
- Tp10 r1.01 -

Maintenant, nous avons un souci d'efficacité dans le stockage des informations. En effet, un même auteur peut avoir écrit de nombreux livres. Jusqu'à cette semaine, lorsque dans la bibliothèque N livres avaient été écrits par le même auteur, les informations sur cet auteur étaient stockées N fois.

Nous voulons éviter cette redondance d'informations en ne stockant qu'une seule fois un auteur qui aurait écrit plusieurs livres.

L'économie ainsi réalisée va nous permettre d'ajouter des informations sur l'auteur, qui sera défini par :

- nom
- prénom
- nationalité
- année de naissance
- année (éventuelle) de décès

1- Ajout et modification des structures de données :

- proposer une structure **Auteur**, et ajouter au programme un *ensemble* des auteurs
- modifier la structure livre en intégrant la structure **Auteur**.
- ajouter un menu permettant d'ajouter un auteur

Proposez dans votre compte rendu une schématisation des liens entre ces deux structures de données.

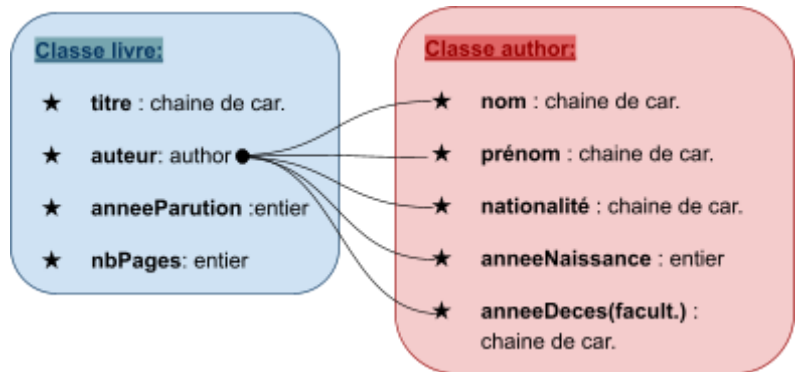
2- Réécrire les fonctionnalités de votre programme avec ces modifications.

En particulier, lors de l'ajout d'un livre, vous devrez présenter la liste des auteurs déjà connus pour que l'utilisateur puisse faire son choix.

1- Ajout et modification des structures de données

structure livre intégrant la classe author

```
class auteur:  
    nom: str  
    prenom: str  
    nationalite: str  
    dateNaissance: int  
    dateDecesFacultative: str  
  
class livre:  
    titre: str  
    auteur: auteur  
    anneeParution: int  
    nbPages: int
```



Ajout d'un auteur

Cette fonction permet d'ajouter une case dans le tableau contenant les auteurs, qui est similaire à un sous tableau contenant les informations demandées sur l'auteur écrites par l'utilisateur.

```
def ajout_auteur (tableau_auteurs: list[auteur], n:int):  
    """  
    fonction permettant d'ajouter un auteur à la bibliothèque  
    entrées: tableau_auteurs(list[auteur]) : tableau d'auteurs, n(int) : nombre d'auteurs dans la bibliothèque  
    sorties: n(int) : nombre d'auteurs dans la bibliothèque indenté de 1  
    """  
    nouvel_auteur = auteur() # Création d'un nouvel auteur  
    nouvel_auteur.nom = input("Entrez le nom de l'auteur: ")  
    nouvel_auteur.prenom = input("Entrez le prénom de l'auteur: ")  
    nouvel_auteur.nationalite = input("Entrez la nationalité de l'auteur: ")  
    nouvel_auteur.dateNaissance = int(input("Entrez l'annee de naissance de l'auteur: "))  
    nouvel_auteur.dateDecesFacultative = str(input("Entrez l'annee de décès de l'auteur(facultatif): ")) # L'année de décès est facultative  
    tableau_auteurs.append(nouvel_auteur) # Ajout de l'auteur au tableau  
    n+=1 # Incrémentation du nombre d'auteurs  
    return n
```

Cette procédure est un affichage qui affiche la liste des auteurs présents dans le tableau avec chacun un identifiant leur étant attribué correspondant à leur place dans le tableau +1 permettant à l'utilisateur de choisir de quel auteur il veut afficher les livre.

