

Using Classes

Math Class

Lecture Contents

- The Math Class
 - Math Constants
 - Math Methods

Math Constants

```
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.

Write method `calculateCircumference()`

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

Circumference: 18.84955592153876

Write method calculateCircumference()

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

Circumference: 18.84955592153876

Write method calculateCircumference()

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

```
public static double calculateCircumference(double radius) {  
    return 2 * Math.PI * radius;  
}
```

Circumference: 18.84955592153876

Write method calculateCircleArea()

```
public static void main(String args[]) {  
    double c = calculateCircumference(3.0);  
    double a = calculateCircleArea(3.0);  
    System.out.println("Circumference: " + c);  
    System.out.println("Area: " + a);  
}  
  
public static double calculateCircumference(double radius) {  
    return 2 * Math.PI * radius;  
}
```

```
Circumference: 18.84955592153876  
Area: 28.274333882308138
```

Write method calculateCircleArea()

```
public static void main(String args[]) {  
    double c = calculateCircumference(3.0);  
    double a = calculateCircleArea(3.0);  
    System.out.println("Circumference: " + c);  
    System.out.println("Area: " + a);  
}  
  
public static double calculateCircleArea(double radius) {  
    return Math.PI * radius * radius;  
}  
  
public static d  
    return 2 * M  
}
```

Circumference: 18.84955592153876
Area: 28.274333882308138

Math Methods

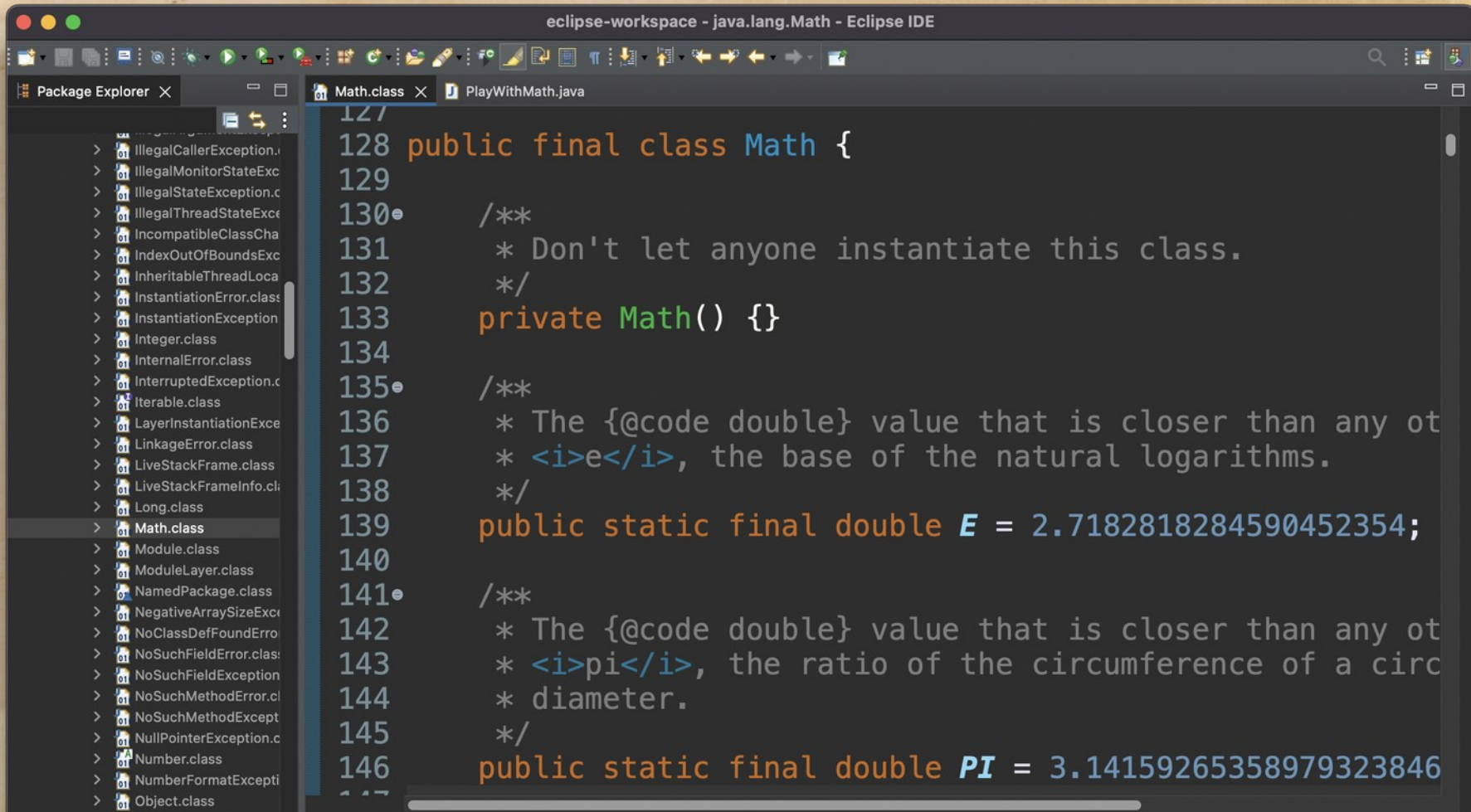
- **Math.abs(-5);**
- **Math.sqrt(2.0);**
- **Math.min(3, 5);**
- **Math.max(3, 5);**
- **Math.sin(3.14);**
- **Math.asin(0.5);**
- **Math.pow(2, 5);**
- **Math.random();**

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

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

Press '^Space' to show Template

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



The screenshot shows the Eclipse IDE interface. The title bar reads "eclipse-workspace - java.lang.Math - Eclipse IDE". The Package Explorer on the left lists various Java classes, with "Math.class" selected. The main editor window displays the source code of the `Math` class. The code is as follows:

```
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
```

Java AP 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>

Using Classes

Math Class