

**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII**

**AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Informatică şi Ingineria Sistemelor**

**gr. IA-231, Chistol Maxim**

**Raport**

**pentru lucrarea de laborator Nr.4**

***la cursul de “Programarea Interactivă”***

Verificat:

**Alexei Victoria,** asistent.universitar.

Departamentul Informatică şi IS,

Facultatea FCIM, UTM

**Chișinău 2024**

**Tema:** OOP în Python.

**Clase. Obiective:**

• Studiul paradigmei OOP în Python și proprietățile.

• Crearea claselor, Atribute, Metode și Instanțe. Cerinţe:

• Revizuiți materialul didactic de la curs

• Executați sarcinile de la punctul I și II varianta conform registrului. I. Crearea unei simple clase.

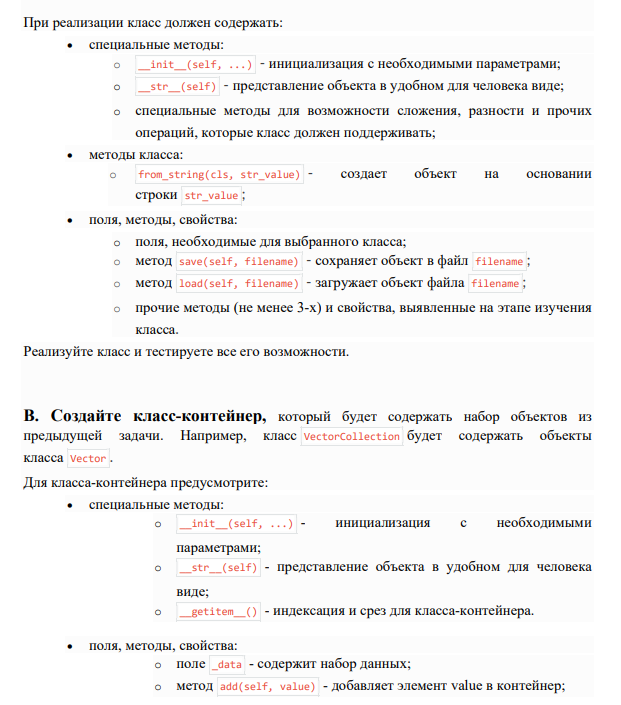
1. Alegeți varianta clasei din tabelul nr.1

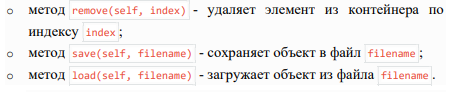


Прежде чем перейти к написанию кода:

• изучите предметную область объекта и доступные операции;

• для каждого поля и метода продумайте его область видимости, а также необходимость использования свойств.





|  |  |
| --- | --- |
| Codul: | Output: |
| import json  class Money:  def \_\_init\_\_(self, valuta: str, suma: float):  self.valuta = valuta  self.suma = suma  def \_\_str\_\_(self):  return f"{self.suma:.2f} {self.valuta}"  def \_\_add\_\_(self, other):  if self.valuta == other.valuta:  return Money(self.valuta, self.suma + other.suma)  raise ValueError("Nu se pot aduna sume în valute diferite.")  def \_\_sub\_\_(self, other):  if self.valuta == other.valuta:  return Money(self.valuta, self.suma - other.suma)  raise ValueError("Nu se pot scădea sume în valute diferite.")  @classmethod  def from\_string(cls, str\_value: str):  valuta, suma = str\_value.split()  return cls(valuta, float(suma))  def save(self, filename: str):  with open(filename, "w") as file:  json.dump({"valuta": self.valuta, "suma": self.suma}, file)  @classmethod  def load(cls, filename: str):  with open(filename, "r") as file:  data = json.load(file)  return cls(data["valuta"], data["suma"])  def convert(self, rata: float, valuta\_noua: str):  return Money(valuta\_noua, self.suma \* rata)  def split(self, numar\_persoane: int):  if numar\_persoane <= 0:  raise ValueError("Numărul de persoane trebuie să fie mai mare decât 0.")  return Money(self.valuta, self.suma / numar\_persoane)  class MoneyContainer:  def \_\_init\_\_(self):  self.\_data = []  def \_\_str\_\_(self):  return "\n".join(str(money) for money in self.\_data)  def \_\_getitem\_\_(self, index):  return self.\_data[index]  def add(self, value: Money):  self.\_data.append(value)  def remove(self, index: int):  if 0 <= index < len(self.\_data):  self.\_data.pop(index)  else:  raise IndexError("Indexul este în afara limitelor.")  def save(self, filename: str):  with open(filename, "w") as file:  json.dump([{"valuta": m.valuta, "suma": m.suma} for m in self.\_data], file)  def load(self, filename: str):  with open(filename, "r") as file:  data = json.load(file)  self.\_data = [Money(d["valuta"], d["suma"]) for d in data]  m1 = Money("USD", 100)  m2 = Money("USD", 50)  m3 = m1 + m2  print(m3)  m4 = Money.from\_string("EUR 200")  print(m4)  m4.save("money.json")  m5 = Money.load("money.json")  print(m5)  container = MoneyContainer()  container.add(m1)  container.add(m2)  container.add(m4)  print(container)  container.save("container.json")  new\_container = MoneyContainer()  new\_container.load("container.json")  print(new\_container) | 100.00 USD  50.00 USD  200.00 EUR  100.00 USD  50.00 USD  200.00 EUR |