Si le tableau d'auteurs et ou le tableau de livres sont vides un message d'erreur est affiché.

```
def livres_d_un_auteur(table: list[livre], tableau_auteurs: list[auteur]):
    """
    procédure permettant d'afficher les livres d'un auteur
    entrées: table(list[livre]) : tableau de livres, tableau_auteurs(list[auteur]) : tableau d'auteurs
    sorties: aucune
    """
    if len(table) == 0: # Si le tableau est vide on affiche un message
        print("La bibliothèque est vide")
        return
    if len(tableau_auteurs) == 0: # Si le tableau d'auteurs est vide on affiche un message
        print("Aucun auteur disponible")
        return
    print("Liste des auteurs disponibles:")
    for idx, auteur in enumerate(tableau_auteurs):
        print(f"{idx + 1}. {auteur.nom} {auteur.prenom}")
    choix_auteur = int(input("Choisissez un auteur par son numéro: ")) - 1
    if choix_auteur < 0 or choix_auteur >= len(tableau_auteurs):
        print("Choix invalide.")
        return
    auteur = tableau_auteurs[choix_auteur]
    print(f"Livres de {auteur.nom} {auteur.prenom}:")
    for livre in table:
        if livre.author == auteur:
            print(f"Titre: {livre.titre}")
            print(f"Année de parution: {livre.anneeParution}")
            print(f"Nombre de pages: {livre.nbPages}")
            print("-----")
```

pseudo code:

livresDUnAuteur(biblio: tab[livre]:100, tabAuteurs : tab[author], nAuthor, nLivre) → retourne rien

début

avec compteur_i : entier

compteur_j : entier

choix : entier

aut :author

si (nLivre = 0) alors

afficher "la bibliothèque de livres est vide"

retourne

si (nAuthor = 0) alors

afficher "la bibliothèque d'auteurs est vide"

retourne

afficher " Liste des auteurs disponibles: "

Pour (compteur i de 0 a nAuthor-1) faire

afficher "id: ", compteur_i +1 , " ", author.nom , " ", author.prénom

afficher "choisir l'id de votre auteur: "

saisir choix

#vérifier si choix dans liste sinon retourne

aut ← tabAuteurs[choix]

afficher "livres de ", author.nom , " ", author.prénom, " : "

à la ligne

pour (compteur_j de 0 à nLivre-1) faire

si biblio[compteur_j] = aut

#afficher le livre

Ensemble des auteurs

Cette procédure permet d'afficher tous les auteurs présents dans le tableau d'auteurs permettant d'avoir une vue d'ensemble à l'utilisateur des auteurs déjà entrés dans son tableau. La procédure affiche un message d'erreur à l'utilisateur si le tableau d'auteur est vide

```
def tousLesAuteurs(tableau_auteurs:list[auteur]):  
    """  
    procédure permettant d'afficher tous les auteurs  
    entrées: tableau_auteurs(list[auteur]) : tableau d'auteurs  
    sorties: aucune  
    """  
  
    if len(tableau_auteurs) == 0: # Si le tableau est vide on affiche un message  
        print("Aucun auteur disponible")  
        return  
    for auteur in tableau_auteurs: # Parcourt le tableau et affiche les auteurs  
        print("Nom: ", auteur.nom)  
        print("Prénom: ", auteur.prenom)  
        print("Nationalité: ", auteur.nationalite)  
        print("Année de naissance: ", auteur.dateNaissance)  
        print("Année de décès: ", auteur.dateDecesFacultative)  
        print("-----") # Ligne vide pour séparer les auteurs
```

