

## NAME

class **PrimeFactors** in **algomaths** Perl6 modules : `maths/modules/pm6/prime-factors.pm6`

## AUTHOR

<https://github.com/Chrissealed/algomaths.git>

## VERSION

2019.01.07

## Ce module contient un rôle : ‘PrimeFactors’

destiné à établir la liste des facteurs premiers d’un entier différent de 0, c’est-à-dire de le décomposer en facteurs premiers.

Il contient deux méthodes, la première :

**breakdown(Int:D \$integer is copy where { (\$integer != 0) or die “Argument invalide! Nombre entier différent de 0 requis.” } -> Hash:D)**

avec **\$integer** différent de 0 qui renvoie un **Hash** comprenant le mappage des nombres avec leurs facteurs. Il affiche ce mappage sous plusieurs formes : en tableau associatif (Hash) clé => valeur, en facteurs successifs puis en facteurs élevés à la puissance requise.

Le rôle possède en outre un attribut requis destiné à écrire dans un fichier les informations qui apparaissent à l’écran et qui décrivent les différentes étapes des opérations. Il s’agit de **Teeput::Tput \$t is required is rw** : il faut lui passer un objet de type **Teeput::Tput** du module **teeput.pm6**. Référez-vous à la doc de ce module pour plus d’informations.

**La deuxième méthode : reduce-fractions-prime-factors(Int:D  
@numerators is copy, Int:D @denominators is copy, Int:D  
\$return-array -> Array:D)**

qui prend 2 tableaux de **Int** en arguments et délivre pour chacun des 2 tableaux un troisième et quatrième tableau contenant les valeurs qui se trouvent dans l'un des tableaux et pas dans l'autre. Selon la valeur de l'argument **\$return-array** (1 ou 2), le premier tableau ou le deuxième tableau sera retourné, pour disposer des valeurs contenues dans les 2 tableaux, il faudra faire deux appels avec une valeur différente de **\$return-array** pour chacun d'eux.