

# Épreuve pratique BAC NSI

## Un refuge de protection des renards

Le renard, longtemps considéré comme "nuisible", est aujourd'hui de plus en plus protégé pour son rôle dans la régulation de la biodiversité. Afin d'aider à la réhabilitation des individus blessés ou orphelins, un refuge de protection a été construit.

La personne en charge de l'infrastructure souhaite modéliser un prémice de base de données en CSV et Python pour stocker les informations essentielles sur les renards pris en charge.

Nous allons modéliser deux entités : le Renard et le Refuge.

- Un Renard est caractérisé par un identifiant (entier), un nom (chaîne de caractères), un poids en kg (flottant) et une date\_arrivee (chaîne AAAA-MM-JJ).
- Un Refuge est caractérisé par son nom, son adresse et une liste d'objets Renard.

Toutes les données sont fournies dans le fichier `renards.csv` qui est au format CSV avec séparateur `;`.

### Exemple:

```
id;nom;poids;date_arrivee
101;Edgar;6.5;2023-01-15
102;Cesar;5.8;2023-02-10
103;Gérard;7.2;2023-03-05
104;Sybille;4.9;2024-11-20
```

Un refuge possède un nom, une adresse et une liste de renards représentée par une liste d'objets Renards qui seront construits par la suite.

Le fichier `gestion_refuge.py` contient les squelettes de classes et fonctions qui vont être complétés et améliorés.

### 1. Modélisation de la classe Renard

Le constructeur de la classe Renard est fourni.

**Écrire** la méthode `__str__` à la classe `Renard` qui renvoie une chaîne de caractères présentant l'animal sous la forme : `"Renard ID [id] - [Nom] (Arrivé le [date_arrivee])"`.

**Tester** la classe en instanciant un renard qui a pour id `200`, son est nom `Oscar`, un poids de 5.1kg et est arrivé le 1er janvier 2025 au refuge et afficher ses informations.

**Appel 1 : Appeler le professeur pour lui présenter vos réponses et votre fonction ou en cas de difficultés de compréhension de la représentation.**

### 2. Modélisation de la classe Refuge

a. **Compléter** le constructeur de la classe `Refuge`. Il doit prendre le `nom` du refuge et l' `adresse` du refuge en paramètre et

initialiser l'attribut `liste_renard` comme une liste vide.

b. **Écrire** la méthode `recueillir(self, renard)` à la classe `Refuge` qui ajoute le renard à la liste `liste_renard`.

**Tester** en créant un refuge nommé "SOS Goupil" et en y ajoutant les renards Gérard, Edgar, César et Sybille de l'exemple.

**Appel 2 : Appeler le professeur pour lui présenter vos réponses et votre fonction ou en cas de difficultés de compréhension de la représentation.**

### 3. Importation des données

Une fonction `importer_donnees(nom_fichier, refuge)` est fournie dans le fichier `gestion_refuge.py`. Cependant, cette fonction contient une erreur liée au traitement des types de données et à l'en-tête du fichier CSV.

Python

### Extrait du code initial erroné de `gestion_refuge.py`

```
def importer_donnees(nom_fichier:str, refuge:Refuge)->None:
    print(f"Tentative d'importation depuis {nom_fichier}...")
    with open(nom_fichier, 'r', encoding='utf-8') as f:
        lignes = csv.DictReader(f, delimiter=';')
        for ligne in lignes:
            renard = Renard((ligne['id']), ligne['nom'], (ligne['poids']), ligne['date_arrivee'])
            refuge.recueillir(renard)
```

a. **Identifier** et expliquer les deux principaux problèmes dans le corps de la boucle `for` de la fonction fournie (un problème d'itération, un problème de type).

b. **Proposer** un test simple sous forme d'assertion qui permettrait de mettre en évidence le problème de type si l'on tentait de calculer la somme des poids des renards importés et **corriger le code de la fonction `importer_donnees`**.

**Tester** la fonction corrigée avec `renards_test.csv` et vérifier le nombre de renards importés.

**Appel 3 : Appeler le professeur pour lui présenter vos réponses et votre fonction ou en cas de difficultés de compréhension de la représentation.**

### 4. Analyse et méthode d'état de santé

Le refuge utilise l'importation de données pour surveiller la santé des animaux. On considère qu'un renard est en 'Sous-poids' si son poids est strictement inférieur à 6.0 kg.

a. **Compléter** le code de la méthode `lister_sous_poids(self)` dans la classe `Refuge`. Cette méthode doit renvoyer une liste des objets `Renard` qui sont en 'Sous-poids'.

b. **Écrire** le code de la méthode `pourcentage_sous_poids(self)` dans la classe `Refuge`. Cette méthode doit utiliser la méthode précédente et renvoyer le pourcentage (type `float`) de renards en 'Sous-poids' par rapport au nombre total de renards du refuge. Si le refuge est vide, elle doit renvoyer 0.0.

**Tester** les deux méthodes avec le refuge rempli par le fichier `renards_test.csv` et justifier le pourcentage obtenu en affichant

le nombre de renards en sous-poids et le nombre total de renards dans la fonction `pourcentage_sous_poids` .

**Appel 4: Appeler le professeur pour lui présenter vos réponses et votre fonction ou en cas de difficultés de compréhension de la représentation.**