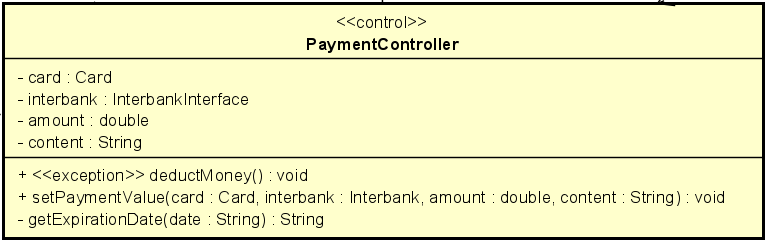
**CLASS DESIGN**

1. **Class “PaymentController”**



***Attribute***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | card | Card | NULL | Represent the card used for payment |
| 2 | amount | double | NULL | Represent total amount of the transaction |
| 3 | content | String | NULL | Represent content of transaction |
| 4 | interbank | InterbankInterface | NULL | Represent the interbank |

***Operation***

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description (purpose) |
| 1 | deductMoney | void | Send request to interbank API to deduct money from card to pay for renting bike |
| 2 | setPaymentValue | void | Set value for all attributes of *PaymentController* class |

*Parameter*:

* card – the credit card used for payment
* interbank – interbank of transaction
* amount – total amount of transaction
* content – content of transaction

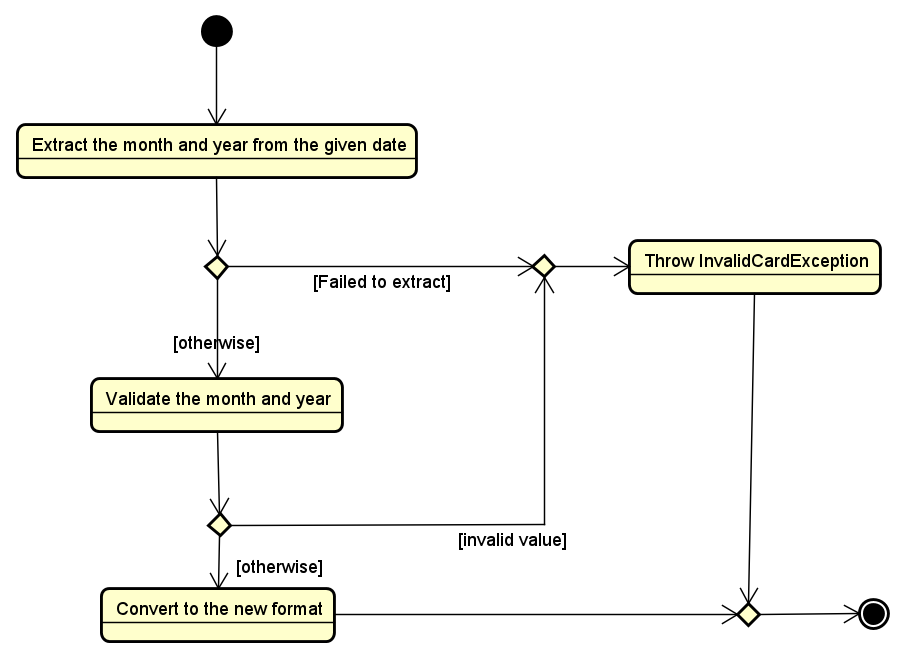
*Exception*:

* PaymentException: if responded with error that transaction is failed

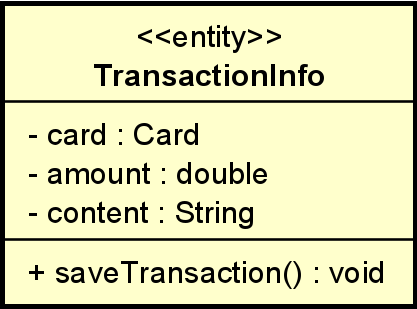
***State*** None

***Method***

* getExpirationDate: given the String “date” representing the expiration date in the format “”mm/yy”, this method convert it into the required format “mmyy”. The algorithm is illustrated as follows.



