|  |
| --- |
| **Object-Oriented and Event-Driven Programming** |
| Octagon Lettings Agency |
| DV003874  Kelvin Hilton |
|  |
|  |
| **By Alex J Davison** |
| **21/04/2009** |
|  |

Contents

Page number

[Property, House and Lodging classes 3](#_Toc226997386)

[UML 3](#_Toc226997387)

[Java documentation 4](#_Toc226997388)

[Property 4](#_Toc226997389)

[House 7](#_Toc226997390)

[Lodging 8](#_Toc226997391)

[LettingsAgency Class 10](#_Toc226997392)

[UML 10](#_Toc226997393)

[Object diagram 11](#_Toc226997394)

[Sequence 12](#_Toc226997395)

[Add a house 12](#_Toc226997396)

[Add a lodging 13](#_Toc226997397)

[Display all free houses 14](#_Toc226997398)

[Display all free lodging 15](#_Toc226997399)

[Display all properties 16](#_Toc226997400)

[Rent property 17](#_Toc226997401)

[Re-Rent property 18](#_Toc226997402)

[Remove property 19](#_Toc226997403)

[Find property 20](#_Toc226997404)

[Exit 21](#_Toc226997405)

[Java documentation 22](#_Toc226997406)

[Graphical User Interface 24](#_Toc226997407)

[Sequence 24](#_Toc226997408)

[Add a house 24](#_Toc226997409)

[Add a lodging 25](#_Toc226997410)

[Display all free houses 26](#_Toc226997411)

[Display all free lodging 27](#_Toc226997412)

[Display all properties 28](#_Toc226997413)

[Rent property 29](#_Toc226997414)

[Re-rent property 30](#_Toc226997415)

[Remove property 31](#_Toc226997416)

[Find property 32](#_Toc226997417)

[Exit 33](#_Toc226997418)

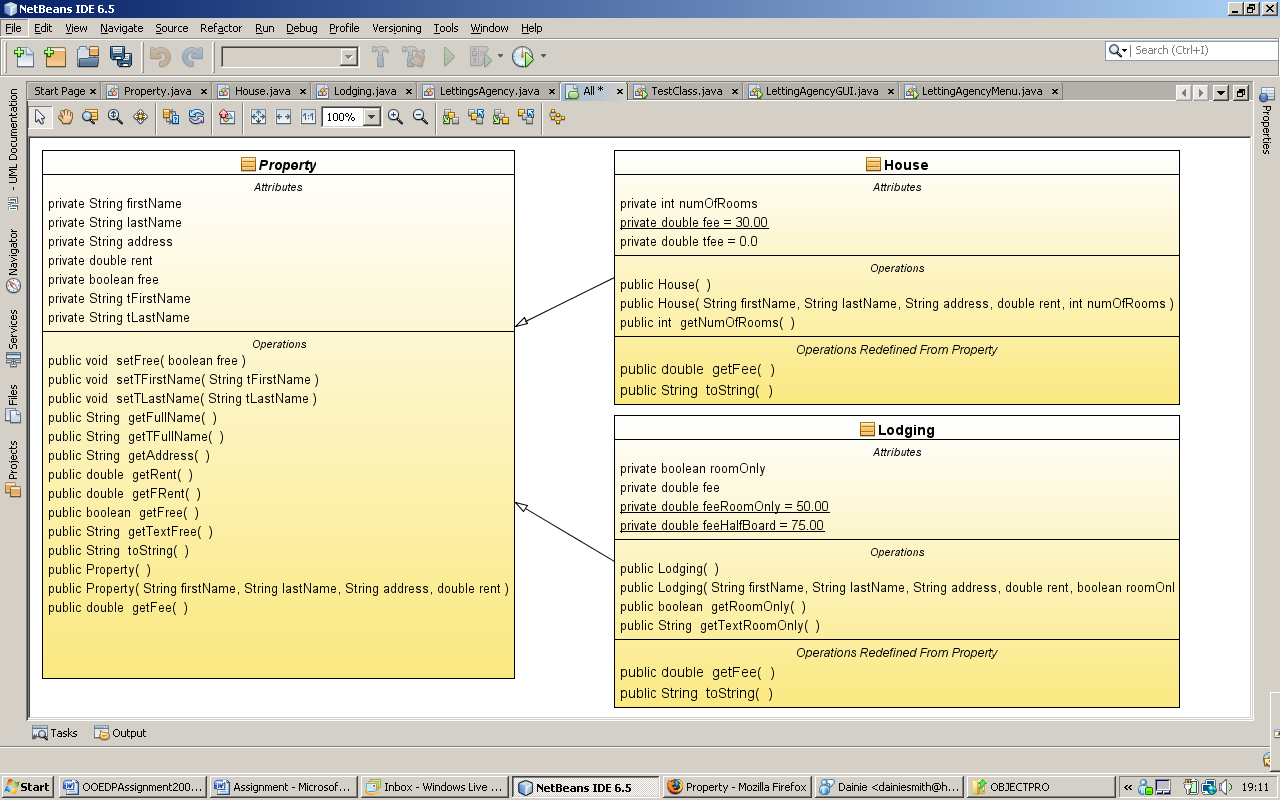
[Additional functionality 34](#_Toc226997419)

# Property, House and Lodging classes

Below are the detail concerning the Property, House and Lodging classes.

## UML

Below are the UML diagrams of the property, house and lodging classes.