|  |
| --- |
| **Codul:** |
| class Player:      def \_\_init\_\_(self):          self.\_status = "Oprit"  # Statusul părintelui (privat)      def porni(self):          self.\_status = "Redare"          print(f"{self.\_\_class\_\_.\_\_name\_\_}: Redarea a început.")      def opri(self):          self.\_status = "Oprit"          print(f"{self.\_\_class\_\_.\_\_name\_\_}: Redarea a fost oprită.")      # Metoda publică      def get\_status(self):          return self.\_status  class AudioPlayer(Player):      def \_\_init\_\_(self):          super().\_\_init\_\_()      def opri(self):          super().opri()          print("AudioPlayer: Redarea audio a fost oprită.")  class VideoPlayer(Player):      def \_\_init\_\_(self):          super().\_\_init\_\_()      def opri(self):          super().opri()          print("VideoPlayer: Redarea video a fost oprită.")  # Clasa DvdPlayer care extinde VideoPlayer  class DvdPlayer(VideoPlayer):      def \_\_init\_\_(self):          super().\_\_init\_\_()          self.\_pozitie\_curenta = 0  # Păstrează poziția de redare pentru DVD      # Implementare a metodei porni() pentru DvdPlayer      def porni(self, dvd\_disk):          self.\_status = "Redare"          self.\_pozitie\_curenta = 0  # Începe redarea de la început          print(f"DvdPlayer: Redarea DVD-ului '{dvd\_disk}' a început.")      # Implementare a metodei opri() pentru DvdPlayer      def opri(self):          self.\_status = "Oprit"          print(f"DvdPlayer: Redarea s-a oprit la poziția {self.\_pozitie\_curenta}.")      # Metodă pentru a simula avansarea la o poziție specifică      def set\_pozitie(self, pozitie):          self.\_pozitie\_curenta = pozitie          print(f"DvdPlayer: Poziția a fost setată la {self.\_pozitie\_curenta}.")  # Funcția principală care conține meniul interactiv  def meniu():      # Creăm obiectele player-elor      audio\_player = AudioPlayer()      video\_player = VideoPlayer()      dvd\_player = DvdPlayer()      while True:          # Afișăm meniul          print("\nMeniu:")          print("1. Pornește AudioPlayer")          print("2. Oprește AudioPlayer")          print("3. Pornește VideoPlayer")          print("4. Oprește VideoPlayer")          print("5. Pornește DvdPlayer")          print("6. Oprește DvdPlayer")          print("7. Setează poziția pe DvdPlayer")          print("8. Vezi statusul player-elor")          print("9. Ieși")          optiune = input("Alege o opțiune: ")          if optiune == '1':              audio\_player.porni()          elif optiune == '2':              audio\_player.opri()          elif optiune == '3':              video\_player.porni()          elif optiune == '4':              video\_player.opri()          elif optiune == '5':              dvd\_disk = input("Introdu numele DVD-ului: ")              dvd\_player.porni(dvd\_disk)          elif optiune == '6':              dvd\_player.opri()          elif optiune == '7':              pozitie = int(input("Introdu poziția pe care dorești să o setezi: "))              dvd\_player.set\_pozitie(pozitie)          elif optiune == '8':              print(f"Status AudioPlayer: {audio\_player.get\_status()}")              print(f"Status VideoPlayer: {video\_player.get\_status()}")              print(f"Status DvdPlayer: {dvd\_player.get\_status()}")          elif optiune == '9':              print("Ieșire din meniu.")              break          else:              print("Opțiune invalidă, te rog alege din nou.")  if \_\_name\_\_ == "\_\_main\_\_":      meniu() |

Meniu:

1. Pornește AudioPlayer

2. Oprește AudioPlayer

3. Pornește VideoPlayer

4. Oprește VideoPlayer

5. Pornește DvdPlayer

6. Oprește DvdPlayer

7. Setează poziția pe DvdPlayer

8. Vezi statusul player-elor

9. Ieși

Alege o opțiune: 1

AudioPlayer: Redarea a început.

