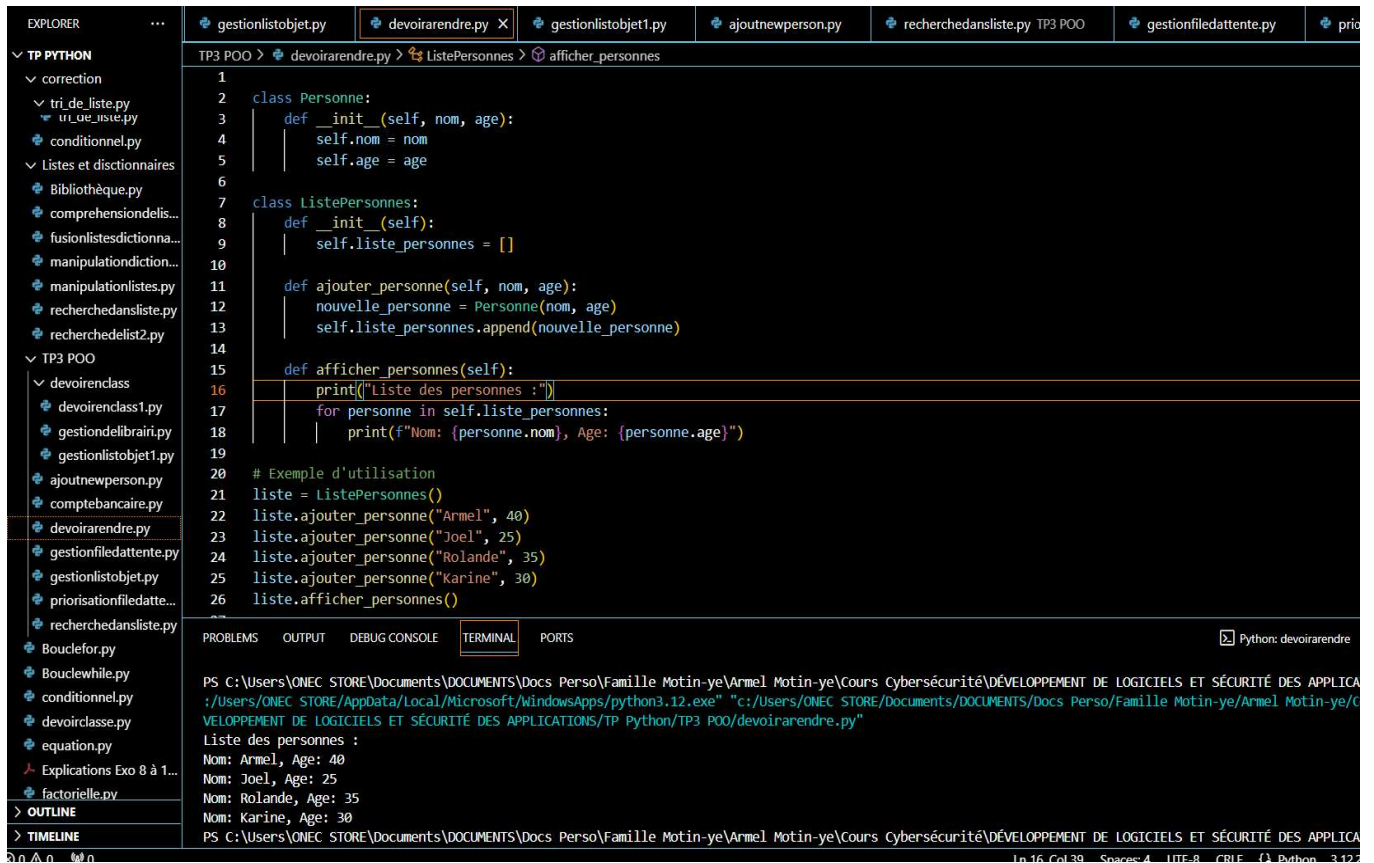


DEVOIR DE GESTION D'UNE LISTE D'OBJET

ARMEL JOEL MOTIN-YE

Gestion d'une liste d'objets :

