Les logins des utilisateurs

L'admin d'un organisme purement imaginaire est chargé de créer les nouveaux arrivants dans sa base de données. La base contient pour chaque utilisateur ses nom, prénom et login, plus d'autres informations qui ne relèvent pas de cet exercice.

Un utilisateur de l'organisme possède un nom, un prénom (des chaînes de caractères) et un login (au plus 8 lettres, minuscules ASCII). L'admin doit se creuser la tête pour trouver, à partir du nom et du prénom, un login qui n'a pas été déjà attribué, en suivant une certaine logique tout de même.

Il décide de suivre les règles suivantes :
- en première approche, le login est composé des (au max 8 premières) lettres du nom.
- si cette première approche échoue parce que ce login est déjà pris, seconde tentative : le login est composé des (au max 7 premières) lettres du nom concaténées à l'initiale du prénom.
- si cette tentative échoue aussi, alors l'admin arrête de chercher un login automatiquement. Il doit inventer lui-même un login et vérifier qu'il n'existe pas déjà.

L'admin programme un petit outil pour se faciliter la vie, avec 2 déclinaisons de la fonctionnalité principale :
- création d'un utilisateur dans la base à partir de ses nom, prénom et login ;
- en mode « même pas peur », création d'un utilisateur à partir de ses nom et prénom, le login étant calculé automatiquement à partir des règles présentées plus haut (ce qui peut échouer, donc).

Dans les 2 cas l'outil retourne soit les caractéristiques de l'utilisateur créé, soit une erreur.

On aura par ex les interactions suivantes, qui montrent dans l'ordre une création d'utilisateur avec recherche de login automatique qui réussit, une création d'utilisateur avec recherche de login automatique qui échoue, et enfin une création d'utilisateur avec login inventé, qui réussit.

```
>>> user = admin.creer_utilisateur("toblerone", "jean")
>>> print(user)
... tobleron ...
>>> user = admin.creer_utilisateur("souris", "anne-marie")
Traceback (most recent call last):
...
LoginExistant: login sourisa deja pris
>>> user = creer_utilisateur("souris", "anne-marie", "sourisam")
>>> print(user)
... sourisam ...
```

Exercice 1 : TDD et COO

Dans cet exercice il vous est demandé d'appliquer ce qu'on a vu en CTD pour tester l'outil de l'admin, cad les principes de COO qui produisent un code de qualité en parallèle du TDD et des tests en isolation :
- principe de SRP (**Single Responsability Principle**) pour obtenir de petites classes et des tests simples faciles à maintenir ;
- **injection des dépendances**, dépendances à des abstractions pour permettre l'utilisation de mocks.

Et surtout ne pas utiliser les anti-patterns qui rendent le code non testable.

Q 1.1 : Avant de commencer, écrire dans un `README` les scénarios que vous aimeriez réaliser pour tester l'outil dans les 2 cas suivants (sans oublier ce qu'on a raconté sur le test fonctionnel) :
- création d'un utilisateur dans la base à partir de ses nom, prénom et login ;
- création d'un utilisateur à partir de ses nom et prénom, le login étant calculé automatiquement à partir des règles présentées plus haut.

□

Q 1.2 : Écrire et tester en TDD et **en isolation** la classe principale qui fournit les fonctionnalités :
- création d'un utilisateur dans la base à partir de ses nom, prénom et login ;

— création d'un utilisateur à partir de ses nom et prénom, le login étant calculé automatiquement à partir des règles présentées plus haut.

Seules ces fonctionnalités sont demandées. □

Q 1.3 : Admirer votre œuvre et vérifier que vous avez respecté les principes de SRP et DI. Pour ce faire, demandez-vous si des changements apportés à l'outil (soit changement technique, par ex de la base de données, soit changement dans la spécification, par ex augmentation de la taille max des login) impactent cette classe et ses tests, et si cela indique une violation du SRP. Si besoin, lancez-vous dans un refactoring du code et des tests. □

Q 1.4 : (si vous avez fini) Écrivez et testez en TDD la classe qui gère les calculs sur chaîne en pensant bien aux cas particuliers : quels cas pénibles peut-on avoir à traiter avec des chaînes de caractères ? □