Rapport de Projet – PEIP2 INFO4

# Titre du projet :

Lecteur de fichiers structurés

# Encadrant :

BONNEVAY Stéphane

# Réalisé par :

SHABANI Eron

GUEZOULI Selmene

DHIF Mohamed-Azize

# Date :

02/05/2025

# Contexte et objectifs

Dans le cadre du module INFO4, il nous a été demandé de développer un programme capable de lire un fichier structuré représentant une base de données de patients et de fournir à l'utilisateur différentes informations issues de ce fichier. L’objectif était d'exploiter un format de fichier spécifique pour structurer des données, les parser, les manipuler, et les afficher de manière interactive, tout en respectant de bonnes pratiques de développement en Python.

# Fonctionnalités attendues

- Lecture d’un fichier de données contenant des fiches de patients.  
- Analyse syntaxique du fichier pour reconstruire des objets correspondant aux fiches.  
- Affichage des fiches patients, éventuellement de manière filtrée (par nom, pathologie, etc.).  
- Interaction utilisateur simple (via un menu textuel).  
- Conception modulaire et évolutive (favorisant l’usage de classes).

# Développement et organisation du travail

## Travail préparatoire – version initiale sans classes

Avant d'arriver à la version finale, nous avons développé une première version du programme sans utiliser de classes. Cette version procédurale nous a permis de bien comprendre la structure du fichier d’entrée, la logique de parsing, et les transformations nécessaires pour exploiter les données. Cependant, cette approche montrait vite ses limites :  
- Code peu modulaire et difficile à maintenir.  
- Complexité croissante avec l’ajout de nouvelles fonctionnalités.  
- Faible réutilisabilité.  
Cela nous a motivés à restructurer le programme autour d’une approche orientée objet.

## Version finale – approche objet

Nous avons ensuite refondu le projet en utilisant des classes Python. Deux fichiers structurent le programme :  
  
- lecture\_fichier.py : contient la classe principale de lecture et de traitement du fichier.  
 - La méthode \_\_init\_\_ permet d’initialiser les données.  
 - Une méthode lecture est chargée de lire le fichier ligne par ligne, en analysant les blocs patients.  
 - Les données sont stockées de manière structurée pour faciliter leur consultation ultérieure.  
  
- main.py : propose une interface textuelle à l’utilisateur pour interagir avec les données.  
 - Menu interactif en boucle (tant que l'utilisateur ne quitte pas).  
 - Possibilité d'afficher tous les patients, de filtrer par nom ou par maladie.  
 - Appels aux méthodes de lecture\_fichier.py pour traiter et afficher les résultats.

# Bilan

- Le projet respecte les exigences du sujet : le fichier est correctement lu, les informations sont extraites et affichées selon différents critères.  
- Le passage d’un code procédural à une architecture orientée objet a amélioré la clarté et la maintenabilité.  
- La répartition du travail s’est faite en groupe de manière équilibrée. Chacun a contribué à la compréhension du format, à l’implémentation, et aux tests.  
- La gestion des erreurs (par exemple fichier introuvable ou mal structuré) pourrait être améliorée dans une version future.  
- Le projet nous a permis de renforcer nos compétences en :  
 - Lecture et parsing de fichiers,  
 - Programmation orientée objet,  
 - Interaction utilisateur simple en console,  
 - Collaboration de groupe autour d’un objectif commun.

# Conclusion

Ce projet nous a permis d’appliquer les notions de programmation vues en cours dans un cadre concret. En partant d’un format brut de données, nous avons construit une application fonctionnelle, réutilisable et claire. L’évolution vers une version orientée objet a été un moment charnière dans le développement, et nous a permis de mieux structurer notre travail. Travailler en groupe nous a également permis de progresser sur le plan organisationnel et technique.