- Création de la classe ListePersonnes qui contient une liste d'objets de type Personne.
- Ajout d'une méthode ajouter_personne(nom, age) pour ajouter une nouvelle personne à la liste.
- Ajout d'une méthode afficher_personnes() pour afficher les détails de toutes les personnes dans la liste.



```
1 class Personne:
2     def __init__(self, nom, age):
3         self.nom = nom
4         self.age = age
5
6
7 class ListePersonnes:
8     def __init__(self):
9         self.liste_personnes = []
10
11     def ajouter_personne(self, nom, age):
12         nouvelle_personne = Personne(nom, age)
13         self.liste_personnes.append(nouvelle_personne)
14
15     def afficher_personnes(self):
16         print("Liste des personnes :")
17         for personne in self.liste_personnes:
18             print(f"Nom: {personne.nom}, Age: {personne.age}")
19
20 # Exemple d'utilisation
21 liste = ListePersonnes()
22 liste.ajouter_personne("Armel", 40)
23 liste.ajouter_personne("Joel", 25)
24 liste.ajouter_personne("Rolande", 35)
25 liste.ajouter_personne("Karine", 30)
26 liste.afficher_personnes()
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS Python: devoirarendre

```
PS C:\Users\ONEC STORE\Documents\DOCUMENTS\Docs Perso\Famille Motin-ye\Armel Motin-ye\Cours Cybersécurité\DÉVELOPPEMENT DE LOGICIELS ET SÉCURITÉ DES APPLICA
./Users/ONEC STORE/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/ONEC STORE/Documents/DOCUMENTS/Docs Perso/Famille Motin-ye/Armel Motin-ye/C
VELOPPEMENT DE LOGICIELS ET SÉCURITÉ DES APPLICATIONS/TP Python/TP3 POO/devoirarendre.py"
Liste des personnes :
Nom: Armel, Age: 40
Nom: Joel, Age: 25
Nom: Rolande, Age: 35
Nom: Karine, Age: 30
PS C:\Users\ONEC STORE\Documents\DOCUMENTS\Docs Perso\Famille Motin-ye\Armel Motin-ye\Cours Cybersécurité\DÉVELOPPEMENT DE LOGICIELS ET SÉCURITÉ DES APPLICA
```

- Recherche dans une liste d'objets
- Filtrage des personnes par âge
- Gestion d'une file d'attente
- Priorisation dans la file d'attente

