

Projet_1_corr

N'oubliez pas d'utiliser des commentaires pour expliquer le fonctionnement de chaque partie de votre code.

Projet

Créer un système de gestion de notes pour une école qui permet de stocker, manipuler et analyser les données liées aux étudiants, aux cours et aux résultats.

Gestion des étudiants :

Stockage des informations sur les étudiants (nom, prénom, numéro d'étudiant, etc.) dans un dictionnaire ou une liste de dictionnaires.

Ajout, suppression et mise à jour des informations des étudiants.

Gestion des cours :

Stockage des informations sur les cours (nom du cours, code du cours, professeur responsable, etc.) dans un dictionnaire ou une liste de dictionnaires.

Ajout, suppression et mise à jour des informations des cours.

Gestion des notes :

Utilisation de tableaux NumPy pour stocker les notes des étudiants pour chaque cours.

Calcul des moyennes, médianes, etc.

Création de rapports :

Utilisation de DataFrames (pandas) pour agréger et analyser les données.

Génération de rapports sur les performances des étudiants, les résultats par cours, etc.

Création de DataFrames avec pandas

```
students_df = pd.DataFrame(students)
```

```
courses_df = pd.DataFrame(courses)
```

```
grades_df = pd.DataFrame(grades, index=[s["id"] for s in students], columns=[c["code"] for c in courses])
```

Création de Graphes :

Ajouter un graphique qui représente visuellement certaines données liées aux étudiants ou aux cours

Correction :

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Données initiales
students = [
    {"id": 1, "nom": "Dupont", "prénom": "Jean", "numéro_d'étudiant": "ETU0001"},
    {"id": 2, "nom": "Leroy", "prénom": "Marie", "numéro_d'étudiant": "ETU0002"},
    {"id": 3, "nom": "Martin", "prénom": "Luc", "numéro_d'étudiant": "ETU0003"}
]

courses = [
    {"code": "MATH101", "nom": "Algèbre", "professeur_responsable": "M. Durand"},
    {"code": "PHYS101", "nom": "Mécanique", "professeur_responsable": "Mme. Dupuis"}
]

# Initialisation des notes
np.random.seed(0)
notes = np.random.randint(0, 21, size=(len(students), len(courses)))

# Création des DataFrames
students_df = pd.DataFrame(students)
courses_df = pd.DataFrame(courses)
grades_df = pd.DataFrame(notes, index=[s["id"] for s in students],
                           columns=[c["code"] for c in courses])

# Fonctions de gestion
def ajouter_etudiant(students, nouvel_etudiant):
    students.append(nouvel_etudiant)

def supprimer_etudiant(students, id_etudiant):
    students[:] = [etudiant for etudiant in students if etudiant["id"] != id_etudiant]

def mise_a_jour_etudiant(students, id_etudiant, nouvelles_infos):
    for etudiant in students:
        if etudiant["id"] == id_etudiant:
            etudiant.update(nouvelles_infos)
            break

def ajouter_cours(courses, nouveau_cours):
    courses.append(nouveau_cours)

def supprimer_cours(courses, code_cours):
    courses[:] = [cours for cours in courses if cours["code"] != code_cours]

def mise_a_jour_cours(courses, code_cours, nouvelles_infos):

```

```

    for cours in courses:
        if cours["code"] == code_cours:
            cours.update(nouvelles_infos)
            break

# Fonction d'exécution de tests
def executer_tests():
    # Test des fonctions de gestion des étudiants
    ajouter_etudiant(students, {"id": 4, "nom": "Moreau", "prénom": "Julie",
"numéro_d'étudiant": "ETU0004"})
    supprimer_etudiant(students, 3)
    mise_a_jour_etudiant(students, 2, {"nom": "Leroy", "prénom": "Marie-Claire"})

    # Affichage des étudiants après modifications
    print("Étudiants après modifications:")
    print(pd.DataFrame(students))

    # Calcul des moyennes et médianes
    moyennes_par_etudiant = grades_df.mean(axis=1)
    medianes_par_etudiant = grades_df.median(axis=1)

    # Affichage des moyennes et médianes
    print("Moyennes par étudiant:")
    print(moyennes_par_etudiant)
    print("Médianes par étudiant:")
    print(medianes_par_etudiant)

    # Création de graphiques
    plt.figure(figsize=(10, 6))
    plt.bar(students_df["numéro_d'étudiant"], moyennes_par_etudiant,
color='skyblue')
    plt.xlabel('Numéro d\étudiant')
    plt.ylabel('Moyenne des notes')
    plt.title('Moyenne des notes par étudiant')
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()

# Exécution des tests
executer_tests()

```

Bon travail