1. **Class “TransactionInfo”**



***Attribute***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | card | Card | NULL | Represent the card used for payment |
| 2 | amount | double | NULL | Represent total amount of the transaction |
| 3 | content | String | NULL | Represent content of transaction |

***Operation***

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description (purpose) |
| 1 | saveTransaction | void | Save the transaction between software and interbank |

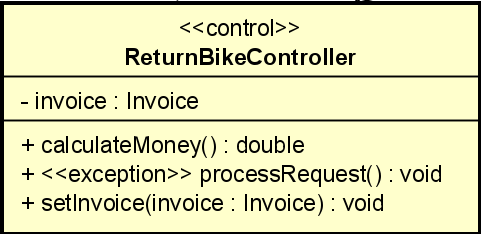
*Parameter*: None

*Exception*: None

***Method*** None

***State*** None

1. **Class “ReturnBikeController”**



***Attribute***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | invoice | Invoice | NULL | Represent invoice of returning bike |

***Operation***

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description (purpose) |
| 1 | calculateMoney | double | Calculate total amount that customer has to pay when return bike |
| 2 | processRequest | void | Process returning bike request, call *ReturnBikeHandler* and *InvoiceHandler* class |
| 3 | setInvoice | void | Set value for invoice after calculate money |

*Parameter*:

* Invoice – invoice of returning bike

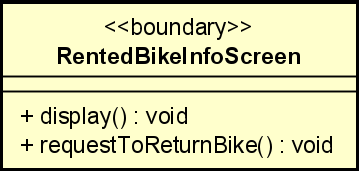
*Exception*:

* ReturnBikeException: if responded with error that return bike request is failed

***State*** None

***Method*** None

1. **Class “RentedBikeInfoScreen”**



***Attribute***

***Operation***

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description (purpose) |
| 1 | display | void | Display detail of renting bike include amount up to now that customer has to pay |
| 2 | requestToReturnBike | void | Request to return bike |

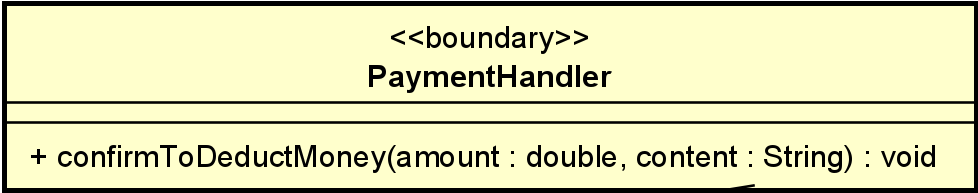
*Parameter*: None

*Exception*: None

***Method*** None

***State*** None

1. **Class “PaymentHandler”**



***Attribute***

***Operation***

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description (purpose) |
| 1 | confirmToDeductMoney | void | Confirm to deduct money from card to pay for renting bike |

*Parameter*:

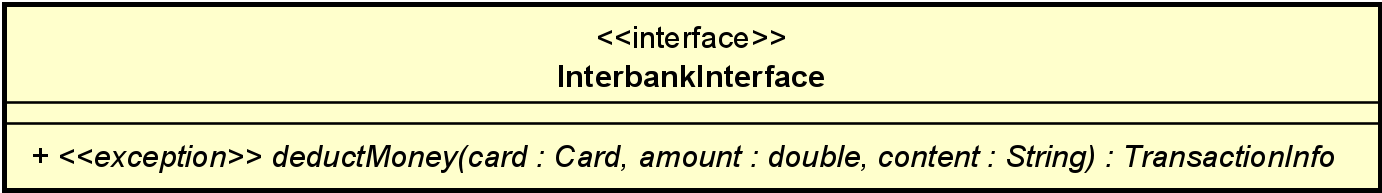
* amount: total amount that interbank will deduct from customer’s card
* content: content of transaction

*Exception*: None

***Method*** None

***State*** None

1. **Class “InterbankInterface”**



***Attribute***

***Operation***

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description (purpose) |
| 1 | deductMoney | void | Confirm to deduct money from card to pay for renting bike |

