

SIE508: Homework 4 ---Implementing Multiple Inheritance and Abstract Classes

Instructor: Silvia Nittel

Spring 2022

Assignment

The assignment is to implement a Real Estate Agency application. A real estate agency has many agents; each agents manages properties, which can be houses or apartments. Both can be for sale or rent (see Multiple Inheritance Lab). The UML diagram is given below.

The assignment is to implement all the classes from the UML diagram below in Python. Similar to HW3, the agent class should also have a class variable, which is a list and contains all the agents that have been created. Each agents has a list of all the properties he/she manages however, this is an object variable. This code can be reused from HW3 (your own or the solution provided).

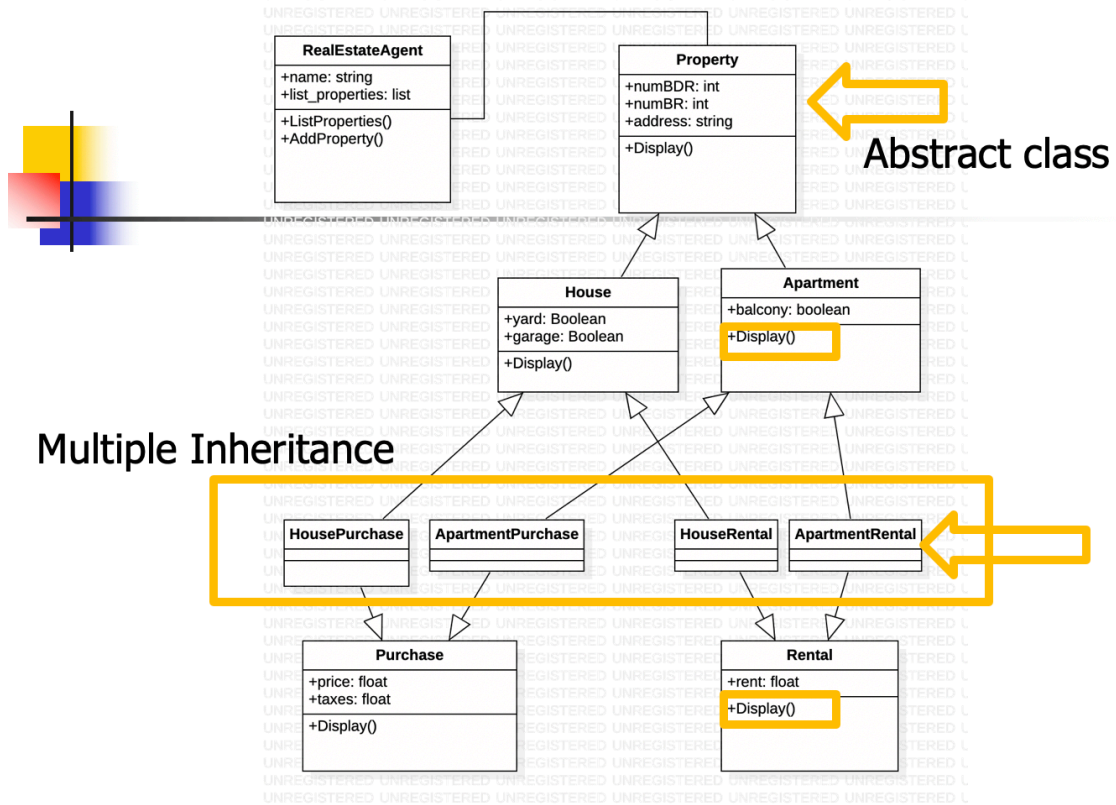
The focus on this assignment is to add **2 new classes**, that is **Purchase** and **Rental**. Furthermore, use multiple Inheritance to create the classes **HousePurchase**, **ApartmentPurchase**, **HouseRental**, and **ApartmentRental**. Please pay attention to the fact that these new classes now inherit the method `Display()` from two parents (check for method resolution order). If you implement the method correctly, you can just call the functions of the parents.

To make this simpler than HW3, none of the classes need methods that prompt the user for information about properties. Objects can be initializes via the constructors. You can use explicit constructors or you can experiment with kwargs, depending on which approach you are more comfortable with.

Furthermore, the **class Property** should now be an **abstract class**, and the method `Display()` should be an abstract method.

Now, write a simple application (main) that creates 2 agents objects. Next, create 1 types of the new multiple inheritance property objects and distribute them among the 2 agents. Third, iterate through the list of agents (objects) and print every agent's properties, listing the property type (rental or sale), price or rent, and address.

Note, if this is done right, you don't need to change the code of the agent nor the main application code besides creating the new objects.



Deliverables

1. Several .py files, all zipped into a single file.

Due: Monday, April 6 2022, 11:59pm via courses.maine.edu.