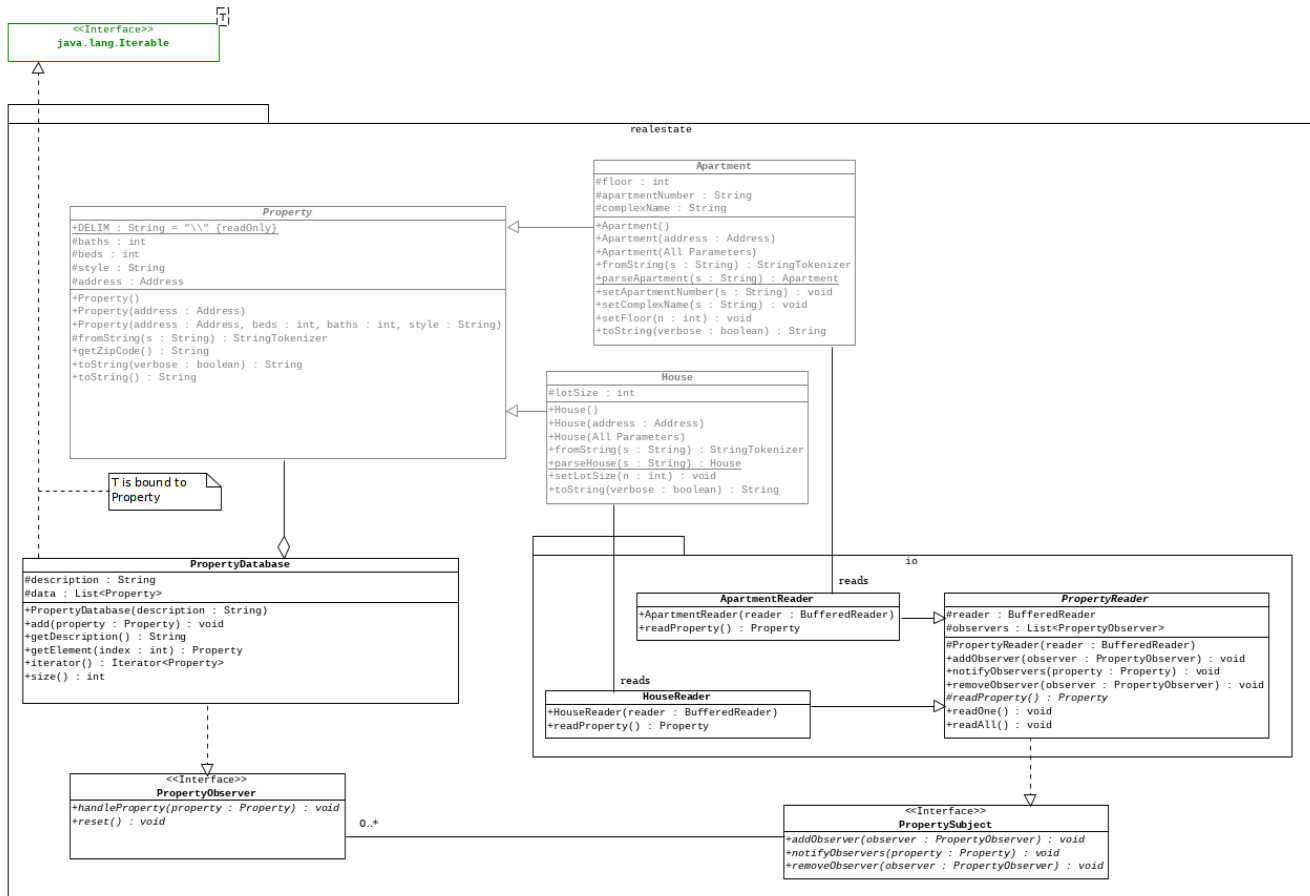


realestate Package v2.0

Class Diagram

The relationships between the various classes and interfaces of the system are illustrated in the following UML class diagram.



Classes that are shown in gray were part of v1.0 of the realestate package, the new classes (and sub-packages) are shown in black. Interfaces that are in jade green are part of the Java API.

In addition to the specifications that are contained in this class diagram, the implementation must comply with the following specifications.

The PropertySubject Interface

A `PropertySubject` is a subject (in the sense of the observer pattern) of `Property` objects. In other words, it "produces" `Property` objects.

The PropertyObserver Interface

A `PropertyObserver` is an observer (in the sense of the observer pattern) of `Property` objects. In other words, it "consumes" `Property` objects.

The PropertyDatabase Class

A `PropertyDatabase` object is a collection of `Property` objects.

Methods

```
add(Property property)
```

Must add the given `Property` to the `PropertyDatabase` if it is non-`null`, otherwise it must do nothing.

```
handleProperty(Property property)
```

Must add the given `Property` to the `PropertyDatabase` if it is non-`null`, otherwise it must do nothing.

```
reset()
```

Must remove all elements from the `PropertyDatabase`.

The PropertyReader Class

A `PropertyReader` object is a `PropertySubject` that reads one or more `Property` objects from a stream, notifying its observers as each one is read.

Methods

```
readOne()
```

Must read a single `Property` object (using the `readProperty()` method) and notify all of the observers (but only if the `Property` object is non-`null`).

```
readAll()
```

Must read all of the `Property` objects (using the `readProperty()` method, until it returns `null`) and notify all of the observers.

The ApartmentReader Class

An `ApartmentReader` object is a `PropertyReader` that reads `String` representations of `Apartment` objects.

Methods

<code>readProperty()</code>

Must read a single `String` representation of an `Apartment` from the inherited attribute named `reader`, construct a `Apartment` object from it, and return it. If the `reader` has reached the end-of-stream, then this method must return `null`.

This method must assume that there is one `String` representation per "line" in the stream, and that the "line" contains a verbose `String` representation of an `Apartment` object.

The HouseReader Class

A `HouseReader` object is a `PropertyReader` that reads `String` representations of `House` objects.

Methods

<code>readProperty()</code>

Must read a single `String` representation of a `House` from the inherited attribute named `reader`, construct a `House` object from it, and return it. If the `reader` has reached the end-of-stream then this method must return `null`.

This method must assume that there is one `String` representation per "line" in the stream, and that the "line" contains a verbose `String` representation of a `House` object.