*Parameter*:

* amount: total amount that interbank will deduct from customer’s card
* content: content of transaction
* card : the credit card used for payment

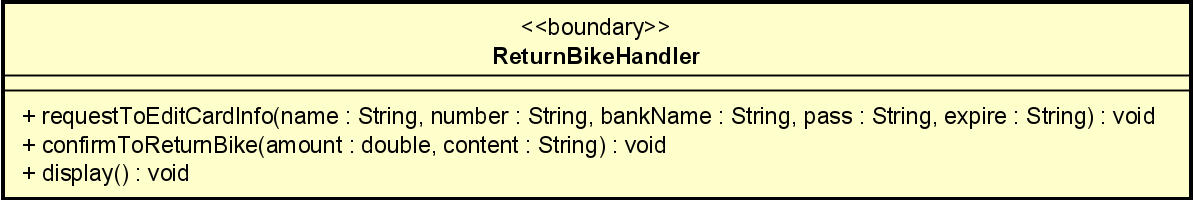
*Exception*:

* InterbankPaymentException: if responded with a pre-defined error code
* UnrecognizeException: if responded with an unknown error code or something goes wrong

***Method*** None

***State*** None

1. **Class “ReturnBikeHandler”**



***Attribute***

***Operation***

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description (purpose) |
| 1 | requestToEditCardInfo | void | Edit card information |
| 2 | confirmToReturnBike | void | Confirm to return bike |
| 3 | display | void | Display detailed information of return bike and card information of customer |

*Parameter*:

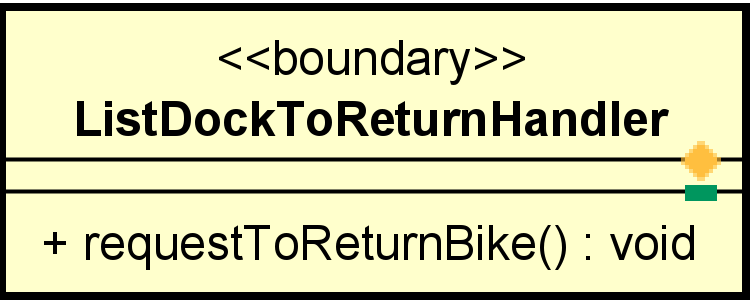
* amount: total amount that customer has to pay for returning bike
* content: content of transaction
* name: card’s holder name
* number: card number
* bankName: name of interbank
* pass: security code of card
* expire: expiration date of card

*Exception*: None

***Method*** None

***State*** None

1. **Class “ListDockToReturnHandler”**



**Attribute**

**Operation**

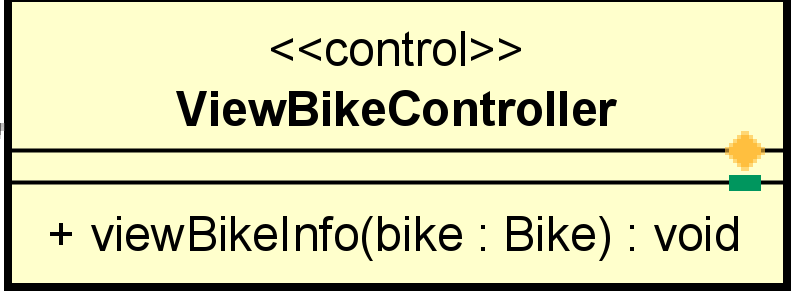
|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description (purpose) |
| 1 | requestToReturnBike | void | Request to return bike, call selectDockMakerController |

**Parameter**  none

**Method** none

**State** none

1. **Class “ViewBikeController”**



**Attribute** none

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description (purpose) |
| 1 | viewBikeInfo | void | Process request view specific bike info |