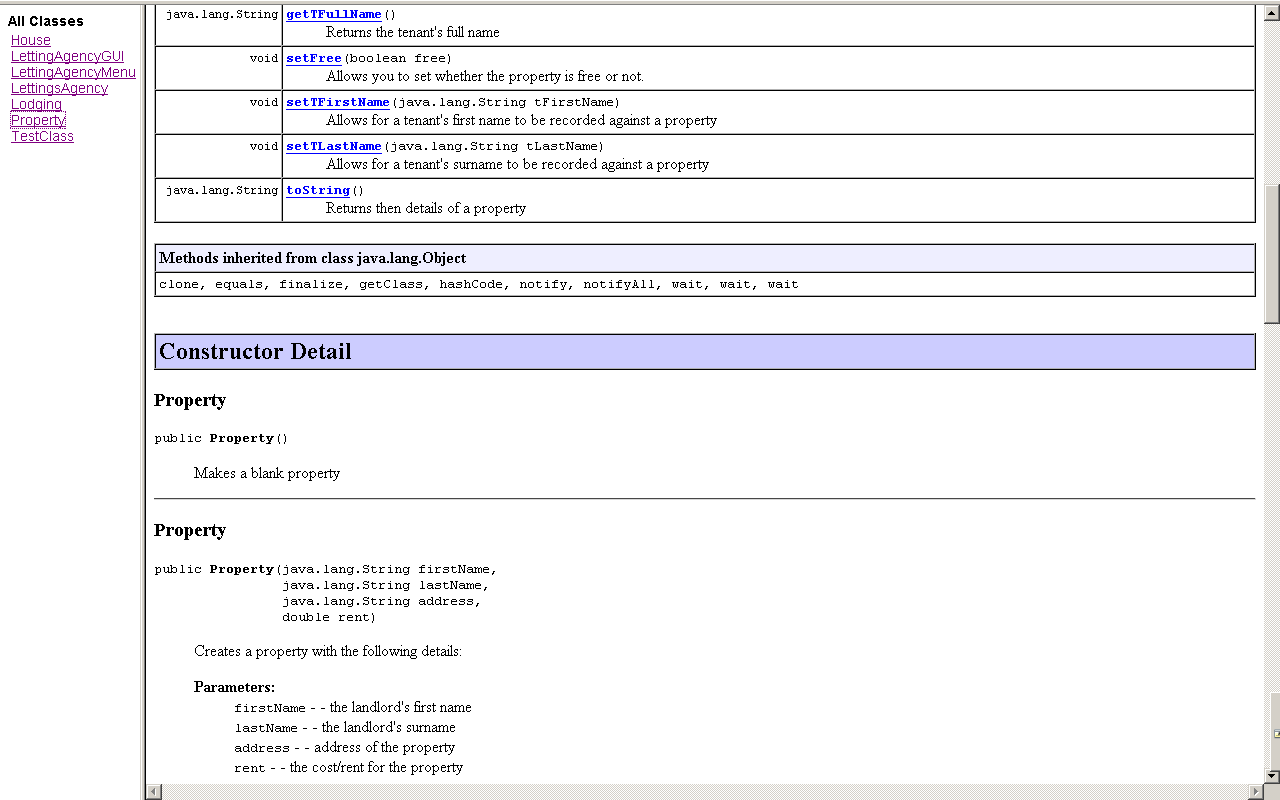


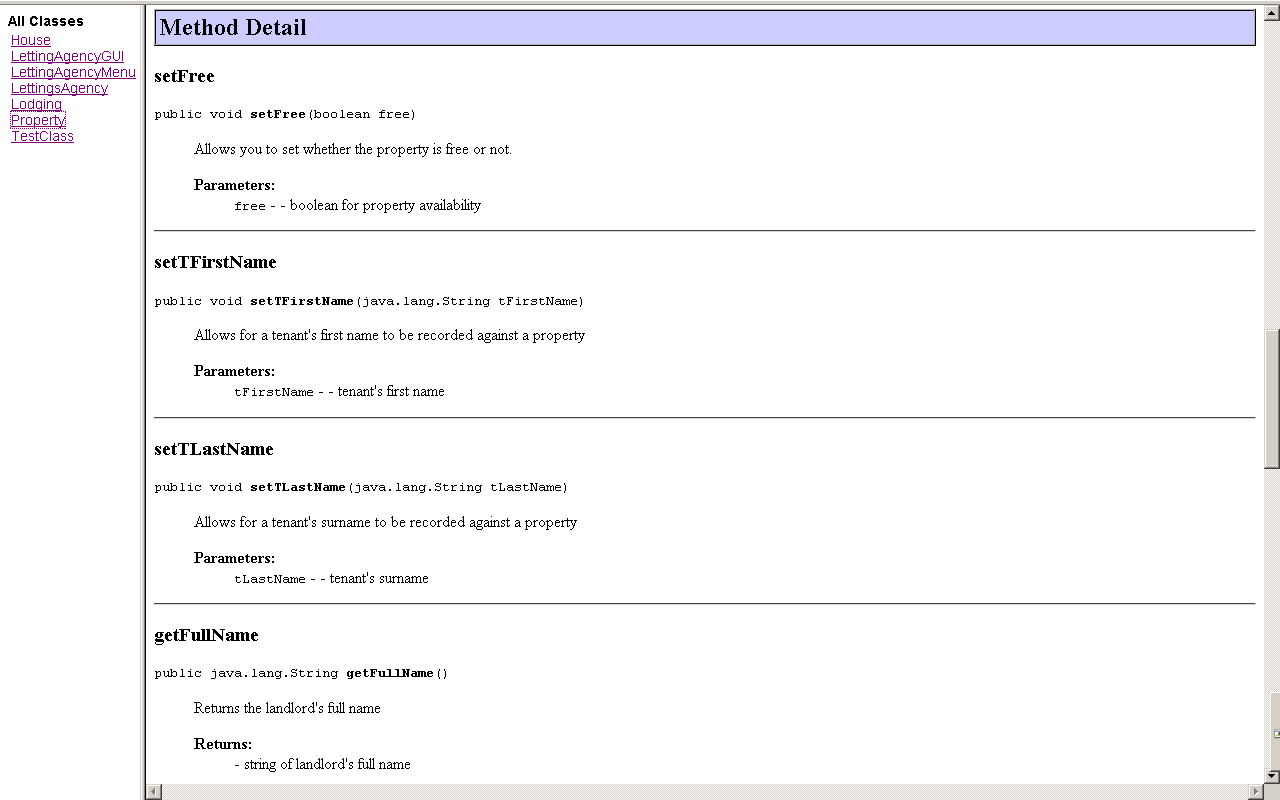
## Java documentation

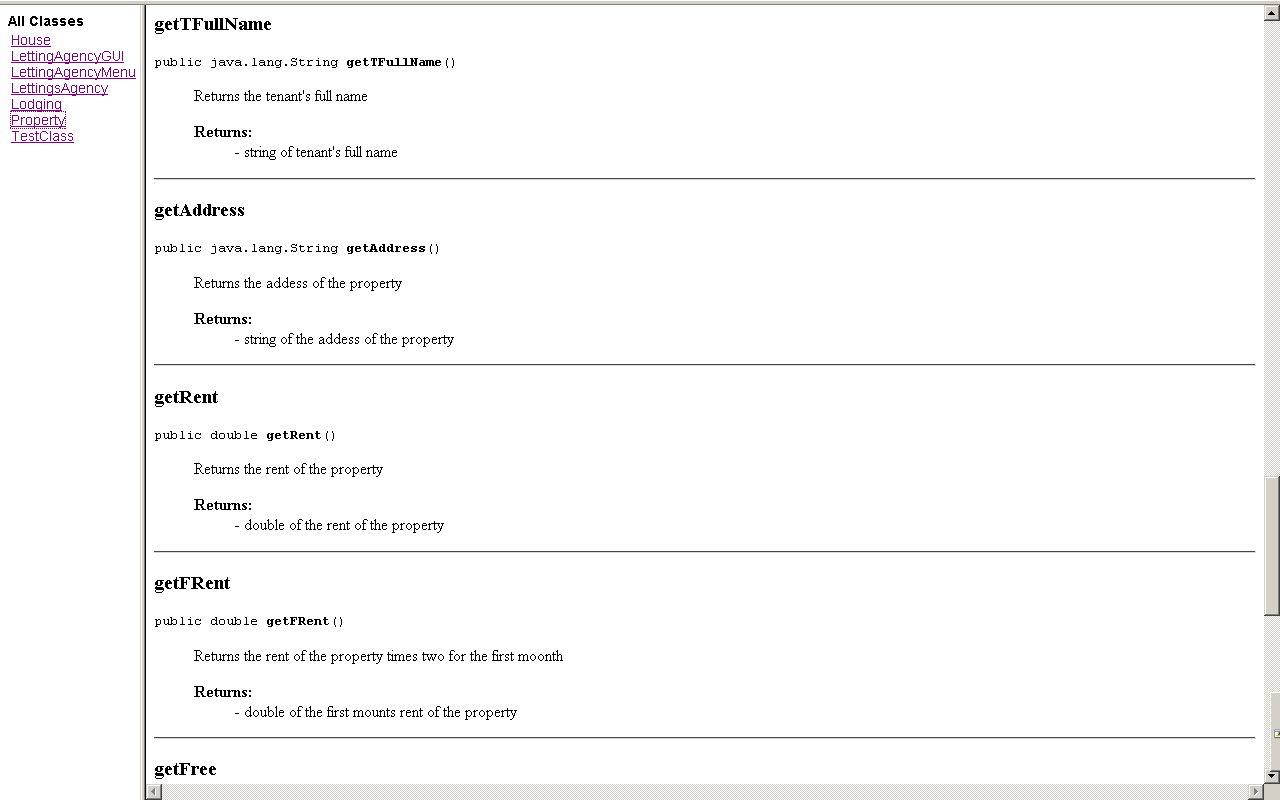
Below are screen shots of the java documentation for the property, house and lodging classes.