2- Réécriture des fonctionnalités de mon programme

modification de mon menu

Je rajoute à mon menu la possibilité d'ajouter un auteur, d'afficher les livres d'un auteur et d'afficher tous les auteurs entrés.

```
menu_options = {  
    1 : "Afficher l'ensemble des livres de la bibliothèque",  
    2 : "Ajouter un nouveau livre",  
    3 : "Rechercher un livre (par le titre)",  
    4 : "afficher les livres d'un auteur",  
    5 : "ajouter un auteur",  
    6 : "Afficher tous les auteurs",  
    7 : "Quitter"  
}
```

Déclaration d'un nouveau tableau étant le tableau d'auteurs

```
if __name__ == '__main__':  
    table: list[livre] = [] # Déclaration du tableau de livres  
    tableau_auteurs: list[auteur] = [] # Déclaration du tableau d'auteurs  
    n:int  
    n_auteur:int  
    n_auteur=0  
    n=0  
    place: int  
    option = str # Déclaration de la variable pour l'option choisie
```

Nouvelle boucle de menu avec les nouvelles options.

```
if option == '1':  
    afficher_livres(table,n)  
elif option == '2':  
    n=ajouter_livre(table,n,tableau_auteurs)  
elif option == '3':  
    place=rechercher_livre(table)  
    if place == -1:  
        print("Le livre n'est pas dans la bibliothèque")  
    elif place == -2:  
        pass  
    else:  
        print("Le livre se trouve à la place ", place+1 )  
  
elif option == '4':  
    livres_d_un_auteur(table,tableau_auteurs)  
  
elif option == '5':  
    n_auteur=ajout_auteur(tableau_auteurs,n_auteur)  
  
elif option == '6':  
    tousLesAuteurs(tableau_auteurs)  
elif option == '7':  
    break
```

modification procédure ajouter livre

La fonction ajouter livre est modifiée dans le but de maintenant afficher le tableau d'auteurs à l'utilisateur permettant à ce dernier de choisir parmi la liste et d'ensuite ajouter les éléments reliés au livre.

Si le tableau d'auteurs est vide, un message d'erreur est retourné à la place.

```
def ajouter_livre(table: list[livre], n:int, tableau_auteurs: list[auteur]):
    """
    fonction permettant d'ajouter un livre à la bibliothèque
    entrées: table(list[livre]) : tableau de livres, n(int) : nombre de livres d
    sorties: n(int) : nombre de livres dans la bibliothèque indenté de 1
    """
    if len(tableau_auteurs) == 0:
        print("Aucun auteur disponible. Veuillez ajouter un auteur d'abord.")
        return n

    nouveau_livre = livre() # Création d'un nouveau livre
    nouveau_livre.titre = input("Entrez le titre du livre: ")
    if len(tableau_auteurs) == 0:
        print("Aucun auteur disponible. Veuillez ajouter un auteur d'abord.")
        return n

    print("Liste des auteurs disponibles:")
    for idx, auteur in enumerate(tableau_auteurs):
        print(f"{idx + 1}. {auteur.nom} {auteur.prenom}")

    choix_auteur = int(input("Choisissez un auteur par son numéro: ")) - 1
    if choix_auteur < 0 or choix_auteur >= len(tableau_auteurs):
        print("Choix invalide.")
        return n

    nouveau_livre.author = tableau_auteurs[choix_auteur]
    nouveau_livre.anneeParution = int(input("Entrez l'année de parution: "))
    nouveau_livre.nbPages = int(input("Entrez le nombre de pages: "))
    table.append(nouveau_livre) # Ajout du livre au tableau
    n+=1 # Incrémentement du nombre de livres
    return n
```

