Classe Math

Vimos anteriormente como tratar números de vários tipos. Veremos agora, como aplicar conceitos matemáticos mais complexos a nossos processamentos usando a classe Math do pacote Java.lang.

A classe Math nos proporciona uma série de operações e constantes matemáticas que são facilmente acessadas estaticamente, ou seja, não precisamos instanciar um objeto para podermos usar seus métodos.

Dentro desta classe encontramos vários métodos e categorizamos os mais utilizados da seguinte forma:

- Máximo e Mínimo;
- Potências e Raízes;
- Logaritmo;
- Arredondamentos e Valores Absolutos;
- Trigonometria;
- Números Randômicos.

Constantes

A classe Math possui duas constantes que são o Π (pi) e E (base de logaritmos naturais ou neperianos) cada uma com o valor de 3,141592653589793 e 2.718281828459045, respectivamente.

Abaixo está um exemplo simples com o cálculo do comprimento do círculo:

```
public class ComprimentoCirculo {
    public static void main(String[] args) {
        float raio = 2.4f;
        double comprimento = 2 * raio * Math.PI;
        System.out.println(comprimento);
    }
}
```

Máximo e Mínimo

Como o título já sugere, a classe Math de Java possui dois métodos que retornam o maior e o menor valor de seus argumentos. Esses métodos são max() e min().

```
public class MenorMaior {
    public static void main(String[] args) {
        float precoProdutoA[] = { 11.2f, 15.12f };
        float precoProdutoB[] = { 19.7f, 20 };
        System.out.println("O maior preço do produto A é "
                    + Math.max(precoProdutoA[0], precoProdutoA[1]));
        System.out.println("O menor preço do produto B é "
                    + Math.min(precoProdutoB[0], precoProdutoB[1]));
    }
}
```

Potências e raízes

Podemos fazer cálculos de potência e raízes com facilidade usando os métodos disponíveis em Math.

- pow (base, expoente) - calcula a potência da base elevada ao expoente.
- sqrt (número) - calcula a raíz quadrada de um número
- cbrt (número) - calcula a raiz cúbica de um número
- exp (expoente) - calcula o valor da constante de Euler (E) elevado ao expoente

```
public class PotenciasRaizes {
    public static void main(String[] args) {
        System.out.println("1 MB tem " + Math.pow(2, 10) + " KB");
        System.out.println("A raiz quadrada de 121 é " + Math.sqrt(121)
            + " e a raiz cúbica de 1331 também é " + Math.cbrt(1331));
    }
}
```

Logaritmo

Na classe Math encontramos funções para cálculo de logaritmo natual, na base de 10 e a soma do número mais 1. Tais métodos são:

- log (número) - logaritmo natural de um número.
- log10 (número) - logaritmo natural de um número na base 10
- log1p (número) - logaritmo natual de um número somado a 1. Esse método retorna um resultado mais confiável se o número em questão for muito próximo a 0 ou um número fracionado. Ou seja, o resultado não é o mesmo entre os métodos log1p (0.1f) e log (1+0.1f).

```
public class Exemplo {
    public static void main(String[] args) {
        float nr = 0.1f;
        System.out.println("Resultado 1: " + Math.log(nr+1));
        System.out.println("Resultado 2: " + Math.log1p(nr));
    }
}
```

Arredondamentos e Valores Absolutos

Existem algumas formas de arredondar um número fracionado (float e double) transformando-o em um número inteiro e também como obter o valor absoluto de qualquer número..

