DATA533 - Collaborative Software Development

Lab Assignment - 2

Team Members:

- Navdeep Singh Saini [84280635]
- Mehul Bhargava [35211721]
- Andrew Nguyen [73490641]

FUNCTION DETAILS

The package name is "realestate". It has various classes, functions defined in it which are used to extract information about real-estate properties like apartments and family houses. It basically helps to find various properties of two types of real-estate properties (Apartments and Family Homes) and display all the relevant information. It has two sub-packages "apartment" and "familyhome". Each of the sub-packages helps to find the information and characteristic of a particular type of real-estate property, based upon the needs and requirements of the user. This package requires the user to enter certain details about their requirements and based on that it gives appropriate results.

The first sub-package that is "apartments" gives the details about an apartment type real-estate property. The relevant details could be the price of an apartment, number of floors it has, the area (in square feets), how much discount the customer will get, maintenance is required or not, furnishing status, its location, the city in which it is located and the face direction.

The second sub-package that is "familyhouse" gives the details about a normal family house (other than apartments/bunglows) type real-estate property. The relevant details could be the price of a house, number of floors it has, the area (in square feets), its build quality, what near by amenities it has (Hospital, Park, Shopping Center, School, Mall, Library, Beach, cafe, restaurant, etc.), the view it has (sea facing, sun facing, etc.), how much population it has in the locality, furnishhing style/conditions, garage details and swimming pool details.

Sub-package 1: Apartment: Displays various dettails for apartments.

- **Module-1 feature**
  - class 1 (parent): Features
    - apartment_price - This method reads the dictionary and returns the value of the appropriate price values based on the values which were used to instantiating the object. For example, if the we passed "1BHK" to the object of the child class, the same value will be passed to the parent class, and then it will be matched by this function in the dictionary with the key value and will return the value of the price associated with the key (In this case "1BHK").

        - apartment_area - This method reads the dictionary and returns the value of the appropriate area values based on the values which were used to instantiating the object. For example, if the we passed "CANADA" to the object of the child class, the same value will be passed to the parent class, and then it will be matched by this function in the dictionary with the key value and will return the value of the price associated with the key (In this case "CANADA").
        - apartment_display - This method is used to output the price and the area of the apartment based on the inputs provided by the end user, for any the categories such as type of apartment, location, or condition of the property in terms of how much years old it is. This is the common method which is used in the super Feature Class for all the objects of the child class.
    - class 2 (child): ApartmentType
        - apartment_floors - This method to is being used to return the list of floors available based on the apartment type which is being used to instantiate the object, for example if we are using the apartment type as "2BHK" we are returning the list as [1, 4, 5, 8, 10], basically we created a dictionary in this class and passed apartment type in the key and list of floors in the value, and the function to accessing the dictionary and returning the list of floors based on the key value from the dictionary.
        - (all other methods of parent class features) - We have already discussed apartment_price and apartment_area above in the parent class, but in the apartment_display function in the child class we are calling the parent class apartment_display method to provide the output of price and the area, we are using using this to call the apartment_floors method to print the floor which is returned from that function.

- **Module-2 location**
    - class: Location(child)
        - apartment_cities - This method to is being used to return the list of cities available based on the location country which is being used to instantiate the object, for example if we are using the location country as "UK" we are returning the list as ['London','Birmighan','Leeds','Southampton'], basically we created a dictionary in this class and passed country in the key and list of cities in the value, and the function to accessing the dictionary and returning the list of cities based on the key value from the dictionary.
        - apartment_locality - This method to is being used to return the list of street localities available based on the location country which is being used to instantiate the object, for example if we are using the apartment type as "USA" we are returning the list as ['Charles Street','Brown Road','Morgan Highway','Alpine Park'], basically we created a dictionary in this class and passed country in the key and list of localities in the value, and the function to accessing the dictionary and returning the list of localities based on the key value from the dictionary.

        - apartment_facing_direction - This method to is being used to return the list of possible directions of the properties available based on location country which is being used to instantiate the object, for example if we are using the apartment type as "CANADA" we are returning the list as ['South West','North','West','South East'], basically we created a dictionary in this class and passed country in the key and list of directions in the value, and the function to accessing the dictionary and returning the list of directions based on the key value from the dictionary.
        - (all methods imported from the parent class, which is in the features module) - We have already discussed apartment_price and apartment_area above in the parent class, but in the apartment_display function in the child class we are calling the parent class apartment_display method to provide the output of price and the area, we are using using this to call the apartment_cities,apartment_locality,apartment_facing_direction methods to print the list of cities,localities,directions which is returned from that function.