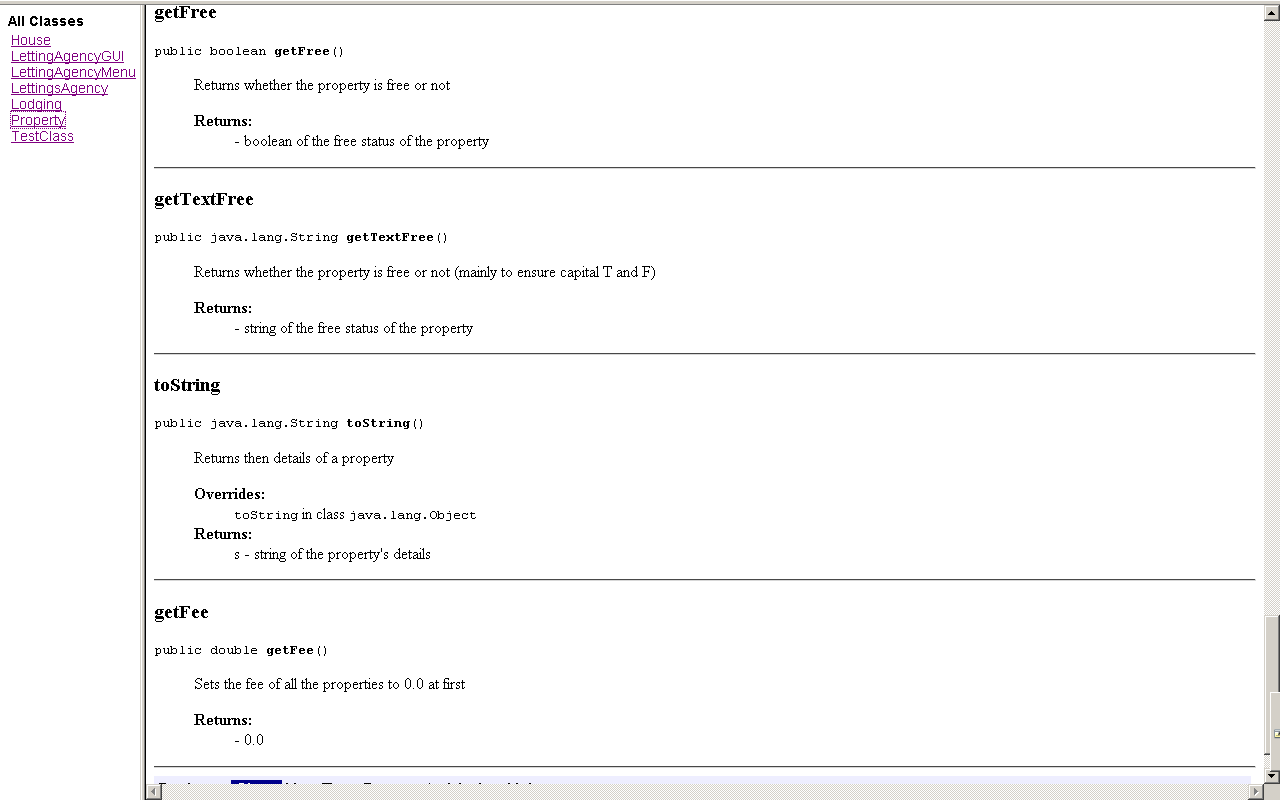
### Property





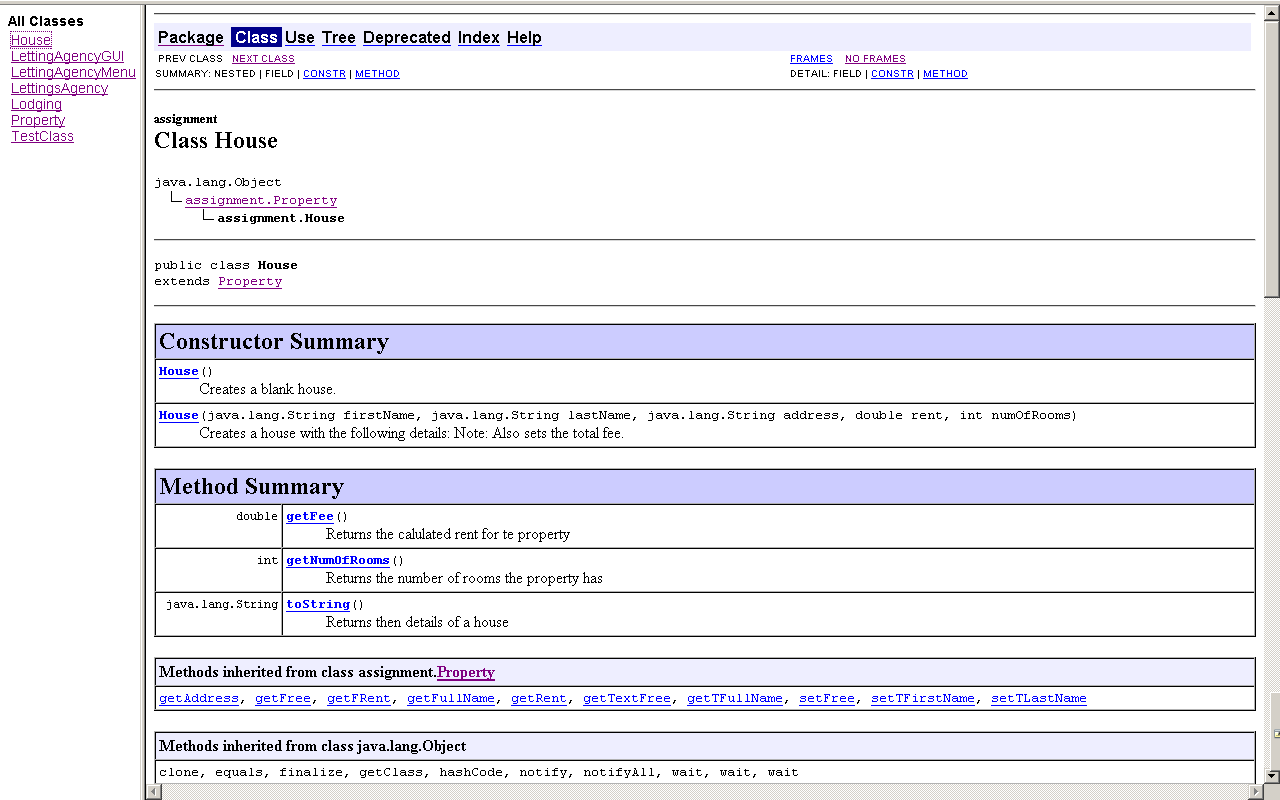


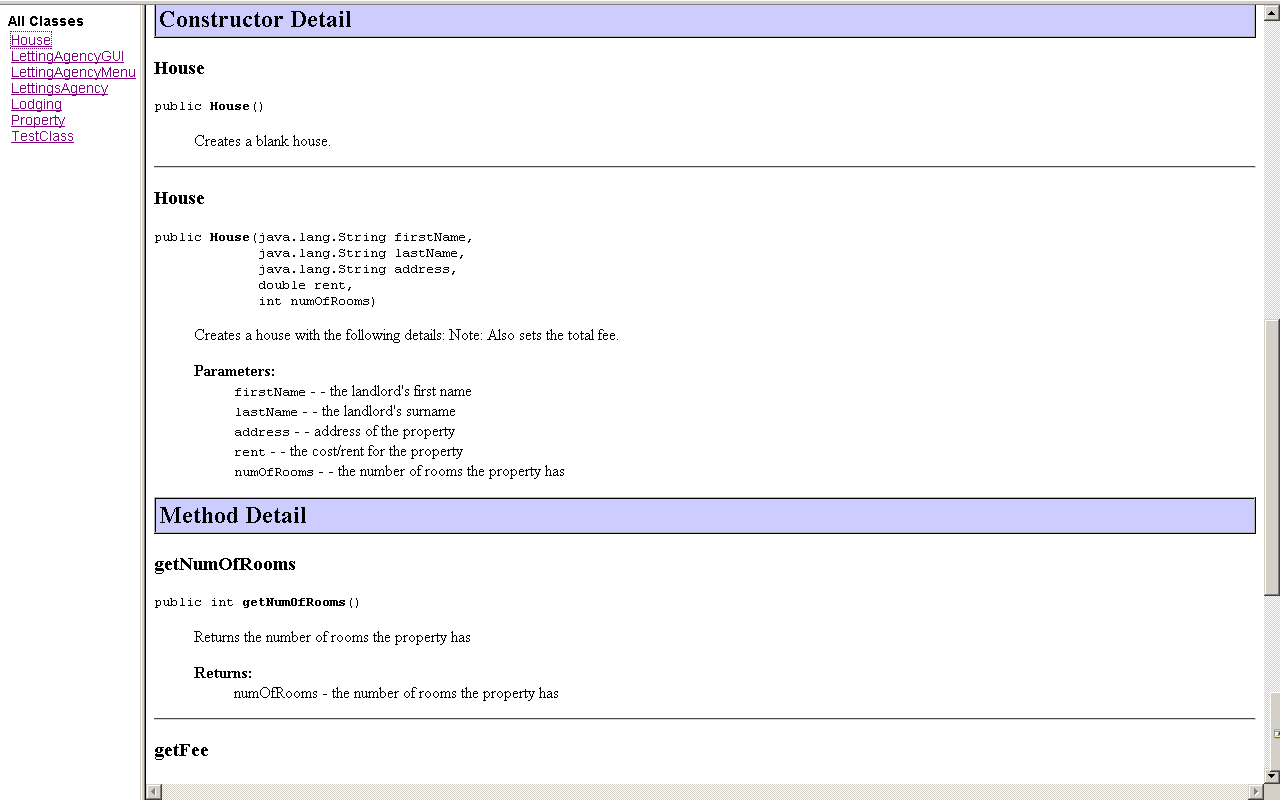