```

11 class ListePersonnes:
24
25     def rechercher_personne(self, nom):
26         found = False
27         for personne in self.personnes:
28             if personne.nom == nom:
29                 personne.afficher_details()
30                 found = True
31         if not found:
32             print("Personne non trouvée.")
33
34     def filtrer_personnes_par_age(self, min_age, max_age):
35         print(f"Personnes entre {min_age} et {max_age} ans:")
36         for personne in self.personnes:
37             if min_age <= personne.age <= max_age:
38                 personne.afficher_details()
39
40 class FileAttente:
41     def __init__(self):
42         self.attente = []
43
44     def ajouter_personne_en_attente(self, nom):
45         self.attente.append(nom)
46
47     def supprimer_personne_de_attente(self):
48         if self.attente:
49             print(f"La personne {self.attente.pop(0)} a été supprimée de la file d'attente.")
50         else:
51             print("La file d'attente est vide.")
52
53 class FileAttentePrioritaire(FileAttente):
54     def __init__(self):
55         super().__init__()
56         self.prioritaires = []
57
58     def ajouter_personne_prioritaire(self, nom):
59         self.prioritaires.append(nom)
60
61     def supprimer_personne_de_attente(self):
62         if self.prioritaires:
63             print(f"La personne prioritaire {self.prioritaires.pop(0)} a été supprimée de la file d'attente.")
64         elif self.attente:
65             print(f"La personne {self.attente.pop(0)} a été supprimée de la file d'attente.")
66         else:
67             print("La file d'attente est vide.")
68
69
70

```

Suite personnalisation : avec la création des menus déroulants pour plus de fluidité dans l'utilisation

TP PYTHON

correction

tri_de_liste.py

conditionnel.py

Listes et dictionnaires

Bibliothèque.py

comprehensiondelis...

fusionlistesdictionna...

manipulationdiction...

manipulationlistes.py

recherchedansliste.py

recherchedelist2.py

TP3 POO

devoirenclass

devoirenclass1.py

gestiondelibrari.py

gestionlistobjet1.py

ajoutnewperson.py

comptebancaire.py

devoirarendre.py

gestionfileattente.py

gestionlistobjet.py

priorisationfiledatte...

recherchedansliste.py

Bouclefor.py

Bouclewhile.py

conditionnel.py

devoirclasse.py

equation.py

Explications Exo 8 à 1...

factorielle.py

moi.py

moyenne.py

parite.py

PGCD.py

presenceelement.py

primalite.py

produit.py

somme.py

temperature.py

TP dictionnaires et list...

TP1.py

Tri_de_liste.py

trideliste.py

OUTLINE

TIMELINE

TP3 POO > devoirenclass > gestionlistobjet1.py > ListePersonnes

54 class FileAttentePrioritaire(FileAttente):

62 def supprimer_personne_de_attente(self):

67 else:

68 print("La file d'attente est vide.")

69

70

71 def menu_principal():

72 print("Menu Principal:")

73 print("1. Gérer Liste de Personnes")

74 print("2. Gérer File d'Attente")

75 print("3. Quitter")

76 choix = input("Entrez votre choix : ")

77 return choix

78

79

80 def menu_gestion_personnes(liste_personnes):

81 while True:

82 print("\nMenu Gestion de Liste de Personnes:")

83 print("1. Ajouter une personne")

84 print("2. Afficher la liste de personnes")

85 print("3. Rechercher une personne")

86 print("4. Filtrer les personnes par âge")

87 print("5. Retour")

88 choix = input("Entrez votre choix : ")

89 if choix == "1":

90 nom = input("Entrez le nom de la personne : ")

91 age = int(input("Entrez l'âge de la personne : "))

92 liste_personnes.ajouter_personne(nom, age)

93 elif choix == "2":

94 liste_personnes.afficher_personnes()

95 elif choix == "3":

96 nom = input("Entrez le nom de la personne à rechercher : ")

97 liste_personnes.rechercher_personne(nom)

98 elif choix == "4":

99 min_age = int(input("Entrez l'âge minimum : "))

100 max_age = int(input("Entrez l'âge maximum : "))

101 liste_personnes.filtrer_personnes_par_age(min_age, max_age)

102 elif choix == "5":

103 break

104 else:

105 print("Choix invalide.")

106

107

108 def menu_gestion_file_attente(file_attente):

109 while True:

110 print("\nMenu Gestion de File d'Attente:")

111 print("1. Ajouter une personne en attente")

112 print("2. Supprimer une personne de la file d'attente")

113 print("3. Ajouter une personne prioritaire en attente")

114 print("4. Retour")

115 choix = input("Entrez votre choix : ")

116 if choix == "1":

117 nom = input("Entrez le nom de la personne à ajouter en attente : ")

118 file_attente.ajouter_personne_en_attente(nom)

119 elif choix == "2":

EXPLORER

devoirarendre.py

gestionlistobjet1.py

ajoutnewperson.py

recherchedansliste.py

TP3 POO

devoirenclass

devoirenclass1.py

gestiondelibrari.py

gestionlistobjet1.py

ajoutnewperson.py

comptebancaire.py

devoirarendre.py

gestionfileattente.py

gestionlistobjet.py

priorisationfiledatte...

recherchedansliste.py

Bouclefor.py

Bouclewhile.py

conditionnel.py

devoirclasse.py

equation.py

Explications Exo 8 à 1...

factorielle.py

moi.py

moyenne.py

parite.py

PGCD.py

presenceelement.py

primalite.py

TP3 POO > devoirenclass > gestionlistobjet1.py > ListePersonnes

107

108 def menu_gestion_file_attente(file_attente):

109 while True:

110 print("\nMenu Gestion de File d'Attente:")

111 print("1. Ajouter une personne en attente")

112 print("2. Supprimer une personne de la file d'attente")

113 print("3. Ajouter une personne prioritaire en attente")

114 print("4. Retour")

115 choix = input("Entrez votre choix : ")

116 if choix == "1":

117 nom = input("Entrez le nom de la personne à ajouter en attente : ")

118 file_attente.ajouter_personne_en_attente(nom)

119 elif choix == "2":

120 file_attente.supprimer_personne_de_attente()

121 elif choix == "3":

122 nom = input("Entrez le nom de la personne prioritaire à ajouter en attente : ")

123 file_attente.ajouter_personne_prioritaire(nom)

124 elif choix == "4":

125 break

126 else:

127 print("Choix invalide.")

128

129

130 def main():

131 liste_personnes = ListePersonnes()

132 file_attente = FileAttentePrioritaire()

133

134 while True:

135 choix = menu_principal()

136 if choix == "1":

137 menu_gestion_personnes(liste_personnes)

138 elif choix == "2":

139 menu_gestion_file_attente(file_attente)

140 elif choix == "3":

141 print("Au revoir !")

142 break

143 else:

144 print("Choix invalide.")

145

146

147 if __name__ == "__main__":

148 main()

Affichage de Résultat :

irection
1.oe.isee.py
onditionnel.py
tes et dictionnaires
ibliothèque.py
omprehensiondelis...
usionlistesdictionna...
manipulationdiction...
manipulationlistes.py
echedansliste.py
echedelist2.py
B POO
evoirenclass
Cascinema.py
Cascinema1.py
devoirenclass1.py
gestiondelibraini.py
gestionlistobjet1.py
joutnewperson.py
omptebancaire.py
evoirarendre.py
estionfiledattente.py
estionlistobjetpy
riorisationfiledatte...
echedansliste.py
uclefor.py
uclewhile.py
nditionnel.py
voirclasse.py
uation.py
plications. Exo 8 à 1...
INE