code complet

Fonctions/procédures

```
class auteur:
    nom: str
    prenom: str
    nationalite: str
    dateNaissance: int
    dateDecesFacultative: str

class livre:
    titre: str
    author: auteur
    anneeParution: int
    nbPages: int

menu_options = {
    1 : "Afficher l'ensemble des livres de la bibliothèque",
    2 : "Ajouter un nouveau livre",
    3 : "Rechercher un livre (par le titre)",
    4 : "afficher les livres d'un auteur",
    5 : "ajouter un auteur",
    6 : "Afficher tous les auteurs",
    7 : "Quitter"
}

def print_menu():
    """
    Affiche le menu des options
    entrées: aucune
    sorties: aucune
    """
    for key in menu_options.keys():
        print(key, '-- ', menu_options[key])
```

```
def afficher_livres(table: list[livre], n: int):
    """
    procédure permettant d'afficher les livres de la bibliothèque de livres renvoie un message d'erreur si le tableau est vide
    entrées: table(list[livre]) : tableau de livres, n(int) : nombre de livres dans la bibliothèque
    sorties: aucune
    """
    if n == 0: # Si le tableau est vide on affiche un message
        print("La bibliothèque est vide")
        return
    for livre in table: # Parcours le tableau et affiche les livres
        print("Titre: ", livre.titre)
        print("Auteur: ", livre.author.nom)
        print("Année de parution: ", livre.anneeParution)
        print("Nombre de pages: ", livre.nbPages)
        print("-----") # Ligne vide pour séparer les livres
```

```
def ajouter_livre(table: list[livre], n: int, tableau_auteurs: list[auteur]):
    """
    fonction permettant d'ajouter un livre à la bibliothèque
    entrées: table(list[livre]) : tableau de livres, n(int) : nombre de livres dans la bibliothèque
    sorties: n(int) : nombre de livres dans la bibliothèque indenté de 1
    """
    if len(tableau_auteurs) == 0:
        print("Aucun auteur disponible. Veuillez ajouter un auteur d'abord.")
        return n

    nouveau_livre = livre() # Création d'un nouveau livre
    nouveau_livre.titre = input("Entrez le titre du livre: ")
    if len(tableau_auteurs) == 0:
        print("Aucun auteur disponible. Veuillez ajouter un auteur d'abord.")
        return n

    print("Liste des auteurs disponibles:")
    for idx, auteur in enumerate(tableau_auteurs):
        print(f"{idx + 1}. {auteur.nom} {auteur.prenom}")

    choix_auteur = int(input("Choisissez un auteur par son numéro: ")) - 1
    if choix_auteur < 0 or choix_auteur >= len(tableau_auteurs):
        print("Choix invalide.")
        return n

    nouveau_livre.author = tableau_auteurs[choix_auteur]
    nouveau_livre.anneeParution = int(input("Entrez l'année de parution: "))
    nouveau_livre.nbPages = int(input("Entrez le nombre de pages: "))
    table.append(nouveau_livre) # Ajout du livre au tableau
    n+=1 # Incrémentation du nombre de livres
    return n
```



```

def rechercher_livre(table: list[livre]):
    """
    fonction permettant de rechercher un livre dans la bibliothèque par son titre
    entrées: table(list[livre]) : tableau de livres
    sorties: id(int) : position du livre dans le tableau
    """
    if len(table) == 0: # Si le tableau est vide on affiche un message
        print("La bibliothèque est vide")
        return -2
    titre = input("Entrez le titre du livre à rechercher: ") # Demande le titre du livre à rechercher
    for id in range(len(table)): # Parcours le tableau
        if table[id].titre == titre: # Si le titre du livre est trouvé on affiche le livre
            print("Titre: ", table[id].titre)
            print("Auteur: ", table[id].author.nom)
            print("Année de parution: ", table[id].anneeParution)
            print("Nombre de pages: ", table[id].nbPages)
            print("-----")
            return id
    return -1 # Si le livre n'est pas trouvé, retourne -1

def ajout_auteur (tableau_auteurs: list[auteur], n:int):
    """
    fonction permettant d'ajouter un auteur à la bibliothèque
    entrées: tableau_auteurs(list[auteur]) : tableau d'auteurs, n(int) : nombre d'auteurs dans la bibliothèque
    sorties: n(int) : nombre d'auteurs dans la bibliothèque indenté de 1
    """
    nouvel_auteur = auteur() # Création d'un nouvel auteur
    nouvel_auteur.nom = input("Entrez le nom de l'auteur: ")
    nouvel_auteur.prenom = input("Entrez le prénom de l'auteur: ")
    nouvel_auteur.nationalite = input("Entrez la nationalité de l'auteur: ")
    nouvel_auteur.dateNaissance = int(input("Entrez l'annee de naissance de l'auteur: "))
    nouvel_auteur.dateDecesFacultative = str(input("Entrez l'annee de décès de l'auteur(facultatif): ")) # L'année de décès est facultative
    tableau_auteurs.append(nouvel_auteur) # Ajout de l'auteur au tableau
    n+=1 # Incrémentation du nombre d'auteurs
    return n

