math.min(3, 5);**
- **Math.max(3, 5);**
- **Math.sin(3.14);**
- **Math.asin(0.5);**
- **Math.pow(2, 5);**
- **Math.random();**

```
public static void main(String[] args) {  
    System.out.println(Math.abs(-5.0));  
    System.out.println(Math.)  
}
```

S E : double - Math
S PI : double - Math
S abs(double a) : double - Math
S abs(float a) : float - Math
S abs(int a) : int - Math
S abs(long a) : long - Math
S absExact(int a) : int - Math
S absExact(long a) : long - Math
S acos(double a) : double - Math
S addExact(int x, int v) : int - Math

Press '^Space' to show Template

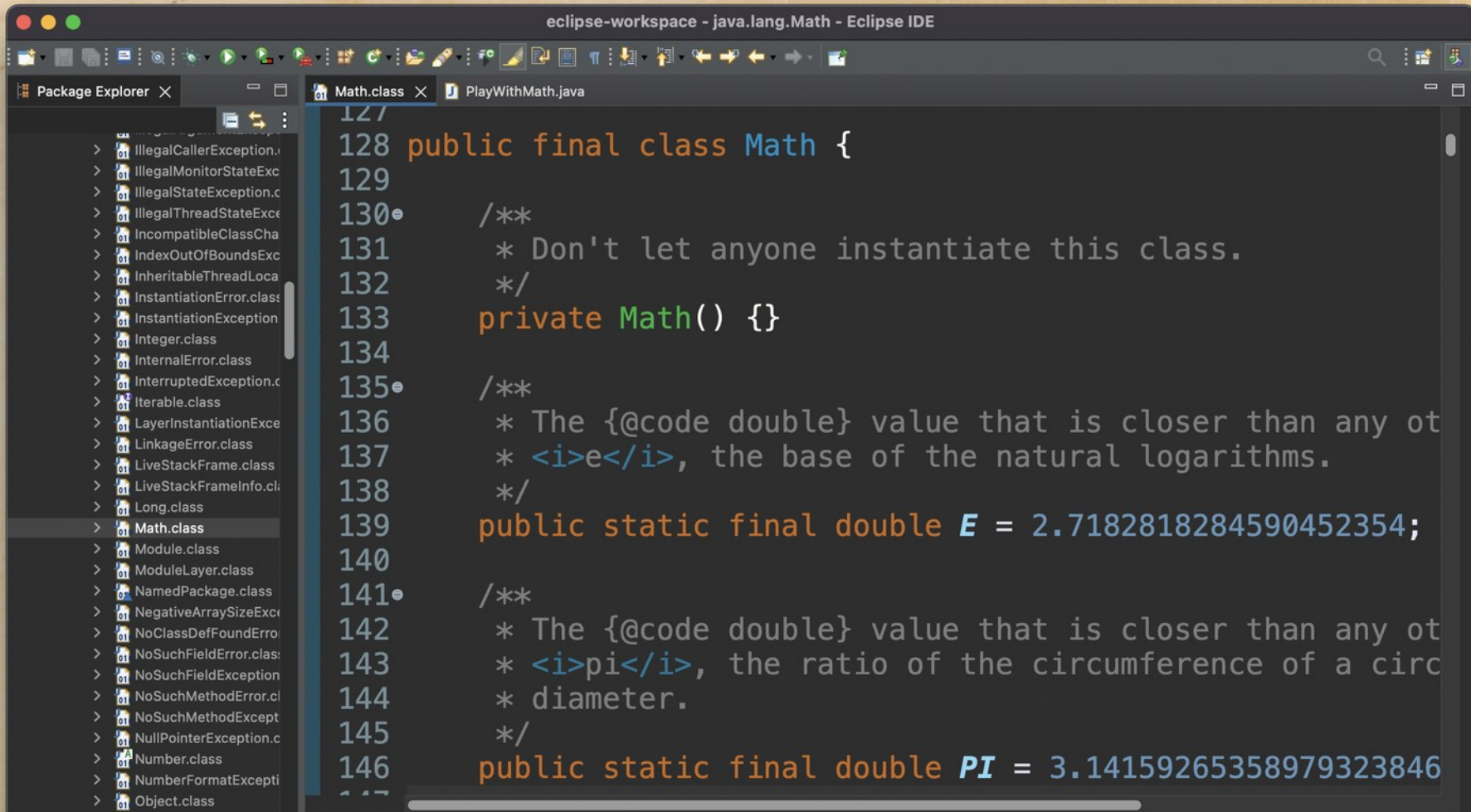
Math Methods

- Use the constant `Math.sqrt` (and `Math.pow`, if you wish) to write method `hypotenuseLength`

```
public static void main(String args[]) {  
    double len = hypotenuseLength(3.0, 4.0);  
    System.out.println("Hypotenuse Length: " + len);  
}
```

Hypotenuse Length: 5.0

- You can view the code of the Math class and compare it to the classes we write.



```
127
128 public final class Math {
129
130     /**
131      * Don't let anyone instantiate this class.
132      */
133     private Math() {}
134
135     /**
136      * The {@code double} value that is closer than any ot
137      * <i>e</i>, the base of the natural logarithms.
138      */
139     public static final double E = 2.7182818284590452354;
140
141     /**
142      * The {@code double} value that is closer than any ot
143      * <i>pi</i>, the ratio of the circumference of a circ
144      * diameter.
145      */
146     public static final double PI = 3.14159265358979323846
147
```

If you take AP CSc A

- AP Java Subset

Math Class	
<code>static int abs(int x)</code>	Returns the absolute value of an <code>int</code> value
<code>static double abs(double x)</code>	Returns the absolute value of a <code>double</code> value
<code>static double pow(double base, double exponent)</code>	Returns the value of the first parameter raised to the power of the second parameter
<code>static double sqrt(double x)</code>	Returns the positive square root of a <code>double</code> value
<code>static double random()</code>	Returns a <code>double</code> value greater than or equal to <code>0.0</code> and less than <code>1.0</code>

Topic 5: Developing Code

Using Java Math Class