```
130 def main():
136     if choix == "1":
137         menu_gestion_personnes(liste_personnes)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

Python: gestionlistobjet1

PS C:\Users\ONEC STORE\Documents\DOCUMENTS\Docs Perso\Famille Motin-ye\Armel Motin-ye\Cours Cybersécurité\DÉVELOPPEMENT DE LOGICIELS ET SÉCURITÉ DES APPLIC...
:/Users/ONEC STORE/AppData/Local/Microsoft/windowsApps/python3.12.exe" "c:/Users/ONEC STORE/Documents/DOCUMENTS/Docs Perso/Famille Motin-ye/Armel Motin-ye/C...
VELOPPEMENT DE LOGICIELS ET SÉCURITÉ DES APPLICATIONS/TP Python/TP3 POO/devoirenclass/gestionlistobjet1.py"
Menu Principal:
1. Gérer Liste de Personnes
2. Gérer File d'Attente
3. Quitter
Entrez votre choix : 1

Menu Gestion de Liste de Personnes:
1. Ajouter une personne
2. Afficher la liste de personnes
3. Rechercher une personne
4. Filtrer les personnes par âge
5. Retour
Entrez votre choix : 5
Menu Principal:
1. Gérer Liste de Personnes
2. Gérer File d'Attente
3. Quitter
Entrez votre choix : 2

Menu Gestion de File d'Attente:
1. Ajouter une personne en attente
2. Supprimer une personne de la file d'attente
3. Ajouter une personne prioritaire en attente
4. Retour
Entrez votre choix : █

CAS CINÉMA

- Simulation d'un système de réservation :

- Création d'une classe SalleCinema pour gérer les réservations dans une salle de cinéma.
- Ajout d'une méthode reserver_place(nom, place) pour réserver une place pour une personne.
- Ajout d'une méthode afficher_places_reservees() pour afficher les places réservées.

```
1 class SalleCinema:
2     def __init__(self, nb_places):
3         self.nb_places = nb_places
4         self.places_reservees = {}
5
6     def reserver_place(self, nom, place):
7         if place <= self.nb_places and place > 0:
8             if place not in self.places_reservees:
9                 self.places_reservees[place] = nom
10                print(f"Place {place} réservée pour {nom}.")
11            else:
12                print(f"La place {place} est déjà réservée.")
13        else:
14            print(f"La place {place} n'existe pas dans cette salle.")
15
16    def afficher_places_reservees(self):
17        if self.places_reservees:
18            print("Places réservées :")
19            for place, nom in self.places_reservees.items():
20                print(f"Place {place} : {nom}")
21        else:
22            print("Aucune place n'a été réservée.")
23
24    # Exemple d'utilisation
25    salle = SalleCinema(50)
26
27    salle.reserver_place("Armel", 5)
28    salle.reserver_place("Joel", 10)
29    salle.reserver_place("Rolande", 15)
30    salle.reserver_place("Karine", 21)
31
32    salle.afficher_places_reservees()
```

```
PS C:\Users\ONEC STORE\Documents\DOCUMENTS\Docs Perso\Famille Motin-ye\Armel Motin-ye\Cours Cybersécurité\DÉVELOPPEMENT DE LOGICIELS> python TP3 POO\devoirenclasse\Cascinema.py
Places réservées :
Place 5 : Armel
Place 10 : Joel
Place 15 : Rolande
Place 21 : Karine
PS C:\Users\ONEC STORE\Documents\DOCUMENTS\Docs Perso\Famille Motin-ye\Armel Motin-ye\Cours Cybersécurité\DÉVELOPPEMENT DE LOGICIELS>
```

