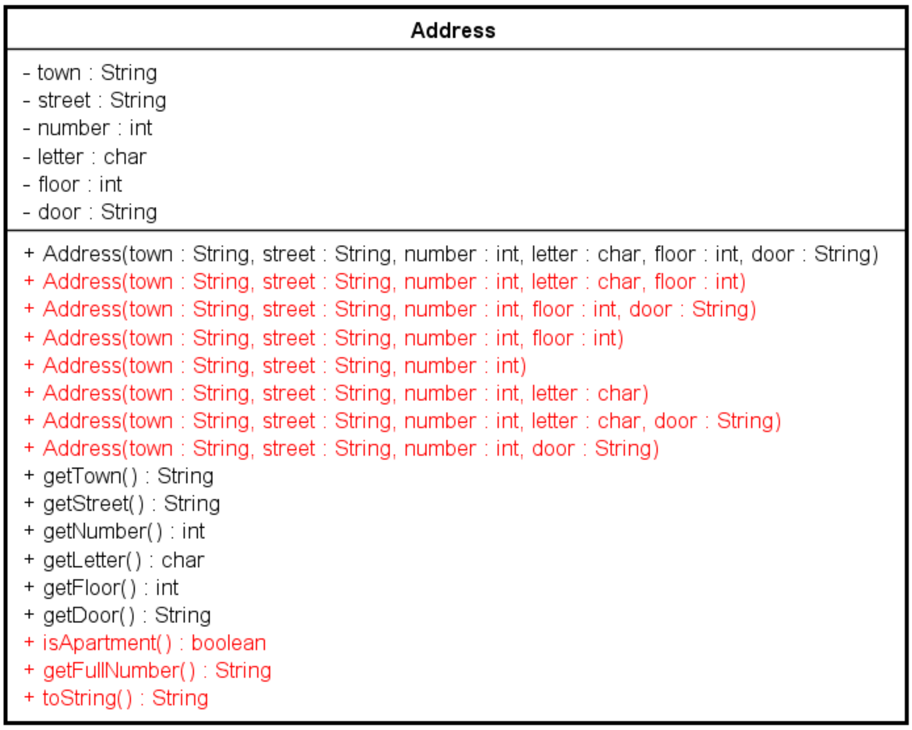
Exercise: Address, version 2



Address

- town : String
- street : String
- number : int
- letter : char
- floor : int
- door : String

+ Address(town : String, street : String, number : int, letter : char, floor : int, door : String)
+ Address(town : String, street : String, number : int, letter : char, floor : int)
+ Address(town : String, street : String, number : int, floor : int, door : String)
+ Address(town : String, street : String, number : int, floor : int)
+ Address(town : String, street : String, number : int)
+ Address(town : String, street : String, number : int, letter : char)
+ Address(town : String, street : String, number : int, letter : char, door : String)
+ Address(town : String, street : String, number : int, door : String)
+ getTown() : String
+ getStreet() : String
+ getNumber() : int
+ getLetter() : char
+ getFloor() : int
+ getDoor() : String
+ isApartment() : boolean
+ getFullNumber() : String
+ toString() : String

Create a new Module in IntelliJ and name it `Address_v2` – and copy the files from `Address_v1`.

Modify class `Address`:

a) Add 7 more constructors (see the class diagram).
   - If `letter` is not given as a parameter variable, initialize it to `'-'`
   - If `floor` is not given as a parameter variable, initialize it to `Integer.MIN_VALUE`
   - If `door` is not given as a parameter variable, initialize it to the empty string `""` or to `null`

b) Add a method `isApartment` returning true if the `floor` is given (i.e. if its value is not equal to `Integer.MIN_VALUE`)

c) Modify method `getFullNumber` with the address number (number, letter, floor and door) in a proper format. Omit the letter if it is `'-'` and omit the `floor` if it is not an apartment.

d) Modify method `toString` showing only the values given (and not `'-'` for `letter` and not `Integer.MIN_VALUE` for `floor`).

Modify the test class `AddressTest` such that you try out all 8 constructors, methods `isApartment` , `getFullNumber` and `toString`.