GUI

Project 1

Sławomir Aleksander Dańczak, M.Sc. Full-time studies 27 March 2021

Write an application that will be used by the developer to manage housing estates.The developer may have many housing estates in which there is any (finite) number of apartments for rent. Each apartment can be inhabited by any number of people for whom information should be stored (objects of type ***Person***). The first person who starts renting the apartment is responsible for the rental fees.

When renting an apartment, it is possible to rent an additional closed parking space. You can store vehicles, but also other items (e.g. cardboard) on the parking space.
Each apartment and parking space have information about their volume and unique identification number, thanks to which we can clearly define the object symbolizing the apartment or parking space.

The tenant of the apartment can freely check in and check out people from the apartment. If the tenant has a parking space, he can insert and take out items and vehicles. Each person may have many apartments and parking spaces rented, provided that the total number of apartments and parking spaces rented by a given person is not more than 5. Each room can have only one tenant at a time.

Make sure that the identifiers are unique and created automatically when creating an object of type ***Apartment*** and ***ParkingPlace***. The volume for both of these room types should be provided when creating the room object.

There are two possible ways of indicating the volume of the room:

- ✗ by entering the volume in cubic meters
- ✓ by entering the length, width and height of the room (we assume that the room is a cuboid. The volume of the room should be calculated based on the provided values).

The room for a specific tenant also has a rent start date and a rent end date. If the end date of the rent has expired, a letter is created (object of type ***TenantLetter***), which is saved to the object of class ***Person*** which symbolizes this particular tenant. The person also has information such as name, surname, pesel number and address.

You should also implement a mechanism for passing time using the thread mechanism. The thread should move the date forward one day every 5 seconds, simulating the passage of time. In parallel, rental issues are checked every 10 seconds - are all the rooms still being rented, or is the room rental already are expired.

If the rent is renewed or the rent is canceled by the tenant, the actual letter is removed from the person's object. If tenant does not renew the rent within 30 days, the room which the tenant has finished should be cleaned and the letter remains on file. In the case of a apartment, we evict people who live in this flat. In the case of a parking space, we utilize all stored items in this room.

If a person with more than three letters wants to start the rent (at least 3 objects of type ***TenantLetter***) an exception ***ProblematicTenantException*** should be thrown with the message - *"Person X had already renting rooms: list_spaces"*, where X is the name of the person, and *list_spaces* defines the rented rooms that have been rented.

Each item and vehicle has information about the name and volume it occupies. The space occupied by the object/vehicle can be provided in two ways as for a room.

The vehicles are divided into different types:

- off-road car,
- city car,
- boat,
- motorcycle,
- amphibious

Each vehicle except for the name and volume must have characteristics features specific to the type of vehicle (e.g. engine capacity, vehicle type, engine type, etc.). Characteristic features may recur for each type of vehicle, however at least one must be different from other vehicles.

When inserting any item or vehicle into a room, we must ensure that the room can accommodate that object. If this is not the case, an ***TooManyThingsException*** exception is thrown with the message *"Remove some old items to insert a new item"*.

It should also be possible to save the status of persons residing in the housing estate together with all data regarding the person, rooms, objects, etc. Saving takes place after selecting the appropriate functionality from the menu. The information stored in the file should be written clearly and human readable, observing the following rules:

- The rooms should be sorted in ascending order by room volume.
- The content of the room should be sorted in descending order by item volume, if it is the same then by name.

Principle of program operation:

- In the ***main*** method, create a housing estate with at least ten ready rooms of various types and volumes, and a few (minimum 5) ready people. Pre-allocate rental of apartments and parking spaces. Also place items and vehicles in parking spaces.
- After starting the application, the user should be able to call all of functionalities via the command console and the implemented menu.

The project is based on PPJ and GUI material.

Attention:

- ***Lack of knowledge of any line of code or plagiarism will result in obtaining 0 points for this project with the possibility of failing the entire subject.***
- ***Not only the practical and substantive correctness of the solution will be assessed, but also the optimality, quality and readability of the code written by you.***
- ***An important part of the project is the use of: inheritance, collections, interfaces or abstract classes, lambda expressions, Java Generics, additional functionalities or structures and other characteristic elements presented in the classes and lecture.***