```

```

def livres_d_un_auteur(table: list[livre], tableau_auteurs: list[auteur]):
    """
    procédure permettant d'afficher les livres d'un auteur
    entrées: table(list[livre]) : tableau de livres, tableau_auteurs(list[auteur]) : tableau d'auteurs
    sorties: aucune
    """
    if len(table) == 0: # Si le tableau est vide on affiche un message
        print("La bibliothèque est vide")
        return
    if len(tableau_auteurs) == 0: # Si le tableau d'auteurs est vide on affiche un message
        print("Aucun auteur disponible")
        return
    print("Liste des auteurs disponibles:")
    for idx, auteur in enumerate(tableau_auteurs):
        print(f"{idx + 1}. {auteur.nom} {auteur.prenom}") #on imprime tout les auteurs avec leurs place dans le tableau +1
    choix_auteur = int(input("Choisissez un auteur par son numéro: ")) - 1 #on retire 1 à la réponse pour permettre d'avoir le bon identifiant
    if choix_auteur < 0 or choix_auteur >= len(tableau_auteurs): #gestion des erreurs
        print("Choix invalide.")
        return
    auteur = tableau_auteurs[choix_auteur]
    print(f"Livres de {auteur.nom} {auteur.prenom}:")
    for livre in table:
        if livre.author == auteur:
            print(f"Titre: {livre.titre}")
            print(f"Année de parution: {livre.anneeParution}")
            print(f"Nombre de pages: {livre.nbPages}")
            print("-----")

```

```

def tousLesAuteurs(tableau_auteurs:list[auteur]):
    """
    procédure permettant d'afficher tous les auteurs
    entrées: tableau_auteurs(list[auteur]) : tableau d'auteurs
    sorties: aucune
    """
    if len(tableau_auteurs) == 0: # Si le tableau est vide on affiche un message
        print("Aucun auteur disponible")
        return
    for auteur in tableau_auteurs: # Parcourt le tableau et affiche les auteurs
        print("Nom: ", auteur.nom)
        print("Prénom: ", auteur.prenom)
        print("Nationalité: ", auteur.nationalite)
        print("Année de naissance: ", auteur.dateNaissance)
        print("Année de décès: ", auteur.dateDecesFacultative)
        print("-----") # Ligne vide pour séparer les auteurs

```

programme squelette

```

if __name__ == '__main__':
    table: list[livre] = [] # Déclaration du tableau de livres
    tableau_auteurs: list[auteur] = [] # Déclaration du tableau d'auteurs
    n:int
    n_auteur:int
    n_auteur=0
    n=0
    place: int
    option = str # Déclaration de la variable pour l'option choisie

```

```

while True:

    print_menu() # Affiche le menu
    option = ''
    try:
        option = input("Entrez votre choix: ") # Lit le choix de l'utilisateur
    except:
        print("Entrez un nombre valide")

    if option == '1':
        afficher_livres(table,n)
    elif option == '2':
        n=ajouter_livre(table,n,tableau_auteurs)
    elif option == '3':
        place=rechercher_livre(table)
        if place == -1:
            print("Le livre n'est pas dans la bibliothèque")
        elif place == -2:
            pass
        else:
            print("Le livre se trouve à la place ", place+1 )

    elif option == '4':
        livres_d_un_auteur(table,tableau_auteurs)

    elif option == '5':
        n_auteur=ajout_auteur(tableau_auteurs,n_auteur)

    elif option == '6':
        tousLesAuteurs(tableau_auteurs)
    elif option == '7':
        break

```