



ORACLE

Academy



Java Foundations

4-5

Clase Math

ORACLE
Academy



Objetivos

- En esta lección se aborda el siguiente objetivo:
 - Conocer los métodos de la clase Math
 - Utilizar los métodos de la clase Math para realizar cálculos matemáticos
 - Utilizar los campos de la clase Math



Realización de cálculos matemáticos

- Durante el desarrollo de programas, es posible que necesite cálculos matemáticos más avanzados que los operadores matemáticos básicos que proporciona Java
- Por ejemplo:
 - Buscar el máximo o el mínimo de dos valores
 - Redondear valores
 - Funciones logarítmicas
 - Raíz cuadrada
 - Funciones trigonométricas
- La clase Math de Java contiene métodos para cálculos matemáticos

Clase Math

- Es una de las muchas clases incluidas en las bibliotecas de clases Java
- Contiene métodos que realizan diversas funciones matemáticas
- Forma parte del paquete `java.lang`

Documentación de la clase Math

- Puede acceder a la documentación desde aquí:
 - <https://docs.oracle.com/en/java/javase/17/docs/api/java.base/module-summary.html>

OVERVIEW MODULE PACKAGE **CLASS** USE TREE DEPRECATED INDEX HELP Java SE 15 & JDK 15

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD SEARCH:

Module java.base
Package java.lang
Class Math
java.lang.Object
 java.lang.Math

public final class Math

Methods for performing basic numeric operations such as the elementary exponential, logarithm, square root, and trigonometric functions.

Specific methods of class StrictMath, all implementations of the equivalent functions of class Math are not defined to return the bit-for-bit same results. Better-performing implementations where strict reproducibility is not required.

Math methods simply call the equivalent method in StrictMath for their implementation. Code generators are encouraged to use platform-specific microprocessor instructions, where available, to provide higher-performance implementations of Math methods. Such higher-performance implementations still must conform to the specification for Math.

The quality of implementations of floating-point operations concern two properties, accuracy of the returned result and monotonicity of the method. Accuracy of the floating-point Math methods is measured in terms of *ulps* (units in the last place). For a given floating-point format, an ulp of a specific real number value is the distance between the two floating-point values bracketing that numerical value. When discussing the accuracy of a method as a whole rather than at a specific argument, the number of ulps cited is for the worst-case error at any argument. If a method always has an error less than 0.5 ulps, the method always returns the floating-point number nearest the exact result; such a method is *correctly rounded*. A correctly rounded method is generally the best a floating-point approximation can be; however, it is impractical for many floating-point methods to be correctly rounded. Instead, for the Math class, a larger error bound of 1 or 2 ulps is allowed for certain methods. Informally, with a 1 ulp error bound, when the

Desplácese para ver una lista de los campos y métodos disponibles en esta clase

Ejercicio 1

- Examine la documentación de la clase Math
- Compruebe si puede encontrar un valor para PI y un método para calcular la raíz cuadrada de un número

Algunos métodos disponibles en la clase Math

Nombre del método	Descripción
<code>abs(value)</code>	valor absoluto
<code>ceil(value)</code>	redondea a la alta
<code>cos(value)</code>	coseno, en radianes
<code>floor(value)</code>	redondea a la baja
<code>log(value)</code>	logaritmo de base e
<code>log10(value)</code>	logaritmo de base 10
<code>max(value1, value2)</code>	el mayor de dos valores
<code>min(value1, value2)</code>	el menor de dos valores
<code>pow(base, exponent)</code>	base elevada a la potencia del exponente
<code>random()</code>	valor double aleatorio entre 0 y 1
<code>round(value)</code>	número entero más cercano
<code>sin(value)</code>	seno, en radianes
<code>sqrt(value)</code>	raíz cuadrada

¿En qué se diferencia la clase Math?

- Los métodos de la clase Math son estáticos
- Los métodos estáticos se pueden llamar a través del nombre de clase
- Esto significa que no tiene que crear un objeto de la clase Math para llamar a los métodos
- Por ejemplo, para llamar a los métodos de la clase Random, tiene que crear un objeto de la clase Random del siguiente modo:

```
Random rndNum = new Random();  
int randomNum = rndNum.nextInt();
```

¿Cómo se llama a los métodos de la clase Math?

- Puede llamar a los métodos de la clase Math sin crear una instancia de la clase Math; por ejemplo, de esta forma:
- Sintaxis:
 - `Math.methodName(parameters)`
- Ejemplo:
 - `Math.sqrt(121.0);`

Llame a los métodos anteponiendo Math y el operador de punto

Llamada a un método y observación de los resultados

- Veamos un ejemplo de llamada a un método y observación de su resultado:

```
public static void main(String[] args) {  
  
    Math.sqrt(121.0);  
} //end method main
```

- Observe la salida:
 - No se muestra ninguna salida
 - Solo llamar a estos métodos no produce ningún resultado visible

¿Cómo funcionan los métodos de la clase Math?