- abs (número) - retorna o valor absoluto do mesmo tipo do parâmetro (ex.: inteiro retorna int positivo, decimal retorna float positivo, etc)
- ceil (decimal) - este método retorna o valor decimal do parâmetro sem a parte fracionada. Ex.: 2.1 será 2, 6.0 será 6, 10.8 será 10...
- floor (decimal) - este método retorna o primeiro inteiro após o valor decimal. Ex.: 2.1 será 3, 6.0 será 6, 10.8 será 11...
- rint (decimal) - retorna um valor double mais próximo do valor do parâmetro.
- round (decimal) - retorna o arredondamento aritmético do número decimal passado como parâmetro

```
public class Exemplo {
    public static void main(String[] args) {
        float nr = -5.75f;
        System.out.println("Absoluto: " + Math.abs(nr) +
                "\nInteiro mais baixo: " + Math.ceil(nr) +
                "\nInteiro mais alto: " + Math.floor(nr) +
                "\nDouble mais próximo: " + Math.rint(nr) +
                "\nArredondamento: " + Math.round(nr));
    }
}
```

Trigonometria

A maior parte dos métodos encontrados na classe Math são trigonométricas, o que ajuda muito em cálculos mais complexos que envolvem figuras.

- sin (graus) - este método retorna o valor do seno de acordo com o número de graus

passado como parâmetro.

- cos (graus) - este método retorna o valor do coseno de acordo com o número de graus passado como parâmetro.
- tan (graus) - este método retorna o valor da tangente de acordo com o número de graus passado como parâmetro.
- asin (graus) - este método retorna o valor do arco seno de acordo com o número de graus passado como parâmetro.
- acos (graus) - este método retorna o valor do arco coseno de acordo com o número de graus passado como parâmetro.
- atan (graus) - este método retorna o valor do arco tangente de acordo com o número de graus passado como parâmetro.
- sinh (graus) - este método retorna o valor hiperbólico do seno de acordo com o número de graus passado como parâmetro.
- cosh (graus) - este método retorna o valor hiperbólico do coseno de acordo com o número de graus passado como parâmetro.
- tanh (graus) - este método retorna o valor hiperbólico da tangente de acordo com o número de graus passado como parâmetro.
- hypot (x , y) - retorna o valor da hipotenusa, ou, basicamente, a distância entre dois pontos fundamentada na fórmula $\sqrt{x^2+y^2}$ » [sqrt (pow(x, 2) + pow(y,2))].
- toRadians (graus) - retorna um valor aproximado de radianos de acordo com o ângulo medido em graus.
- toDegrees (raio) - retorna um valor aproximado de graus de acordo com o ângulo medido em raios.

```
public class Hipotenusa {
    public static void main(String[] args) {
        double ponto1 = 30;
        double ponto2 = 40;
        System.out.println("A distancia entre o "
                + ponto1 + " e o " + ponto2 + " é "
                + Math.hypot(ponto1, ponto2));
    }
}
```

Números Randômicos

Os números randômicos são obtidos usando o método random().

O método random() retorna um valor double em 0.0 e 1.0.

Para conseguirmos um valor limite ou um alcance (comumente chamado de range) delimitado, devemos fazer pequenas operações matemáticas.

Essas operações são simples e podem ser resumidas da seguinte maneira.

- O limite inferior, ou valor inicial (start value) será sempre somado ao número randômico.
- O limite superior, ou alcance (range) será sempre o limite superior subtraído o limite inferior e depois multiplicado ao número randômico.

Por exemplo, se quisermos que um número randômico sempre fique entre 5 e 10, procederíamos da seguinte maneira:

O menor número possível é 5, portanto ele será nosso start value ou limite inferior.

O maior número possível é 10, então 10 será nosso limite superior.

Como temos ambos os limites (inferior e superior) devemos criar um alcance (range). Para

obtermos isso, subtrairemos o limite superior com o limite inferior (10-5=5).

```
public class Exemplo {
    public static void main(String[] args) {
        int limiteInferior = 5;
        int limiteSuperior = 10;
        int alcance = limiteSuperior - limiteInferior;
        double nrRandomico = Math.random();
        System.out.println("O número randômico escolhido entre 5 e 10 foi "
                + Math.round(limiteInferior + nrRandomico * alcance));
    }
}
```