### 

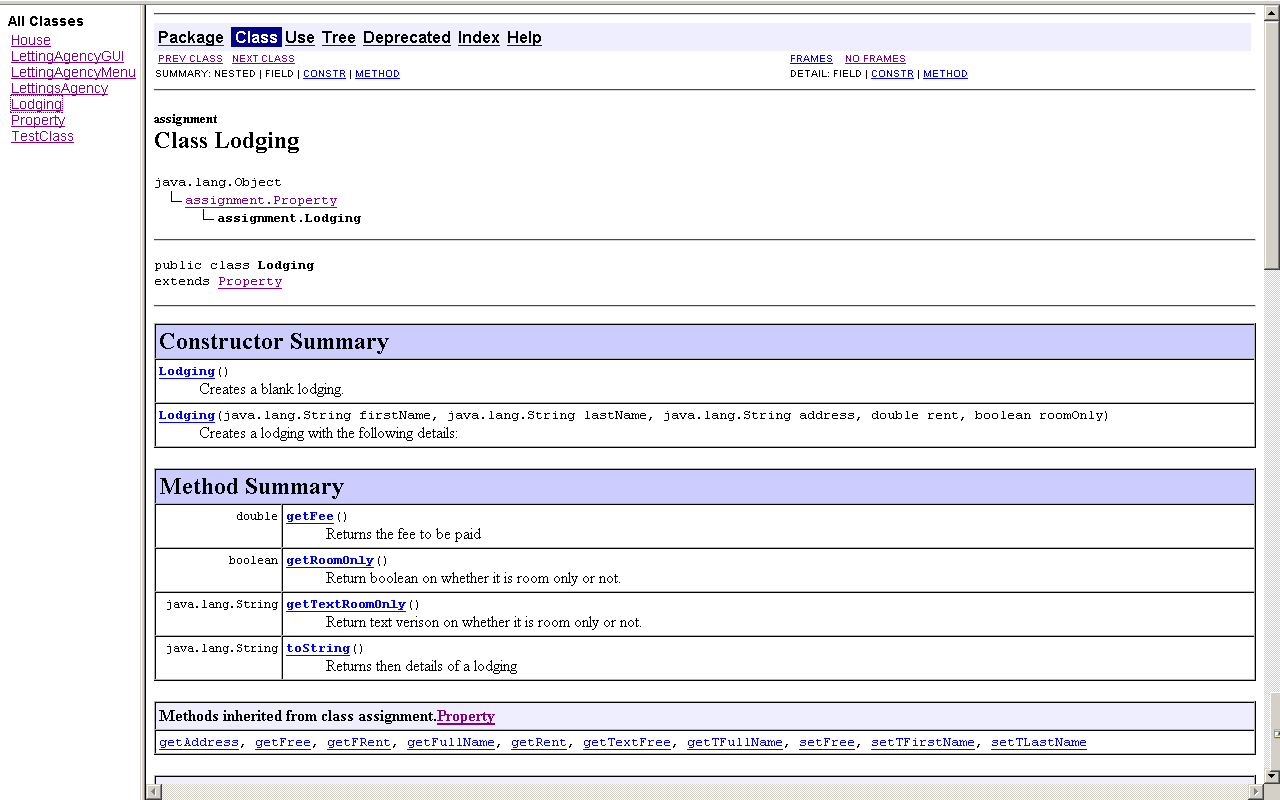
### House

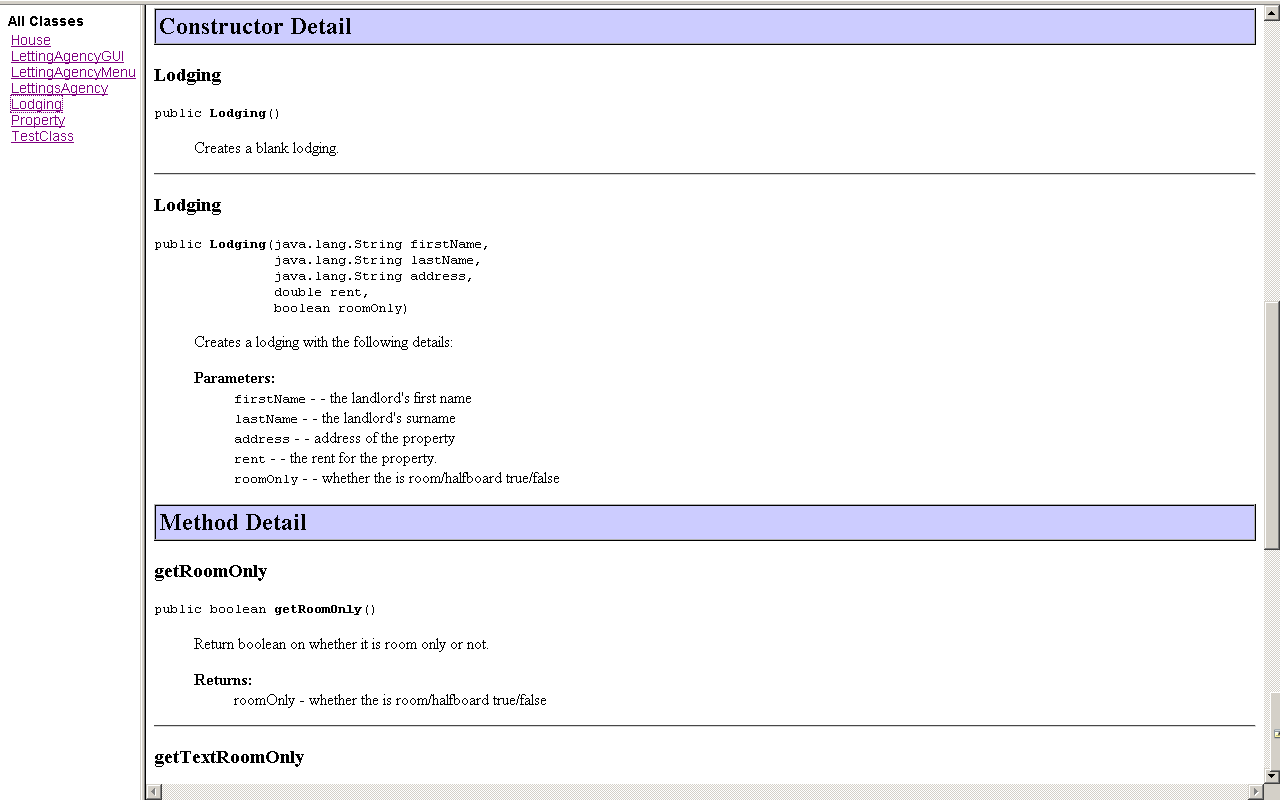






