



Exercice 4.1 – Module csv

Grâce au module *csv*, ajouter une méthode dans *NavireDeGuerre* qui sauvegarde toutes les informations d'un navire sous la forme d'un CSV.

Le nom du fichier CSV doit être de la forme « *navire_<nom du navire>.csv* ».

Exercice 4.2 – Module pickle

Grâce au module *pickle*, créer une méthode dans *NavireDeGuerre* qui sauvegarde un navire de guerre dans un fichier.

Le nom du fichier sauvegardé doit être de la forme « *navire_<nom du navire>.pickle* ».

Exercice 4.3 – Module pickle

Toujours grâce au module *pickle*, créer une méthode dans *NavireDeGuerre* qui charge un navire auparavant enregistré dans un fichier.

Exercice 4.4 – Module json

Grâce au module *json*, ajouter une méthode dans *NavireDeGuerre* qui sauvegarde toutes les informations d'un navire sous la forme d'un JSON.

Le nom du fichier CSV doit être de la forme « *navire_<nom du navire>.json* ».

Exercice 4.5 – Module json

Grâce au module *json*, ajouter une méthode dans *NavireDeGuerre* qui permet de recréer un navire à partir d'une sauvegarde JSON précédente.

Exercice 4.6 – Module sqlite3

Créer un module *connexion_bd*.

Grâce au module *sqlite3*, créer une instance de connexion à une base de données « *Connexion.db* ».

Créer une table *Navire*, qui permet de stocker toutes les informations d'un navire (pour simplifier, on n'enregistrera pas les informations sur la coque, et on n'enregistre uniquement le nom de l'arme).

La requête SQL est la suivante :

```
CREATE TABLE IF NOT EXISTS Navire(id INTEGER PRIMARY KEY AUTOINCREMENT,  
nom TEXT, fabricant TEXT, annee INTEGER, longueur FLOAT, puis_moteur  
FLOAT, kilometrage FLOAT, arme TEXT, passagers INTEGER, capacite INTEGER)
```

Exercice 4.7 – Module sqlite3

Toujours dans le module *connexion_bd*, créer une instance de *NavireDeGuerre* et une instance de *NavireCivil*.

Pour éviter des erreurs avec les attributs tels que annee, fabricant, etc., on va les ajouter manuellement. Ajouter cette instruction après la création du NavireDeGuerre et du NavireCivil (n doit être remplacé par le nom de la variable adéquate) :

```
n.__dict__.update(dict.fromkeys(['fabricant', 'annee', 'longueur', 'puis_moteur']))
```

Insérer les informations de ces deux navires dans la base de données.

La requête pour insérer un navire de guerre est la suivante :

```
INSERT INTO Navire(nom, fabricant, annee, longueur, puis_moteur,  
kilometrage, arme) VALUES (<les valeurs correspondantes>)
```

La requête pour insérer un navire civil est la suivante :

```
INSERT INTO Navire(nom, fabricant, annee, longueur, puis_moteur,  
kilometrage, passagers, capacite) VALUES (<les valeurs correspondantes>)
```

Exercice 4.8 – Module sqlite3

Afficher tout le contenu de la base de données.

La requête pour sélectionner toutes les informations de la table Navire est la suivante :

```
SELECT * FROM Navire
```

Exercice 4.9 – Pylint

Mettre en place le module *pylint* (voir le support).

Lancer une analyse de la qualité du code de la classe *NavireDeGuerre*, et modifier la classe jusqu'à ce que *pylint* ne signale plus aucun problème.

Exercice 4.10 – Tests unitaires

Effectuer quelques tests unitaires sur la classe *NavireDeGuerre*.