Meniu:

1. Pornește AudioPlayer

2. Oprește AudioPlayer

3. Pornește VideoPlayer

4. Oprește VideoPlayer

5. Pornește DvdPlayer

6. Oprește DvdPlayer

7. Setează poziția pe DvdPlayer

8. Vezi statusul player-elor

9. Ieși

Alege o opțiune: 2

AudioPlayer: Redarea a fost oprită.

AudioPlayer: Redarea audio a fost oprită.

Meniu:

1. Pornește AudioPlayer

2. Oprește AudioPlayer

3. Pornește VideoPlayer

4. Oprește VideoPlayer

5. Pornește DvdPlayer

6. Oprește DvdPlayer

7. Setează poziția pe DvdPlayer

8. Vezi statusul player-elor

9. Ieși

Alege o opțiune: 3

VideoPlayer: Redarea a început.

Meniu:

1. Pornește AudioPlayer

2. Oprește AudioPlayer

3. Pornește VideoPlayer

4. Oprește VideoPlayer

5. Pornește DvdPlayer

6. Oprește DvdPlayer

7. Setează poziția pe DvdPlayer

8. Vezi statusul player-elor

9. Ieși

Alege o opțiune: 4

VideoPlayer: Redarea a fost oprită.

VideoPlayer: Redarea video a fost oprită.

Meniu:

1. Pornește AudioPlayer

2. Oprește AudioPlayer

3. Pornește VideoPlayer

4. Oprește VideoPlayer

5. Pornește DvdPlayer

6. Oprește DvdPlayer

7. Setează poziția pe DvdPlayer

8. Vezi statusul player-elor

9. Ieși

Alege o opțiune: 5

Introdu numele DVD-ului: Maxio

DvdPlayer: Redarea DVD-ului 'Maxio' a început.

Meniu:

1. Pornește AudioPlayer

2. Oprește AudioPlayer

3. Pornește VideoPlayer

4. Oprește VideoPlayer

5. Pornește DvdPlayer

6. Oprește DvdPlayer

7. Setează poziția pe DvdPlayer

8. Vezi statusul player-elor

9. Ieși

Alege o opțiune: 6

DvdPlayer: Redarea s-a oprit la poziția 0.

Meniu:

1. Pornește AudioPlayer

2. Oprește AudioPlayer

3. Pornește VideoPlayer

4. Oprește VideoPlayer

5. Pornește DvdPlayer

6. Oprește DvdPlayer

7. Setează poziția pe DvdPlayer

8. Vezi statusul player-elor

9. Ieși

Alege o opțiune: 7

Introdu poziția pe care dorești să o setezi: 4

DvdPlayer: Poziția a fost setată la 4.

Meniu:

1. Pornește AudioPlayer

2. Oprește AudioPlayer

3. Pornește VideoPlayer

4. Oprește VideoPlayer

5. Pornește DvdPlayer

6. Oprește DvdPlayer

7. Setează poziția pe DvdPlayer

8. Vezi statusul player-elor

9. Ieși

Alege o opțiune: 8

Status AudioPlayer: Oprit

Status VideoPlayer: Oprit

Status DvdPlayer: Oprit

Meniu:

1. Pornește AudioPlayer

2. Oprește AudioPlayer

3. Pornește VideoPlayer

4. Oprește VideoPlayer

5. Pornește DvdPlayer

6. Oprește DvdPlayer

7. Setează poziția pe DvdPlayer

8. Vezi statusul player-elor

9. Ieși

Alege o opțiune: 9

Ieșire din meniu.

# **Concluzii:**

Am realizat o ierarhie de clase pentru gestionarea diferitelor tipuri de playere, inclusiv AudioPlayer, VideoPlayer și DvdPlayer, fiecare având funcționalități specifice pentru pornirea, oprirea și setarea poziției de redare. Aceste clase derivă dintr-o clasă de bază Player, care gestionează statusul comun al tuturor playerelor. Am adăugat și un meniu interactiv, care permite utilizatorului să selecteze acțiuni pentru fiecare tip de player, să seteze poziția pentru player-ul DVD și să vizualizeze statusul fiecărui player.

Prin această implementare, am obținut un sistem modular, flexibil și ușor de utilizat, care poate fi extins cu ușurință prin adăugarea de noi tipuri de playere sau funcționalități suplimentare. Meniul interactiv facilitează testarea funcționalităților și interacțiunea cu aplicația, oferind o soluție eficientă pentru gestionarea diferitelor tipuri de playere. Acest proiect demonstrează cum putem organiza și controla diverse obiecte într-un mod structurat și intuitiv, respectând cerințele funcționale.