- Los métodos de Math no imprimen los resultados en la consola
- Cada método devuelve un resultado numérico
- El valor de devolución es más flexible que el de impresión
- Puede guardarlo, imprimirlo o combinarlo con una expresión mayor

Almacenamiento e impresión de los resultados

- Para ver el resultado, debe imprimirlo o almacenarlo en una variable.
- Por ejemplo:
 - Imprimir el resultado:

```
public static void main(String[] args) {  
    System.out.println("Square root: " + Math.sqrt(121.0)); //11.0  
} //end method main
```

- Almacenar el valor:

```
public static void main(String[] args) {  
    double sqroot= Math.sqrt(121.0);  
    System.out.println("Square root: " + sqroot); //11.0  
} //end method main
```

Combinación de los resultados

- Puede combinar los resultados y utilizarlos en una expresión mayor, por ejemplo, de esta forma:

```
public static void main(String[] args) {  
    double result = Math.min(3, 7) + Math.abs(-50);  
    System.out.println("Result is " + result); //53  
} //end method main
```

Ejercicio 2

- En papel, evalúe las siguientes sentencias Java y registre los resultados:
 - `Math.abs(-1.23)`
 - `Math.pow(3, 2)`
 - `Math.sqrt(121.0) - Math.sqrt(256.0)`
 - `Math.abs(Math.min(-3, -5))`



Ejercicio 3

- Considere una variable de entero denominada `age`
- Utilice los métodos `Math.max` y `Math.min` para responder a las siguientes preguntas:
 - ¿Qué expresión sustituiría las edades negativas con 0?
 - ¿Qué expresión limitaría la edad máxima a 40?

Campos de la clase Math

- La clase `Math` contiene dos campos constantes:
 - `PI` y `E`

Field	Description
<code>Math.E</code>	2.7182818...
<code>Math.PI</code>	3.1415926...

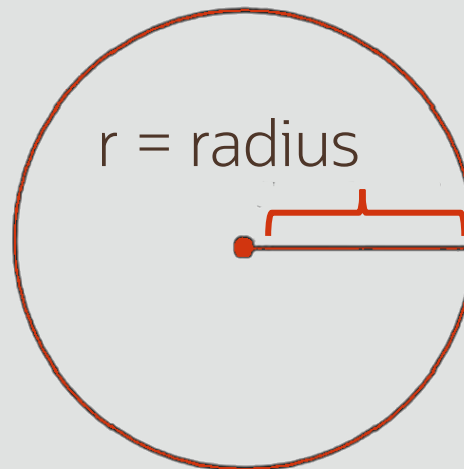


Campo PI

- La clase Math contiene una constante, PI
- Contiene un valor double: 3.14159265358979323846
- Recuerde que los métodos de la clase Math son estáticos y se puede acceder a ellos mediante el nombre de clase Math
- Del mismo modo, PI es una variable estática de la clase Math, por lo que se puede acceder a ella utilizando el nombre de clase Math
- Para utilizar PI en un programa, especifique el nombre de clase (Math) y PI, separados por el operador de punto:
 - **Math.PI**

Cálculo del área de un círculo

- Suponga que tiene que escribir un programa Java para calcular el área de un círculo
- La fórmula para calcular el área de un círculo es la siguiente:
 - $\text{Área} = \text{PI} * \text{radio} * \text{radio}$
 - Donde PI es una constante (aproximadamente 3,1416)





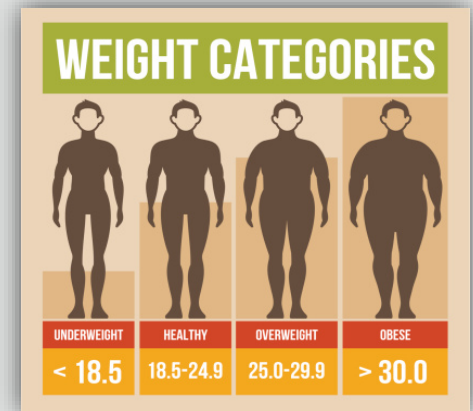
Cálculo del área de un círculo

- Utilizar el campo `Math.PI` para calcular el área produce un resultado más preciso que usar un valor constante para pi, por ejemplo, 3,14

```
public class AreaOfCircle {  
    public static void main(String args[]) {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter the radius: ");  
        double radius = sc.nextDouble();  
        double area = Math.PI * radius * radius;  
        System.out.println("The area of circle is: " + area);  
    } //end method main  
} //end class AreaOfCircle
```

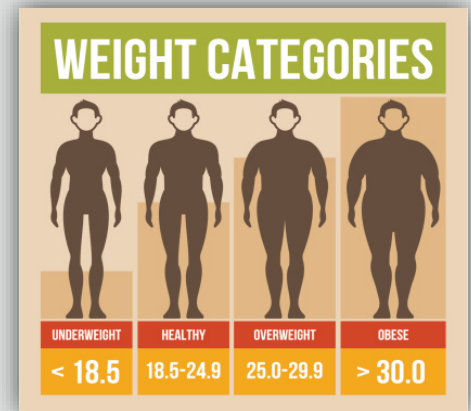
Ejercicio 4

- El índice de masa corporal (IMC) de una persona se calcula del modo siguiente: $BMI = \frac{\text{peso}}{\text{altura}^2} \times 703$
- Cree un nuevo proyecto y agréguele el archivo `ComputeBMI.java`
- Escriba un programa que calcule el IMC y lo redondee



Ejercicio 4

- Utilice los métodos de la clase Math y muestre la salida del siguiente modo:
 - Introduzca el peso en libras: 132,5
 - Introduzca la altura en pulgadas: 62,5
 - El índice de masa corporal es 24



Resumen

- En esta lección, debe haber aprendido lo siguiente:
 - Conocer los métodos de la clase Math
 - Utilizar los métodos de la clase Math para realizar cálculos matemáticos
 - Utilizar los campos de la clase Math





ORACLE
Academy

