ICS4U1 Assignment 1 – Realty Database

You are hired by the AYJ Realty to implement a database to keep track of the properties that are listed with them. The main goal of the database is to efficiently and accurately search for properties based on clients' needs.

There are two main types of properties that AYJ lists: condominiums and houses. Besides its address, each property has a unique id for easy referral. When matching properties for clients, there are two sets of specifications that the database needs to look at: the primary spec, and the secondary spec.

The primary spec is a list of characteristics of a property that must match all of client's requests. This list includes location (represented by a zone code) price, size and number of bedrooms. For example, if a client has a budget of $450,000, and is looking for a property with 4 bedrooms property and of size 2500 sq ft in area N11, the match must be in N11, at least 2500 sq ft, has at least 4 bedrooms and less than $450,000.

The secondary spec is a list of characteristics and amenities that may not need to match all of client's requests. Each of condominiums and houses has a different list. The list for condos includes availability of swimming pool, exercise room, and locker and whether hydro and cable are included in the monthly management fee. The list for houses includes availability of central vacuum, central air conditioner, fireplace, and whether the house has hardwood flooring and finished basement. When requesting for a match, clients can specify the minimum percentages of items that has to meet their request. For example, if a client is looking for a house with central vacuum, AC, a fireplace, but no hardwood flooring and finished basement, and he only requires 50% match for his request, then a house with central vacuum and AC, hardwood flooring, but no fireplace and finished basement can be a match since it satisfy 3 out of 5 (more than 50%) of the request.

Besides searching for properties, the database should also be able to provide other information to assist both the agents and the clients. Here is the complete list of its functionalities and they should be displayed as options on a menu of the application.

1. Load properties from a file
    - Load all properties from a text file "data.txt". The text file must follow the format specified in the next section
2. List all properties
    - List the id, address and all specs of all properties
3. List all condominiums
    - List the id, address and all specs of all condominiums
4. List all houses
    - List the id, address and all specs of all houses
5. Search for a property by ID
    - List the id, address and all specs of the properties with the given ID, or output message if ID not found.
6. Search for properties by specifications

- Prompt user to enter the required specifications (all primary and secondary specs) and the percentage of secondary specs that needs to be matched
- Print the list of properties (id, address and all specs) that match the required specifications

7. Print average price of all properties in the specified zone
    - Prompt user for a zone
    - Print the average price of all properties in the given zone
8. Find the largest condo
    - Output the largest condo (id, address, and all specs)

You may assume the user will always call option 1 at the start of execution.

Text File Format

The text file begins with the total number of properties followed by a blank line, then the information of the properties in the following order:

If the property is a house

```
house
<id>
<address>
<zone code>
<price>
<size>
<number of bedrooms>
<vacuum (Y/N)>
<air condition (Y/N)>
<fireplace (Y/N)>
<hardwood floor (Y/N)>
<finished basement (Y/N)>
```

If the property is a condo

```
condo
<id>
<address>
<zone code>
<price>
<size>
<number of bedrooms>
<pool (Y/N)>
<exercise (Y/N)>
<locker (Y/N)>
<hydro included (Y/N)>
<cable included (Y/N)>
```

Information of individual properties are separated by a blank line. Here is an example of the text file:

```
2

house
1
15 Francine Drive
N10
800000
3500
4
Y
Y
Y
N
Y

condo
2
1415 2500 Bayview Ave
N11
350000
1200
2
Y
Y
N
Y
N
```

Design

The realty database can be modeled with the following classes:

- **Property**
  - Fields: `id, address, primSpec, secSpec`
  - Methods:
    Constructor – initialize all fields
    `matchProperty` - returns whether this property matches the specified primary and secondary spec (with given matching percentage)
    `toString` – returns the id and address ONLY in organized format
- **Condo** – subclass of Property
  - Methods:
    Constructor – initialize all fields
- **House** – subclass of Property
  - Methods:
    Constructor – initialize all fields
- **PrimarySpec**
  - Fields: `zoneCode, price, size, numBedRoom`

