Comprehensive Python course for AI

Exercises 19 && 20

Deadline: 2023 21 December

Score: 600

2023 14 December

Comprehensive Python course for AI

Exercises 19

Score: 250

2023 14 December

Question 01 (OOP)

- Develop a simple banking application in Python, focusing on creating and handling custom exceptions.

- Custom Exception Classes:
  - Create two custom exception classes: InsufficientFundsError and InvalidTransactionError. Both should inherit from Python's built-in Exception class.

- BankAccount Class:
  - Implement a BankAccount class with private attributes: __account_number and __balance.
  - Include methods deposit(amount) and withdraw(amount).
  - Ensure that these methods raise InvalidTransactionError if the deposit amount is negative, and InsufficientFundsError if a withdrawal attempt exceeds the account balance.

Comprehensive Python course for AI

Exercises 20

Score: 350

2023 14 December

Question 01 (OOP)

- Build a foundational Python application for a library management system, focusing on basic Object-Oriented Programming concepts, private attributes, and custom string representations.

- Enhanced Person Class:
  - Use the Person abstract class with attributes name, age, and id_ (representing a unique identifier).
  - Include an abstract method introduce().
  - Add a __str__ method to return a string representation of the person.

- Subclasses with Specific Roles:
  - Librarian Class: Inherit from Person. Add employee_id as a public attribute.
    - Implement introduce() to return librarian-specific details.
  - Member Class: Also a subclass of Person, with a member_id.
    - Implement introduce() to return member information.

Question 01 (OOP)

- **Book Class with Private Attribute:**
  - Create a Book class with attributes title, author, and a private attribute isbn.
  - Implement a __str__ method to return a formatted string displaying the book's title and author.

- **Simple Library Class:**
  - Develop a Library class that manages books and members. Include methods for adding books and registering members.
  - Ensure the class has a __str__ method to represent the current state of the library (like the number of books and members).

- **Practical Implementation:**
  - Write a script to demonstrate the functionality of your classes. Create instances of Librarian, Member, and Book, and perform basic operations like adding books to the library.

Thanks

Good Luck!

Don't give up on your dreams ☺