

La classe `java.lang.Math`

Temps de lecture : 2 minutes

La classe `java.lang.Math` propose des constantes et des méthodes statiques très utiles pour réaliser un large panel de calculs mathématiques. Comme tout est statique, nous n'allons pas créer d'objet pour s'en servir. Toutes les méthodes statiques reçoivent en paramètres d'entrée des types primitifs. `Math` permet donc de réaliser des calculs sur les types primitifs qui ne sont pas couverts par les opérateurs binaires.

Note : Cette classe n'a pas besoin d'être importée manuellement, elle est utilisable directement dans le code. C'est un avantage des classes du package `java.lang` (`String` fait également partie de ce package, nous ne l'importons jamais).

Les constantes

Les constantes suivantes sont présentes dans la classe `Math` :

```
public static final double E = 2.7182818284590452354;  
public static final double PI = 3.14159265358979323846;  
private static final double DEGREES_TO_RADIANS = 0.0174532  
92519943295;  
private static final double RADIANS_TO_DEGREES = 57.295779  
51308232;
```

- `Math.E` : une approximation de la valeur du nombre d'Euler.
- `Math.PI` : une approximation de la valeur de pi.
- `Math.DEGREES_TO_RADIAN` : une approximation du facteur pour convertir des degrés en radians.
- `Math.RADIANS_TO_DEGREES` : une approximation du facteur pour convertir des radians en degrés.

Note : Les approximations de ces constantes ont la précision maximale offerte par `double`. Ces constantes sont utiles pour rendre certains calculs le plus rapide possible sans avoir à les recalculer chaque fois.

Les fonctions de calculs

Les fonctions trigonométriques

Math déclare les fonctions classiques de la trigonométrie

: `sin(n)`, `cos(n)`, `tan(n)`, `asin(n)`, `acos(n)`, `atan(n)`.

Elle déclare également les versions hyperboliques des fonctions trigonométriques

: `sinh(n)`, `cosh(n)`, `tanh(n)`.

Important : Ces fonctions acceptent en entrée un paramètre d'entrée un `double`.

Celui-ci doit être en radians. Si vous travaillez avec des degrés, les deux méthodes suivantes existent pour faire les conversions : `toDegrees()` et `toRadians()` qui prennent également un `double` en entrée.

Les fonctions de comparaisons

Math déclare deux fonctions de comparaisons : `min(n1, n2)` et `max(n1, n2)`.

Elles acceptent deux numériques en paramètres d'entrées et retournent la valeur minimales ou la valeur maximale entre les deux.

Les fonctions d'arrondis

Pour arrondir des valeurs numériques, les fonctions suivantes sont déclarées :

- `round(n)`
 - Pour les types `float` et `double`, cette méthode ajoute 0,5 à l'argument et restitue la plus grande valeur entière (`int`) inférieure ou égale au résultat.
- `rint(n)`
 - Pareil que `round(n)` mais renvoie un `double`.
- `floor(n)`
 - Cette méthode renvoie l'entier le plus proche inférieur ou égal à l'argument.
- `ceil(n)`
 - Cette méthode renvoie l'entier le plus proche supérieur ou égal à l'argument

Valeur absolue

Math déclare la méthode `abs(n)`. Celle-ci reçoit un numérique en entrée et retourne sa valeur absolue (transforme les valeurs négatives par leur opposé).

Puissances et exponentielles

Math déclarer les méthodes suivantes :

- `pow(n1, n2)`
 - Cette méthode élève le premier argument à la puissance indiquée par le second.
- `sqrt(n)`
 - Cette méthode calcule la racine carrée de son paramètre.
- `exp(n)`
 - Cette méthode calcule l'exponentielle de l'argument
- `log(n)`
 - Cette méthode calcule le logarithme naturel de l'argument

Et les vecteurs ?

Il n'y a pas d'abstraction standard disponible en Java pour représenter les vecteurs. Cependant, on peut utiliser des matrices en mathématiques pour représenter des vecteurs. Il est donc assez facile de représenter des vecteurs en utilisant des tableaux ou des collections de données.

Note : Pour aller plus loin, vous pouvez donc vous amuser à représenter des vecteurs de cette manière. Sinon, des librairies de moteurs de jeu comme libGDX offrent des abstractions autour des vecteurs, mais pas seulement puisque c'est un moteur de jeu. En effet, les vecteurs sont très importants dans les calculs des moteurs graphiques.