Parameter

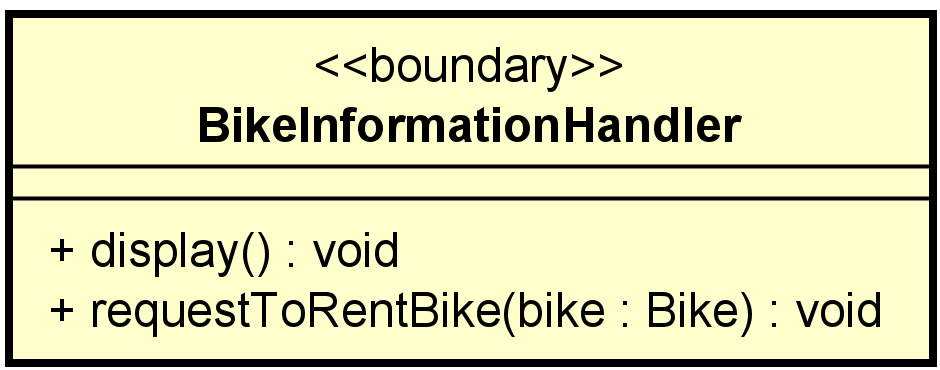
* bike: bike that want to view information

Exception : ViewBikeException

**Method** none

**State** none

1. **Class “BikeInformationHandler”**



**Attribute** none

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description (purpose) |
| 1 | display | void | Display bike information |
| 2 | requestToRentBike | Void | When user submit to rent bike, sent request to rentBikeController |

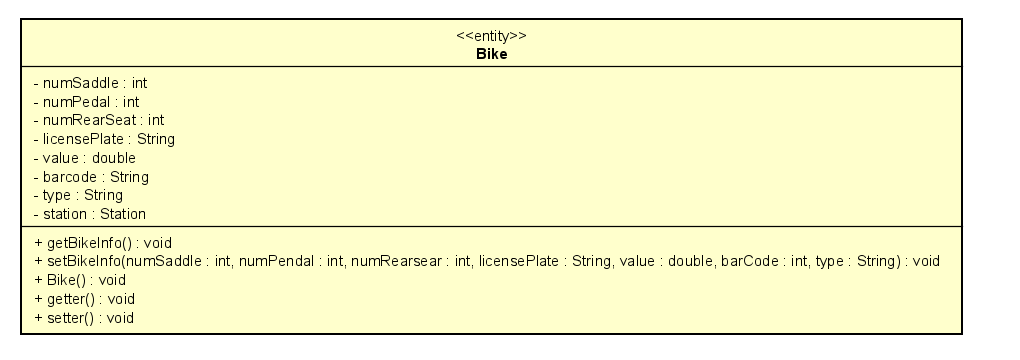
Parameter

* bike: bike that user want to rent

**Method** none

**State** none

1. **Class “Bike”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | numSaddle | int | NULL | Number saddle of the bike |
| 2 | numPedal | int | NULL | Number pedal of the bike |
| 3 | numRearSeat | Int | NULL | Number rear seat of the bike |
| 4 | licensePlate | String | NULL | Represent license plate of the bike |
| 5 | value | double | NULL | Represent value of the bike |
| 6 | barcode | String | NULL | Represent barcode of the bike |
| 7 | type | String | NULL | Represent type of the bike |
| 8 | station | Station | NULL | Represent bike in which station |