- Methods:
  Constructor – initialize all fields
  `matchSpec` - check if this primary spec matches the given primary spec
  `toString` - Returns the value all of the specification in this object in an organized format

- **SecondarySpec** - abstract class
  - Methods:
    `matchSpec` – abstract method
- **CondoSpec** – subclass of SecondarySpec
  - Fields: `pool, exRoom, locker, hydroIncluded, cableIncluded`
  - Methods:
    Constructor – initialize all fields
    `matchSpec` - check if this condo spec matches the given percentage of the given spec
    `toString` - Returns the value all of the specification in this object in an organized format

- **HouseSpec** – subclass of SecondarySpec
  - Fields: `vacuum, ac, fireplace, hardwood, basement`
  - Methods:
    Constructor – initialize all fields
    `matchSpec` - check if this condo spec matches the given percentage of the given spec
    `toString` - Returns the value all of the specification in this object in an organized format

- **PropertyDatabase**
  - Fields: `numProperty, list`
  - Methods:
    `seachByID` - return the Property object that has the specified ID, null if not found
    `printAllMatch` - Prints the list of properties (their ID and address) that match the specified primary and secondary specification
    `averagePriceInZone` - Returns the average price of all properties in the specified zone
    `largestCondo` - Returns the Condo object that has the largest size
- **Realty**
  - Methods:
    `main` – display menu and perform actions according to user's choice

Marking Rubrics

Criteria	4	3	2	1	0
Application: (20%) Correctness on Program Execution	All test runs of the program produce correct results	Most run situations provide correct results	The program produces some correct results	The program produces few correct results	The program produces no correct results
Application: (15%) Correctness on Object Oriented Programming	Object oriented programming structure is used consistently to solve problem	Object oriented programming structure is used frequently to solve problem	Object oriented programming structure is used moderately to solve problem	Limited use of object oriented programming structure to solve problem	Problem is solved without object oriented programming concept
Knowledge: (15%) Understanding on Object Oriented Concepts – Inheritance, Polymorphism and Abstract Classes	Evidence of understanding on OO concepts is high (e.g. correct use of super class` fields and method, class casting only when necessary)	Evidence of understanding on OO concepts is considerable	Evidence of understanding on OO concepts is moderate	Evidence of understanding on OO concepts is limited	No evidence of understanding on OO concepts
Thinking / Inquiry: (25%) Object Oriented Design	Purposes of all classes and methods are carefully thought through (all of them have isolated purpose, e.g., some methods should return the result rather than printing it)	Purposes of most classes and methods are carefully thought through	Purposes of some classes and methods are carefully thought through	Purposes of few classes and methods are carefully thought through	Purposes of no class and method is carefully thought through
Knowledge: (5%) Programming Style • identifier names • use of constants • program order	Evidence of the correct use and formatting of Java code is high	Evidence of the correct use and formatting of Java code is considerable	Evidence of the correct use and formatting of Java code is moderate	Evidence of the correct use and formatting of Java code is limited	No evidence of the correct use and formatting of Java code

• efficiency					
Communication: (5%) User Friendliness	Instructions for user show a high degree of clarity Uses proper spelling and grammar Program flow is intuitive	Instructions for user show a moderate degree of clarity Few spelling and grammar errors Program flow is moderately intuitive	Instructions for user show some degree of clarity	Limited attention to user friendliness	No attention to user friendliness
Communication: (5%) Class, Method Comment Header	All class and method comment headers contains all elements	All class and method comment are present but miss one or two elements	Some comment headers are missing	Most comment headers are missing	No comment header
Communication: (5%) Internal Comments explain the program	Comments are extremely useful and help understand the program	Some comments are missing or some are not needed (redundant)	Missing comments	Very few comments	No comments
Communication: (5%) Spacing and Indentation Aids Readability	Easily readable	Minor improvement needed in spacing and indentation	Poor spacing and indentation (difficult to read the program)	Very poor spacing and indentation	No indentation or spacing