Designing an Object-Oriented Stylized University System (2)

Introduction

Building on the university system built last time, you will now add functionality to represent PhD students and Monitors (PhD students who also teach). This exercise will deepen your understanding of class inheritance, including multiple inheritance, and how to structure your classes to accurately model complex real-world relationships.

New Requirements

You are tasked with extending the university system by adding two new classes:

1. `PhDStudent` – Represents a PhD student with specific research interests and a supervisor.
2. `Monitor` – A PhD student who also has teaching duties, embodying multiple inheritance.

Additionally, you will need to adjust the existing classes to accommodate these new roles.

Implementing the PhDStudent Class

- Extend the `Student` class to create the `PhDStudent` class, including attributes for their research area and supervisor.
- Ensure `PhDStudent` instances can enroll in courses just like any other student.

Creating the Monitor Class

- Define the `Monitor` class, which inherits from both `PhDStudent` and `Teacher`, to represent PhD students who teach.
- Address the challenge of multiple inheritance, particularly how to initialize both parent classes and manage their attributes effectively.

  For that purpose, you might need to change the Student and Teacher Classes.

Mixin class

Use a Mixin class `AttributePrinterMixin` to make it possible for each Student / Teacher to get all their attributes.

Conclusion

This exercise challenges you to extend your university system by incorporating advanced object-oriented programming concepts, including multiple inheritance.