### Lodging



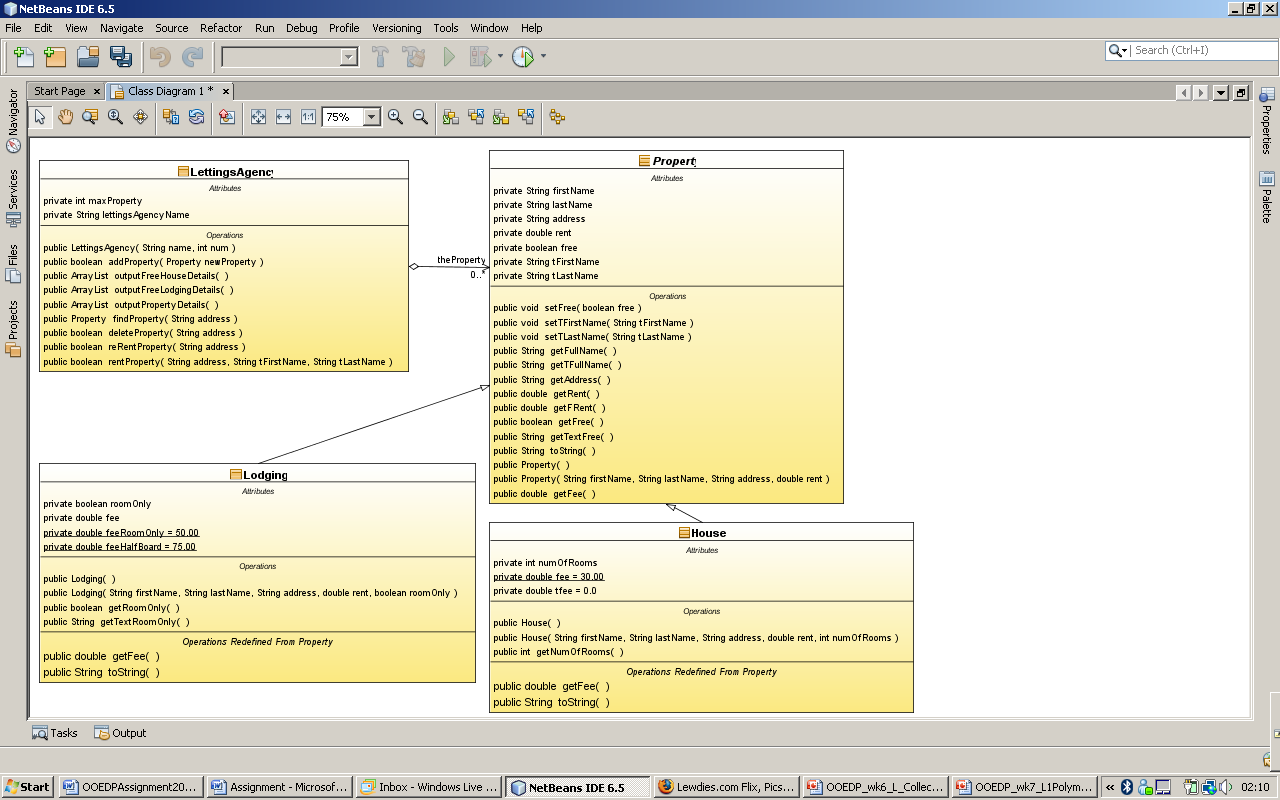




# LettingsAgency Class

Below is the UML concerning the LettingsAgency class.