- Gestion de la capacité de la salle
- Filtrage des réservations par personne
- Annulation des réservations
- Gestion des places spéciales

```

EXPLORER ...
TP PYTHON
  ✓ correction
  ✓ ur_de_liste.py
  ✓ conditionnel.py
  ✓ Listes et dictionnaires
  ✓ Bibliothèque.py
  ✓ comprehensiondelis...
  ✓ fusionlistesdictionna...
  ✓ manipulationdiction...
  ✓ manipulationlistes.py
  ✓ recherchedansliste.py
  ✓ recherchedelist2.py
  ✓ TP3 POO
    ✓ devoirenclass
      ✓ Cascinema.py
      ✓ Cascinema1.py
      ✓ devoirenclass1.py
      ✓ gestiondelibrairi.py
      ✓ gestionlistobjet1.py
      ✓ ajoutnewperson.py
      ✓ comptebancaire.py
      ✓ devoirarendre.py
      ✓ gestionfiledattente.py
      ✓ gestionlistobjet.py
      ✓ priorisationfiledatte...
      ✓ recherchedansliste.py
      ✓ Bouclefor.py
      ✓ Bouclewhile.py
      ✓ conditionnel.py
      ✓ devoirclasse.py
      ✓ equation.py
      ✓ Explications Exo 8 à 1...
  26

TP3 POO > devoirenclass > Cascinema1.py > ...
1 class SalleCinema:
2     def __init__(self, nb_places, nb_places_speciales):
3         self.nb_places = nb_places
4         self.nb_places_speciales = nb_places_speciales
5         self.places_reservees = {}
6         self.places_speciales_reservees = {}
7
8     def nombre_places_disponibles(self):
9         return self.nb_places - len(self.places_reservees)
10
11     def reserver_place(self, nom, place):
12         if place <= self.nb_places and place > 0:
13             if place not in self.places_reservees:
14                 self.places_reservees[place] = nom
15                 print(f"Place {place} réservée pour {nom}.")
16             else:
17                 print(f"La place {place} est déjà réservée.")
18         else:
19             print(f"La place {place} n'existe pas dans cette salle.")
20
21     def reserver_place_speciale(self, nom):
22         if len(self.places_speciales_reservees) < self.nb_places_speciales:
23             place_speciale = self.nb_places + len(self.places_speciales_reservees) + 1
24             self.places_speciales_reservees[place_speciale] = nom
25             print(f"Place spéciale réservée pour {nom}.")
26         else:
27             print("Toutes les places spéciales sont déjà réservées.")
28
29     def afficher_places_reservees(self):
30         if self.places_reservees:
31             print("Places réservées :")
32             for place, nom in self.places_reservees.items():
33                 print(f"Place {place} : {nom}")
34         else:
35             print("Aucune place n'a été réservée.")
36

```

```

EXPLORER ...
TP PYTHON
  ✓ correction
  ✓ ur_de_liste.py
  ✓ conditionnel.py
  ✓ Listes et dictionnaires
  ✓ Bibliothèque.py
  ✓ comprehensiondelis...
  ✓ fusionlistesdictionna...
  ✓ manipulationdiction...
  ✓ manipulationlistes.py
  ✓ recherchedansliste.py
  ✓ recherchedelist2.py
  ✓ TP3 POO
    ✓ devoirenclass
      ✓ Cascinema.py
      ✓ Cascinema1.py
      ✓ devoirenclass1.py
      ✓ gestiondelibrairi.py
      ✓ gestionlistobjet1.py
      ✓ ajoutnewperson.py
      ✓ comptebancaire.py
      ✓ devoirarendre.py
      ✓ gestionfiledattente.py
      ✓ gestionlistobjet.py
      ✓ priorisationfiledatte...
      ✓ recherchedansliste.py
      ✓ Bouclefor.py
      ✓ Bouclewhile.py
      ✓ conditionnel.py
      ✓ devoirclasse.py
      ✓ equation.py
      ✓ Explications Exo 8 à 1...
  70
  > OUTLINE
  > TIMELINE

TP3 POO > devoirenclass > Cascinema1.py > ...
1 class SalleCinema:
29     def afficher_places_reservees(self):
37         if self.places_speciales_reservees:
38             print("\nPlaces spéciales réservées :")
39             for place, nom in self.places_speciales_reservees.items():
40                 print(f"Place spéciale {place} : {nom}")
41
42     def filtrer_reservations_par_personne(self, nom):
43         reservations = [(place, nom_personne) for place, nom_personne in self.places_reservees.items() if nom_personne == nom]
44         if reservations:
45             print(f"Réservations pour {nom} :")
46             for place, nom_personne in reservations:
47                 print(f"Place {place}")
48         else:
49             print(f"Aucune réservation trouvée pour {nom}.")
50
51     def annuler_reservation(self, nom):
52         reservations_a_annuler = [place for place, nom_personne in self.places_reservees.items() if nom_personne == nom]
53         if reservations_a_annuler:
54             for place in reservations_a_annuler:
55                 del self.places_reservees[place]
56                 print(f"Réservations annulées pour {nom}.")
57         else:
58             print(f"Aucune réservation trouvée pour {nom}.")
59
60 # Fonction pour afficher le menu principal
61 def afficher_menu_principal():
62     print("\nMenu Principal :")
63     print("1. Réserver une place")
64     print("2. Réserver une place spéciale")
65     print("3. Afficher les places réservées")
66     print("4. Vérifier le nombre de places disponibles")
67     print("5. Filtrer les réservations par personne")
68     print("6. Annuler des réservations")
69     print("7. Quitter")
70
71 # Exécution du programme

