**HW7 CS5004 codewalk**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Properties**  **inheritance, immutable** | | **Contracts**  **inheritance ,immutable** | | **Listing**  **<generic> composition**  **immutable** | **Agent**  **<T, E>, mutable** |
| Features: | Features:  String address;  Integer size; | | Features:  Double asking price;  Boolean negotiable; | | Features:  T Property; <generic>  E Contract; <generic>  <T, E> | Features:  String name;  Listing[]; <generic>  Double commission rate;  Double total earnings; |
| **Commercial**  Integer number of offices;  Boolean suitable for retail | **Residential**  Integer: number of bedrooms;  Double: number of bathroom | **Rental**  Term in months | **Sale** |
| Methods(at parent class) |  | |  | |  | **Interface:**  Void addListing(Listing);  Void completeListing(Listing);  Void dropListing(Listing);  Double getTotalPortolioValue(); |
| Additional methods we may consider |  | |  | |  | **Overload:**  Void printListing();  Void printListing(askingprice); |
| Group member | Peng | | Peng | | Liu | Yang |
| DDL | Saturday 10:30 PM PST(1:30 PM Beijing) | |  | |  |  |

1. How you split up the work between team members?

2. How your design is extensible?

Building various classes instead of single class, which make the program extensible. It is flexible to add more attributes and method for any class and keep other classes intact.

Generic

3.Where you chose to use inheritance vs composition and vice versa.

In Properties class and Contracts class, we implement inheritance because their subclasses have more specialized version of the superclass.

In the Listing class, we implement composition because Listing has Property and Contract.

4.How you approach the generic requirement?

5.Any other key design decisions e.g. if you though a particular class should be immutable, Why?

We let Properties, Contracts, and Listing are immutable because their attributes will not change after it is constructed, which makes the coding and maintenance very easier.

In contrast, we let Agent class be mutable because we need update and edit the status of an agent object frequently.