## UML



## Object diagram

TestClass

H2: House

firstName String = "Dainie”

lastName String = "Smith"

address String = "17 Ranton"

rent double = 100.00

numOfRooms int = 2

free boolean = true

TFirstName String = ""

TFastName String = ""

H1: House

firstName String = "Alex"

lastName String = "Davison"

address String = "12 Haughton"

rent double = 120.50

numOfRooms int = 3

free boolean = false

TFirstName String = "Dainie"

TFastName String = "Smith"

L2: Lodging

firstName String = "Dan”

lastName String = "Baxter"

address String = "11 Haughton"

rent double = 49.99

roomOnly boolean = false

free boolean = true

TFirstName String = ""

TFastName String = ""

L1: Lodging

firstName String = "Eddie”

lastName String = "Smith"

address String = "13 Haughton"

rent double = 49.99

roomOnly boolean = true

free boolean = false

TFirstName String = "Alex"

TFastName String = "Davison"

## Sequence

Below are sequence diagrams for the following options:

### Add a house

: LettingAgencyMenu

: LettingsAgency

Menu option 1

h = new House()

addHouse()

addHouse(h)

: House

### Add a lodging

: LettingAgencyMenu

: LettingsAgency

Menu option 2

l = new Lodging()

addLodging()

addLodging(l)

: Lodging

### Display all free houses

outputFreeHouseDetials()

outputFreeHouseDetials()

: House

outputFreeHouseDetials()

: PrintStream

: LettingAgencyMenu

: LettingsAgency

Menu option 3

\*[instance of house]

\*[free = true]

toString()

### Display all free lodging

outputFreeLodgingDetials()

outputFreeLodgingDetials()

: Lodging

outputFreeLodgingDetials()

: PrintStream

: LettingAgencyMenu

: LettingsAgency

Menu option 4

\*[instance of lodging]

\*[free = true]

toString()

### Display all properties

: PrintStream

println()

outputPropertyDetials()

: House

outputPropertyDetials()

: LettingsAgency

: LettingAgencyMenu

Menu option 5

toString()

### Rent property

rentProperty()

: Property

p = findProperty()

: LettingsAgency

getAddress()

: LettingAgencyMenu

Menu option 6

setFree = False

setTFirstname = “*Input*”

setTSurname = “*Input*”

[p!=null]

### Re-Rent property

reRentProperty()

: Property

p = findProperty()

: LettingsAgency

getAddress()

: LettingAgencyMenu

Menu option 7

setFree = True

setTFirstname = “”

setTSurname = “”

[p!=null]

### Remove property

[p!=null]

remove(p)

Menu option 8

: LettingAgencyMenu

getAddress()

: LettingsAgency

p = findProperty()

: Property

removeProperty()

### Find property

: PrintStream

println()

findProperty()

: Property

p = findProperty()

: LettingsAgency

getAddress()

: LettingAgencyMenu

Menu option 9

toString()

[p!=null]