- **Module-3 condition**
    - class: Condition(child)
        - apartment_maintenance - This method to is being used to return whether maintenance is required or not based on the hwo many years the property is old which is being used to instantiate the object, for example if we are using the age of the property as "0-3" we are returning the values as "Not Required", basically we created a dictionary in this class and passed years in the key and maintenance status as the value, and the function to accessing the dictionary and returning the value of maintenance status based on the key value from the dictionary.
        - apartment_discount - This method to is being used to return range of discount based on the hwo many years the property is old which is being used to instantiate the object, for example if we are using the age of the property as "3-5" we are returning the values as "5-10%", basically we created a dictionary in this class and passed years in the key and and discount range as value, and the function to accessing the dictionary and returning the value of discount range based on the key value from the dictionary.
        - apartment_furnishing_status - This method to is being used to return whether it is fully or semi furnished required based on the hwo many years the property is old which is being used to instantiate the object, for example if we are using the age of the property as "0-3" we are returning the values as "Fully Furnished", basically we created a dictionary in this class and passed years in the key and furnishing status as the value, and the function to accessing the dictionary and returning the value of furnishing status based on the key value from the dictionary.
        - (all methods imported from the parent class, which is in the features module) - We have already discussed apartment_price and apartment_area above in the parent class, but in the apartment_display function in the child class we are calling the parent class apartment_display method to provide the output of price and the area, we are using using this to call the

apartment_maintenance,apartment_discount,apartment_furnishing_status methods to print the maintenance status,discount range,furnishing status which is returned from that function.

Sub-package 2: FamilyHome: predicts various properties for family homes.

This subpackage has three modules. They are termed as attributes, areatype and highlights. In this sub-package, the concept of inheritance is used in all of the modules. The first module has two classes, one of them is a parent class/super class/base class and another one is a child class. Other two modules withing this sub-packages use the parent class of the first module and inherit features from it. Other two modules (areatype and highlights) are using the parent class features from the first module (attributes).
In each of the modules in this sub-package has its several dictionaries. The keys of these dictionaries have the values related to what user is going to enter and the value is the expected characteristic associated with each choice of a user. In this way all of the functions are returning the values of the dictionaries as an output by using the values entered by a user as a key.

- **Module-1 attributes** This module has two classes that are called as Characteristics (parent) and Quality (child). The characteristics and use of the functions defined in these classes are defined below.

    - class 1 (parent): Characteristics
        - house_price: This function analyses all of the details given by the user based on their requirements and it gives the price range for some houses according to the entries given by the user. The price range is an important criteria to buy a good house and take decisions wisely. Everyone has their own budget and this function helps the user to know the house prices available. It returns the house price range.
        - house_area: Considering the area while buying a new house is important. It gives you many benefits. And this function is designed for this purpose. This function analyses all of the details given by the user based on their requirements and it gives the area for some houses according to the entries given by the user. It returns the area of the house in square feet.
            - Characteristics_display: This function is common in all of the modules defined in this sub-package. It is the most important function as it is used by all of the classes defined in each of the module of this sub-package by inheritance. It returns details in a readable format for the user.
            - class 2 (child): Quality
            - build_quality: This function is within a child class. Build quality is important for deciding whether to give importance to a house or not while buying. Build quality decides the longitivity of a house. This function tells about the build quality of the house, based on the requirements of the user, which are entered by them. It returns the build quality of the house.
            - (all other methods of parent class): As we are using inheritence, all of the attributes of the parent class

(Characteristics) are inherited in this class.

- **Module-2 areatype**
  This module has one class that is called as AreaType. This class is a child class, and the parent class is Characteristics class which is in another module. Here, inheritance is used again. The characteristics and use of the functions defined in these classes are defined below.

  - class: AreaType (Child)
    - nearby_amenities: This function analyses all of the details given by the user based on their requirements and it gives all of the amenities near to this house property. Some examples of amenitites are Hospital, Park, Shopping Center, School, Mall, Library, Beach, cafe, restaurant, etc. The near byamenities is an important criteria to buy a good house and take decisions wisely. This function returns the list of all amenities that are near to the house.
    - view: This function returns the type of views available from different houses based on the user's needs. The view could be sea view, park view, etc. Views and direction to which house faces is really important and this function gives results for that properly.
    - population: Everyone prefer house whose locality is moderatley populated and not densely populated. This function returns the population count which is based on the entries given by the user. It uses a general retutn functon defined in python.
    - Characteristics_display: This function is common in all of the modules defined in this sub-package. It uses inheritance. It returns details in a readable format for the user.
    - (all methods imported from the parent class): As we are using inheritence, all of the attributes of the parent class (Characteristics) are inherited in this class.

- **Module-3 highlights** This module has one class that is called as Highlights. This class is a child class, and the parent class is Characteristics class which is in another module. Here, inheritance is used again. The characteristics and use of the functions defined in these classes are defined below.

  - class: Highlights (child)
    - garage_capacity: There could be no garage in a house, a garage with one car capacity or a garage with two car capacity. This function returns this property of a house. It uses the return function defined in python. These results are based on the user's requirements.
    - swimming_pool_type: This functions tells about the swimming pool in a house. It could be no swimming poop, indoor or outdoor swimming pool. This function returns this attribute of a house. Swimming pools are not necessarily important in homes, but some people usually prefer them while buying anew home.
    - furnish_type: This function returns the furnishing condition of a house. It uses the data entered by the user and gives the results

accordingly. Furnish type could be partially furnished, fully furnished or luxuriously furnished.

- Characteristics_display: This function is common in all of the modules defined in this sub-package. It uses inheritance. It returns details in a readable format for the user.
- (all methods imported from the parent class, which is in the features module): As we are using inheritence, all of the attributes of the parent class (Characteristics) are inherited in this class.