```

EXPLORER

TP PYTHON

correction

un_ue_liste.py

conditionnel.py

Listes et dictionnaires

Bibliothèque.py

comprehensiondelis...

fusionlistesdictionna...

manipulationdiction...

manipulationlistes.py

recherchedansliste.py

recherchedelist2.py

TP3 POO

devoirenclass

Cascinema.py

Cascinema1.py

devoirenclass1.py

gestiondelibrari.py

gestionlistobjet1.py

ajoutnewperson.py

comptebancaire.py

devoirarendre.py

gestionfileattente.py

gestionlistobjet.py

priorisationfileatte...

recherchedansliste.py

Bouclefor.py

Bouclewhile.py

conditionnel.py

devoirclasse.py

equation.py

Explications Exo 8 à 1...

OUTLINE

devoirarendre.py

gestionlistobjet1.py

Cascinema.py

Cascinema1.py X

ajoutnewperson.py

reche...

TP3 POO > devoirenclass > Cascinema1.py > ...

72

salle = SalleCinema(50, 5)

73

74

while True:

75

afficher_menu_principal()

76

choix = input("Entrez votre choix : ")

77

78

if choix == "1":

79

nom = input("Entrez votre nom : ")

80

place = int(input("Entrez le numéro de la place : "))

81

salle.reserver_place(nom, place)

82

83

elif choix == "2":

84

nom = input("Entrez votre nom : ")

85

salle.reserver_place_speciale(nom)

86

87

elif choix == "3":

88

salle.afficher_places_reservees()

89

90

elif choix == "4":

91

print("Nombre de places disponibles :", salle.nombre_places_disponibles())

92

93

elif choix == "5":

94

nom = input("Entrez le nom de la personne : ")

95

salle.filtrer_reservations_par_personne(nom)

96

97

elif choix == "6":

98

nom = input("Entrez le nom de la personne dont vous voulez annuler les réservations : ")

99

salle.annuler_reservation(nom)

100

101

elif choix == "7":

102

print("Au revoir !")

103

break

104

105

else:

106

print("Choix invalide. Veuillez entrer un nombre entre 1 et 7.")

107

Affichage du résultat :

recherchedelist2.py

TP3 POO

devoirenclass

Cascinema.py

Cascinema1.py

devoirenclass1.py

gestiondelibrari.py

gestionlistobjet1.py

ajoutnewperson.py

comptebancaire.py

devoirarendre.py

gestionfileattente.py

gestionlistobjet.py

priorisationfileatte...

recherchedansliste.py

Bouclefor.py

Bouclewhile.py

conditionnel.py

devoirclasse.py

equation.py

Explications Exo 8 à 1...

OUTLINE

TIMELINE

86

87

elif choix == "3":

88

salle.afficher_places_reservees()

89

90

elif choix == "4":

91

print("Nombre de places disponibles :", salle.nombre_places_disponibles())

92

93

elif choix == "5":

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Entrez votre choix : & "C:/Users/ONEC STORE/AppData/Local/Microsoft/windowsApps/python3.12.exe" "c:/Users/ONEC STORE/Documents/DOCUMENTS/Docs ye/Cours Cybersécurité/DÉVELOPPEMENT DE LOGICIELS ET SÉCURITÉ DES APPLICATIONS/TP Python/TP3 POO/devoirenclass/Cascinema1.py"

Choix invalide. Veuillez entrer un nombre entre 1 et 7.

Menu Principal :

1. Réserver une place

2. Réserver une place spéciale

3. Afficher les places réservées

4. Vérifier le nombre de places disponibles

5. Filtrer les réservations par personne

6. Annuler des réservations

7. Quitter

Entrez votre choix :