### Exit

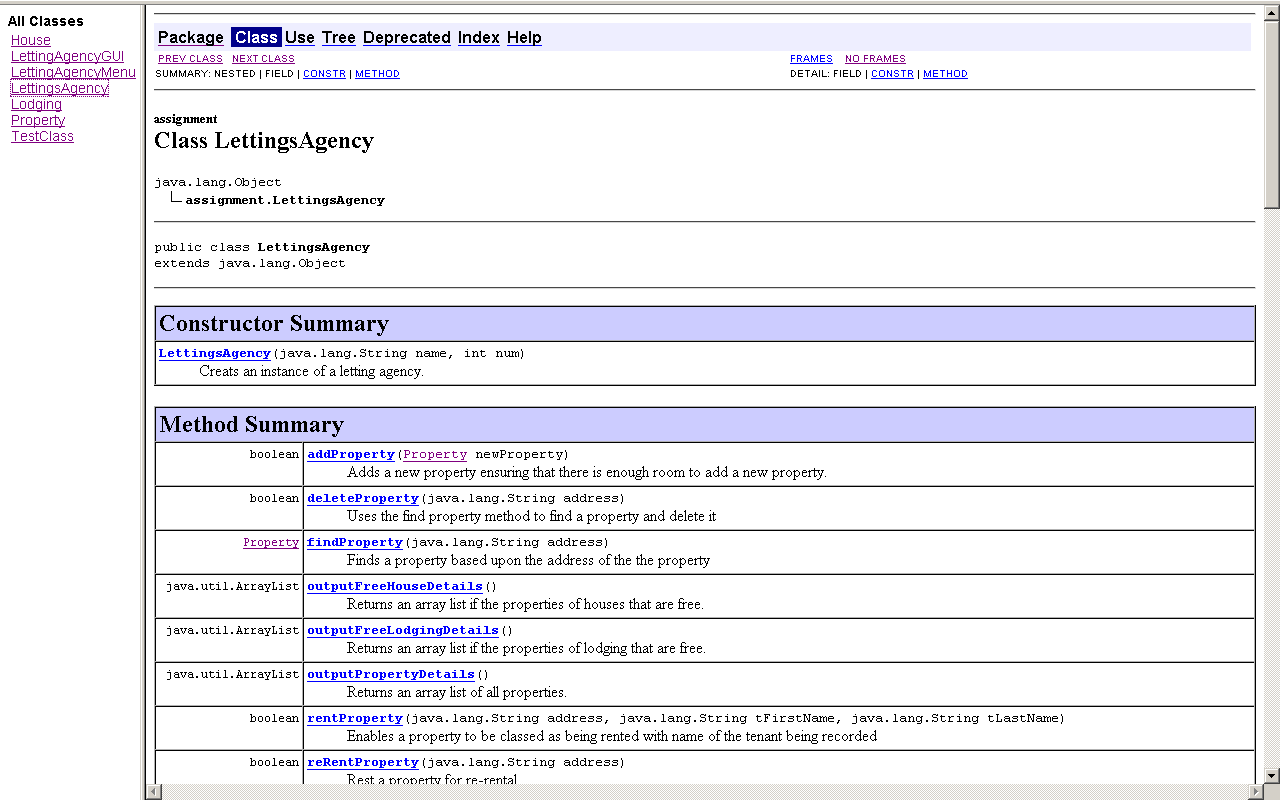
: LettingAgencyMenu

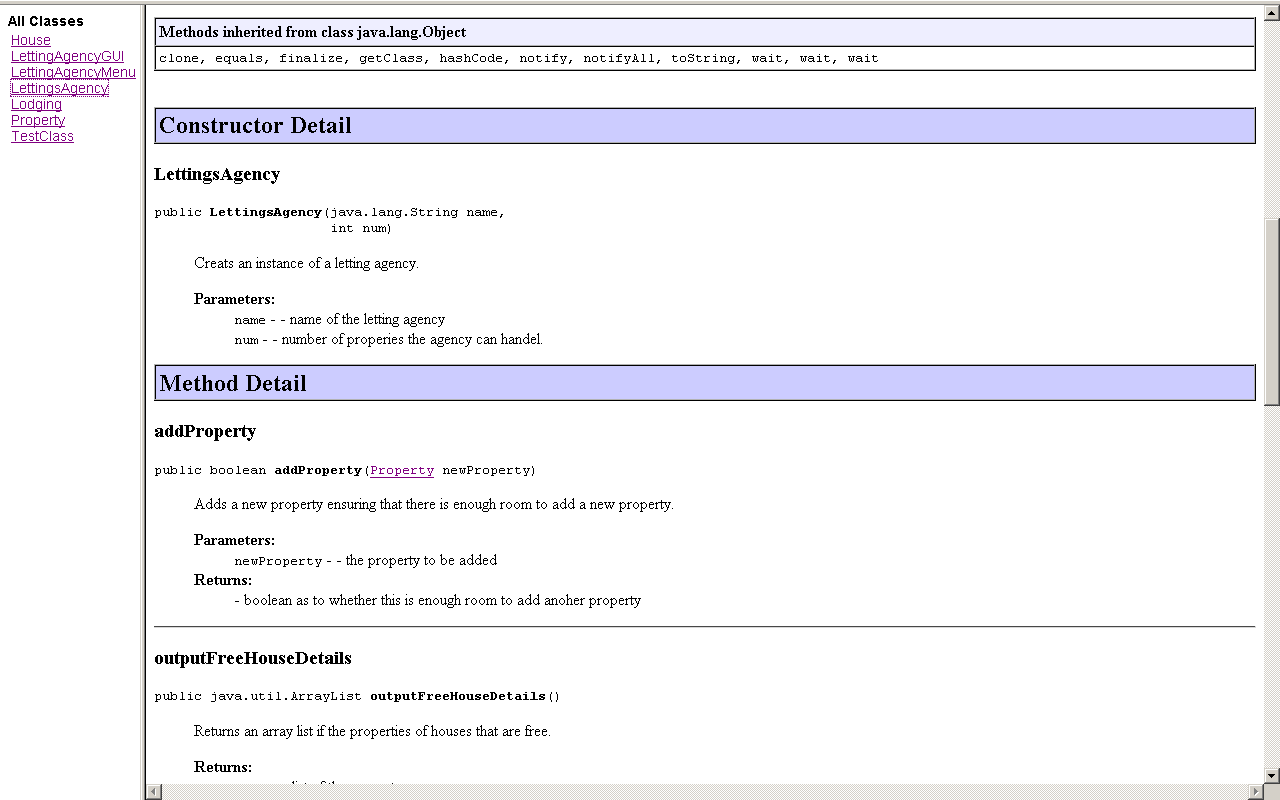
Menu option 10

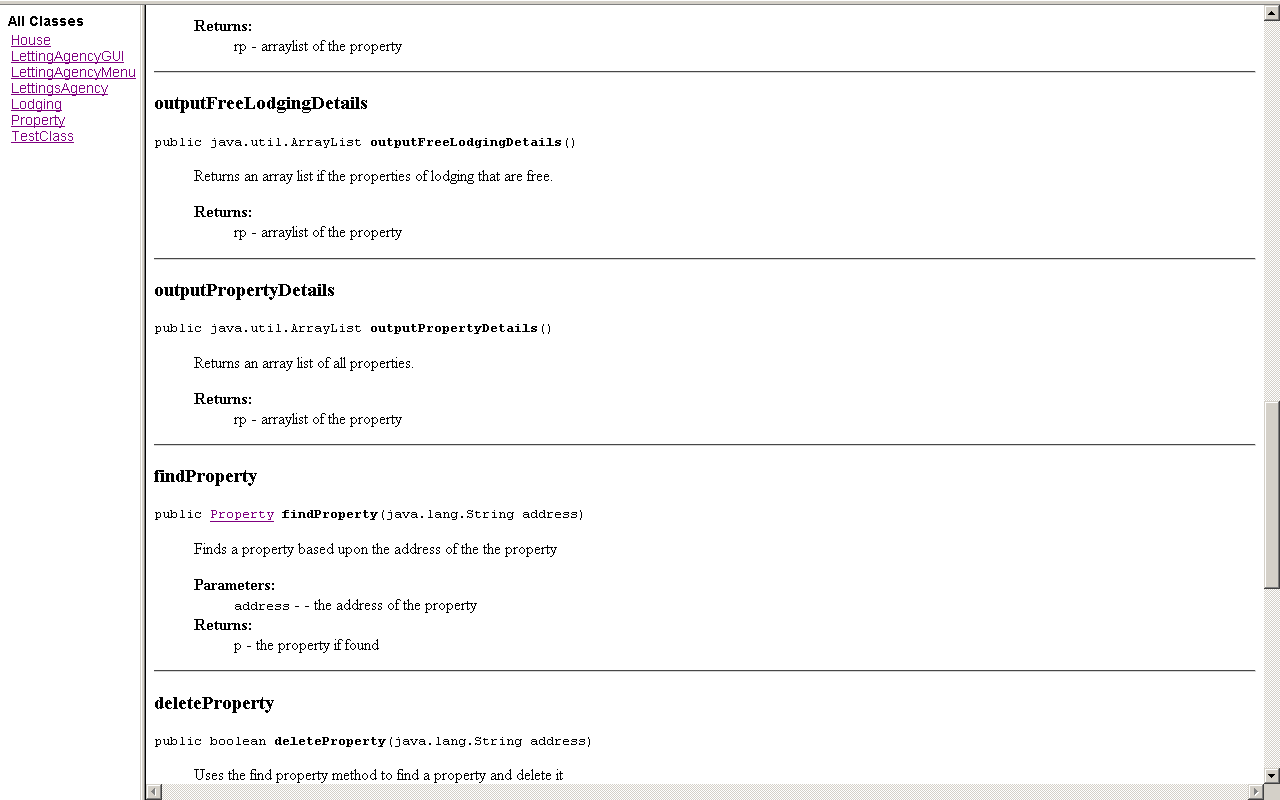
go()

