

Decent Landlords Lettings - Property Class

This class stores the details of a property in the system.

Variables:

- Property Number (String) – Should be a string, in case its e.g. house 4b on a street, or has a name even. Only ever set once.
- Property PostCode (String) – Should conform to standard UK format: X[X]N[N] NXX, where [X] implies X is optional, and X is a letter and N is a number. Must include a space in the middle. Should therefore never be more than 8 characters. Only ever set once.
- To Let Count (int) – number of To Let votes received, auto increment when vote received
- Toilet Count (int) – number of Toilet votes received, auto increment when vote received
- Occupant Rating (double) – Current rating average from occupant ratings received.
- Occupant Rating Count (int) – number of ratings received, used in order to calculate mean values.
- Property Status (int) – Where 0 = Deactivated, 1 = Under Review, 2 = active, 3 = rented.

1) Default Constructor (no parameters)

If no variables are passed to the constructor, then it should create an empty address with values “X” for number and “XXXX XXX” for post code. Should set all values to 0, status should be set to under review.

2) Main Constructor

Takes two parameters – number and postcode. These should be set if valid, and the rest of the values set as per the default constructor.

3) Get method for Property Number

Returns the property number as a string.

4) Get method for Property Postcode

Returns the postcode as a string

5) Get method for Property Overall Rating

Combines the ToLet/Toilet ratio with occupant rating to return a score for their combined rating. Should be converted to the next integer so e.g. 0.7 becomes 1, and 4.3 becomes 5.

6) Get method for ToLet / Toilet Ratio Score

Converts the TT balance to a star rating out of 5. Should be converted to the next integer, so 0.7 becomes 1, and 4.3 becomes 5.

7) Get method for Occupant Rating

Returns the raw double for the occupant rating.

8) Return if under review

Returns true if status is under review

9) Return if visible or not on the website.

Returns true if active, but returns false if under review, or deactivated, or currently rented.

10) Set method for address

If the address isn't already set by the constructor, this takes the house number and postcode, checks them for correct format, and sets them. Should throw an error if postcode does not conform to expected format, or if the values are already set.

11) Add method for TT Rating

Takes two Booleans, tolet (true if it's a to let rating) and toilet (true if it's a toilet rating). Updates the correct variable given the passed values.

12) Add method for Occupant Rating

Takes an integer rating out of 5 (should check that it's a value rating score). And then adds to the average occupant rating. Should increment the number of ratings received after calculating the new average. Should throw an error if a score is not out of 5.

13) Set Under Review

Takes a Boolean for whether it is now under review or not. Should set the status to under review if true, or to active if not.

14) Set Deactivated

Takes a Boolean for whether it is deactivated or not. Should set the status to deactivated if true, or Under Review if false.

15) Set Rented

Takes a Boolean for if the property is rented or not. Should set the status to Rented if true, or Active if false.

16) toString() override

Should produce a custom String representation for this object. Should show as Property Address, followed by whether it is under review), and then the Rating the property has. E.g.

- "1 NG8 1BB (Under Review) - Rating: 4/5"

Decent Landlords Lettings – Landlord Class

This class will store the details of the landlord.

Variables:

- Landlord Name (String) – should be at least 5 characters presuming all first names and surnames are at least two characters long, separated by a space. Can only be set once.
- Landlord Email Address (String) – should conform to a proper email address format, including an @ and a dot between the domain and tld.
- Landlord Comms Rating (double) – their current average Comms Rating
- Landlord Comms Rating Count (int) – a count of ratings received so far.
- Landlord Maintenance Rating (double) – their current average rating for how well they maintain properties.
- Landlord Maintenance Rating Count (int) – a count of those ratings received so far.

1) Default Constructor

This constructor (which would take no parameters) should set the main values (name and email address) to null pending being set. All other numbers should be set to 0 and new landlord status set to true.

2) Main Constructor

Takes name and email address as values. Presuming they are correct, should set these values. Set all numbers to 0, and new landlord status to true.

3) Get Landlord Name

Should return the landlord name – should return an empty string if currently null.

4) Get Landlord Email Address

Should return the email address as string – should return an empty string if currently null.

5) Get Landlord Comms Rating

Should return the Communications Rating as a double.

6) Get Landlord Maintenance Rating

Should return the Maintenance Rating as a double.

7) Get New Landlord Status

Should count the number of ratings received and return true if <10 and false otherwise.

8) Set Landlord Name

Should only work if the name is currently null, as the name can only be set once. Should be checked for appropriate length and contains a space. Should throw an error if it is too short, doesn't include an error, or is already set.

9) Set Landlord Email Address

Should check that it conforms to the correct format (see above), and sets the string if ok. Otherwise, it should throw an error if the format is incorrect.

10) Add Ratings function

Takes two ratings (one for comms and one for maintenance). If these are both in the range of 1 to 5 (cant give them a zero), then these values are added to the rating, and the number of ratings is incremented. Otherwise (if both are not valid ratings), an error is thrown.

12) toString() Override

Creates a custom toString output for this object. Should say landlords name, followed by their ratings. Alternatively, if a new landlord, it should return their name followed by (New Landlord).

E.g.

- Wax Milson (New Landlord)
- Tammie Jycross – Communication: 4/5, Maintenance: 3/5

DLL App Main Class

This main class is responsible for managing the user input, for the first integration phase (no real user interface). It stays in a forever waiting loop, until the exit command is issued, waiting for the user to issue commands.

App Variables:

- ArrayList of DLL Landlord objects – A list of all the landlords
- ArrayList of DLLProperty objects – A list of all properties

1) Main Loop Method

This main method should run a loop offering the user commands.

For now, this offers users to

- List the Landlords
- List the properties
- List bad properties (below the 2.5 threshold)
- Add a rating to a property
- Add a rating to a landlord
- Exit

If the user, at any point, enters a command not recognised, it shows an error, but then re-offers the user the current options.

2) List Properties function

Loops through all the properties, listing them out to the user in format, giving each one an ID that can be referenced (correcting for the offset from 0, so the first item is listed as 1. But if they enter 1, then it should refer to object 0 in the arraylist).

- Takes a boolean for if listing only bad properties.
- If this Boolean is true, then properties are only listed if they are below the 2.5 threshold

3) List Landlords

Loops through all the landlords, listing them out to the user in format produced by toString. Giving each on an ID that can be referenced (correcting for the offset from 0, so the first landlord is listed as 1, but if they enter 1 then it should refer to object 0 in the arraylist).

4) Add ToLet/Toilet Rating for a property.

This function first lists the properties, and then prompts the user for which property (by ID) to add to. If a valid property ID is entered, then it asks for a rating to give the property. This value is set using the property objects addTTRating function. If that throws an error, it should handle this error, notify the user, and then return to the loop prompt, so they can try again from the start of this action.

5) Add Landlord Ratings

This function first lists the landlords, and then prompts the user for which landlord (by ID) to add to. If a valid landlord ID is entered, then it asks next asks for the comms rating, and then prompts them for the maintenance rating. This function passes these values to the function from the landlord object, and if that throws an error (because the ratings aren't valid), then it should handle this error, notify the user, and then return to the loop prompt, so they can try again from the start of this action.

6) Change property status

This function first lists the properties, and then prompts the user for which property (by ID) to update. If a valid property ID is entered, then it asks for which status to set it to (Decactivated, Active, Under Review, Rented). Calls the appropriate functions in the property object to change its status.