Question 1

A prime number is an integer that cannot be divided evenly by any integer except by 1 and by itself. Implement a class called `PrimeChecker`, with a main method, in which you:

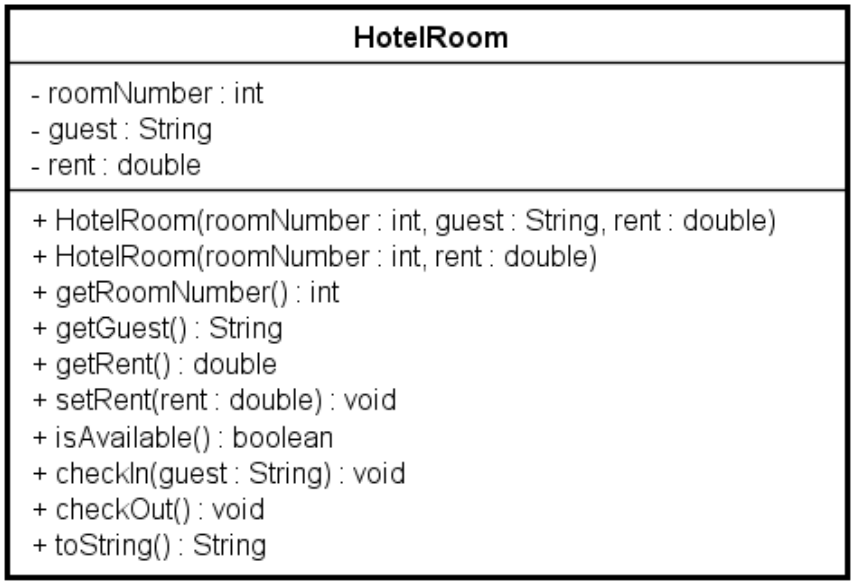
a) Use a `Scanner` to read a positive `int` from the keyboard and store it in a variable called `n`. Use a loop to continue asking the user until the entered number is positive.
b) Check whether the entered number is a prime number, and then print out "`Prime`" if it is and "`Not prime`" if it is not.

Hint: if `n` can be divided evenly by `k`, then the following is true: `n % k == 0`

Question 2

Implement a class `HotelRoom` that holds information about a hotel room (room number, the name of the guest who has rented the room, and the rent for the room). You must implement the UML class diagram below, so that all field, method, and parameter names are identical to the ones used in the diagram. The class contains the following:

a) Three instance variables (`roomNumber`, `guest`, and `rent`)
b) A three-argument constructor setting all instance variables
c) A two-argument constructor with `roomNumber` and `rent` as arguments that also sets `guest` to the value "`No one`" indicating that no guest has rented the hotel room
d) Get-methods for all three instance variables and a set-method `setRent()` that sets the value of the `rent` instance variable
e) A `boolean` method `isAvailable()` that returns `true` if the hotel room has no guest and returns `false` if there is a guest
f) A method `checkIn()` that is used to set the guest of the room if the room is available. If the room is not available then nothing should be done.
g) A method `checkOut()` that is used to remove the guest by setting the `guest` instance variable to "`No one`"
h) A method `toString()` that returns a text string with information of a hotel room, i.e. room number, rent, and only if the hotel room has a guest also the name of the guest



Question 3

Create a test class (`HotelRoomTest`) with a main method and test the class `HotelRoom` in the following way:

a) Create two `HotelRoom`-objects, one with a guest and one without a guest
b) For both `HotelRoom`-objects, print the message "`The room is for rent`" if the room is for rent and if not then print the name of the guest
c) Check out the guest of the room that has a guest
d) Check in a guest to one of the rooms
e) Ask the user to input a new rent. Rent has to be a number greater than zero, so keep asking the user until that is the case. Afterwards set the rent of one of the rooms to that value.
f) Print out all information of both the `HotelRoom`-objects using the `toString()`-method

Running the program could look like the following. The program's output is in *`italic`*, and keyboard input is in **`bold`**:

A hotelRoom1 is created with the data: 217, 742.0
A hotelRoom2 is created with the data: 1408, 1923.42, "Mike"

hotelRoom1: *`The room is for rent`*
hotelRoom2: *`The room is not for rent, the guest is Mike`*

The guest of hotelRoom2 is checked out

"Jack" has rented hotelRoom1

`Enter a new rent:` **`-4223.0`**
`Enter a new rent:` **`0.0`**
`Enter a new rent:` **`1516.0`**

Rent of hotelRoom2 is set to 1516.0

hotelRoom1: *`217, 742.0, Jack`*
hotelRoom2: *`1408, 1516.0`*