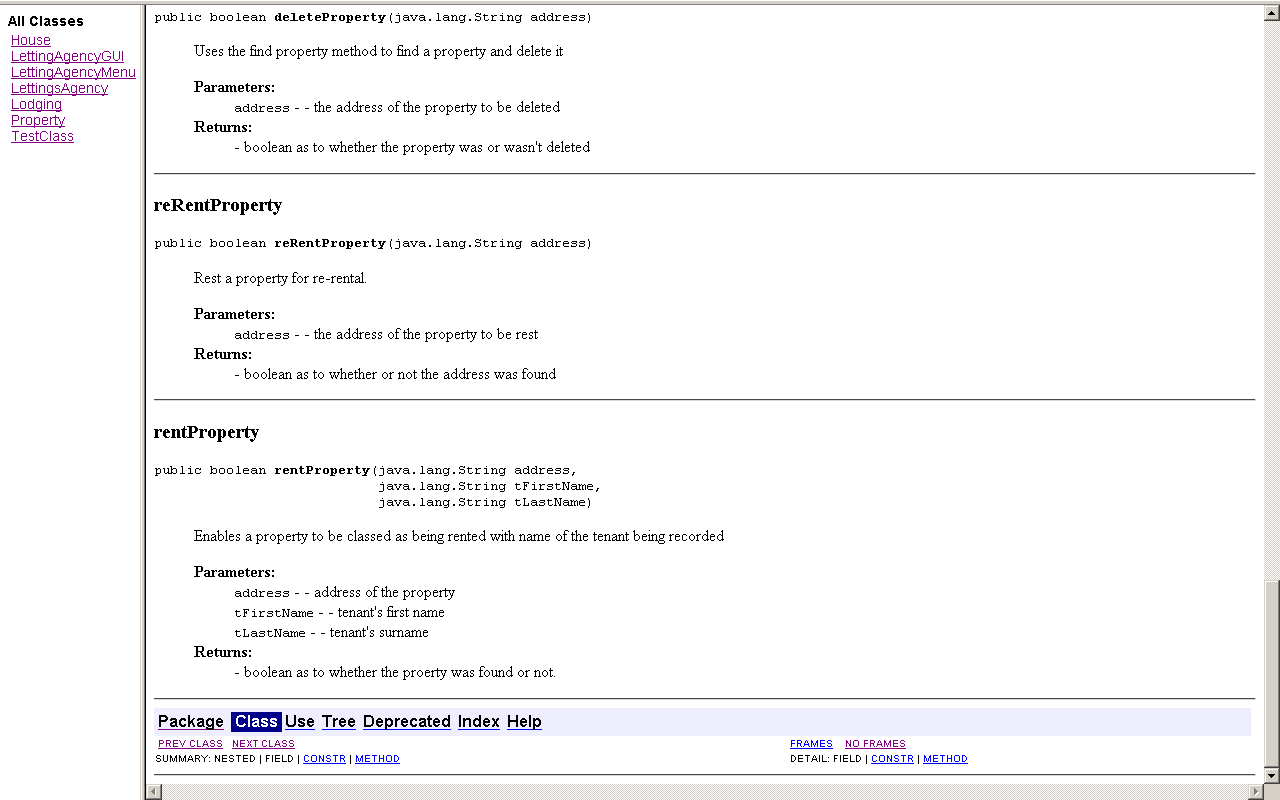
## 

## Java documentation









# Graphical User Interface

Below are details regarding the graphical user interface.

## Sequence

Below are sequence diagrams for the following options:

### Add a house

: LettingAgencyGUI

: LettiangsAgency

Menu option 1

h = new House()

addHouse()

addHouse(h)

: House

### Add a lodging

: LettingAgencyGUI

: LettingsAgency

Menu option 2

l = new Lodging()

addLodging()

addLodging(l)

: Lodging

### Display all free houses

outputFreeHouseDetials()

outputFreeHouseDetials()

: House

outputFreeHouseDetials()

: PrintStream

: LettingAgencyGUI

: LettingsAgency

Menu option 3

\*[instance of house]

\*[free = true]

toString()

### Display all free lodging

outputFreeLodgingDetials()

outputFreeLodgingDetials()

: Lodging

outputFreeLodgingDetials()

: PrintStream

: LettingAgencyGUI

: LettingsAgency

Menu option 4

\*[instance of lodging]

\*[free = true]

toString()

### Display all properties

: PrintStream

outputPropertyDetials()

outputPropertyDetials()

: House

outputPropertyDetials()

: LettingsAgency

: LettingAgencyGUI

Menu option 5

toString()

### Rent property

rentProperty()

: Property

p = findProperty()

: LettingsAgency

getAddress()

: LettingAgencyGUI

Menu option 6

setFree = False

setTFirstname = “*Input*”

setTSurname = “*Input*”

[p!=null]

### Re-rent property

reRentProperty()

: Property

p = findProperty()

: LettingsAgency

getAddress()

: LettingAgencyGUI

Menu option 7

setFree = True

setTFirstname = “”

setTSurname = “”

[p!=null]

### Remove property

remove(p)

[p!=null]

Menu option 8

: LettingAgencyGUI

getAddress()

: LettingsAgency

p = findProperty()

: Property

removetProperty()

### Find property

: PrintStream

println()

findProperty()

: Property

p = findProperty()

: LettingsAgency

getAddress()

: LettingAgencyGUI

Menu option 9

toString()

[p!=null]

### Exit

: LettingAgencyGUI

Menu option 10

go()

# Additional functionality

The main additional functionality is the data input validations, an individual example of this is that no property cannot have a blank field. All properties must have a landlord first name, surname, property address, rent value greater than zero. I have also insured that a house can not have zero number of rooms also that a lodging must have a true or false value regarding the room type.

Also other functionality include user feedback insuring that the user knows what the program is doing at all times, for example, when a house is add there is a dialog box that appears to confirm the addition of the house to the properties.

Below is some example of the extra functionality that has been added.

Figure 1 shows that the input is checked before a house is entered.

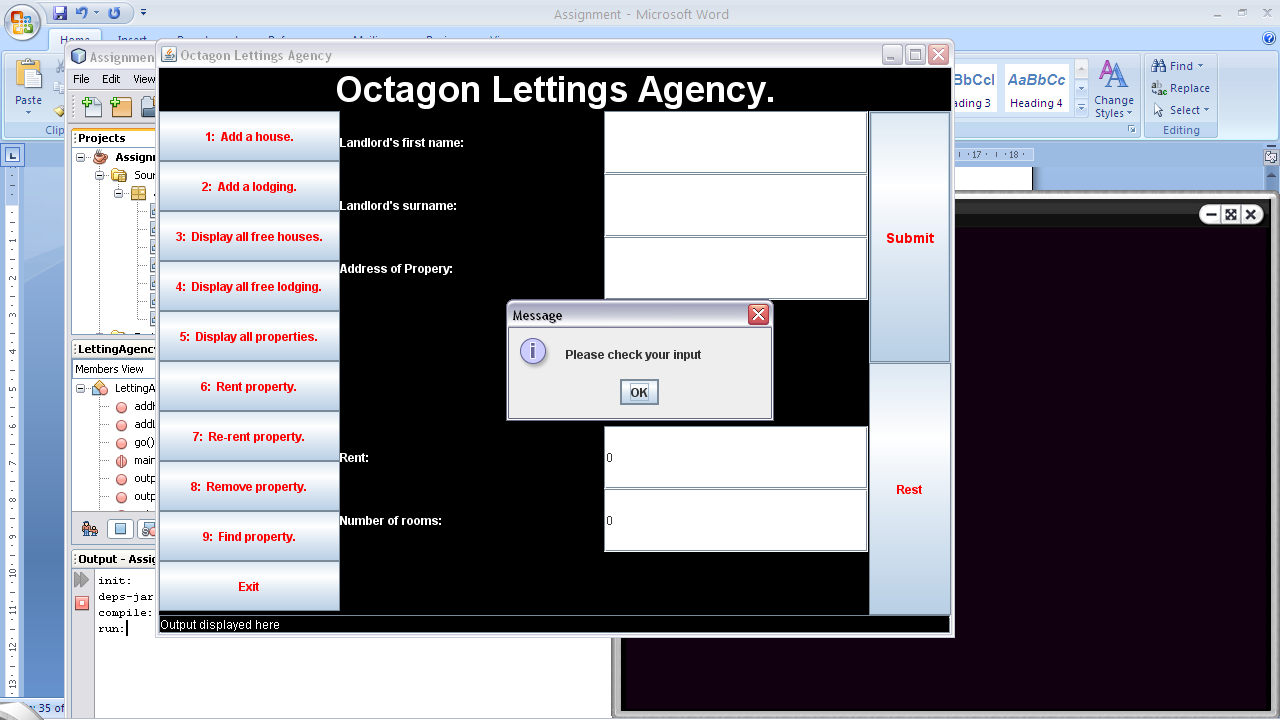


Figure 1

Figure 2 to 5 shows that there also use of message boxes to inform the users of small errors that are due to user error.



Figure 2

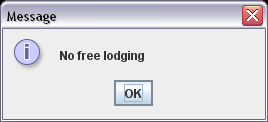


Figure 3

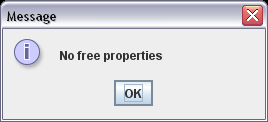


Figure 4

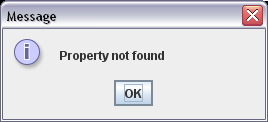


Figure 5