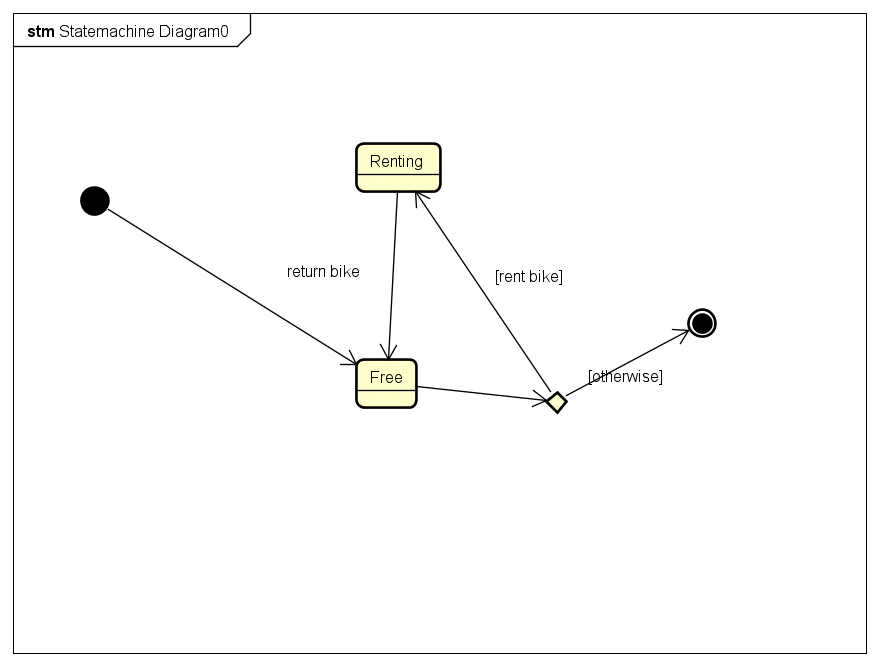
**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description (purpose) |
| 1 | getBikeInfo | void | Get bike information for display |
| 2 | setBikeInfo | void | Set bike information |
| 3 | Bike | void | Constructor |
| 4 | getter | void | Get all attribute in acronym |
| 5 | setter | void | Set value for each attribute in acronym |

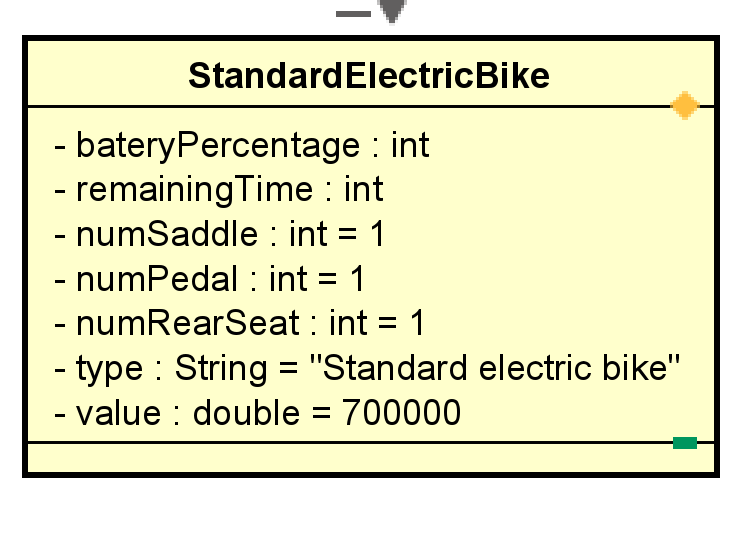
Parameter : same like attribute

**Method** none

**State**



1. **Class “StandardElectricBike”**



**Attribute**

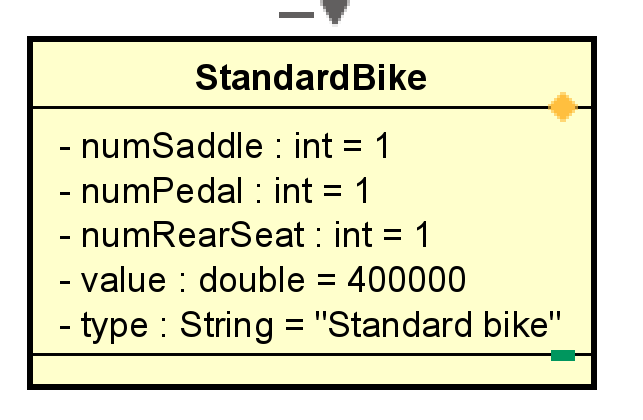
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | numSaddle | int | 1 | Unchanged value |
| 2 | numPedal | int | 1 | Constant value |
| 3 | numRearSeat | int | 1 | Constant value |
| 4 | value | double | 700000 | Constant value |
| 5 | type | String | “Standard electric bike” | Constant type |

**Operation** inherit

**Method** none

**State** none

1. **Class “Standard bike”**



***Attribute***

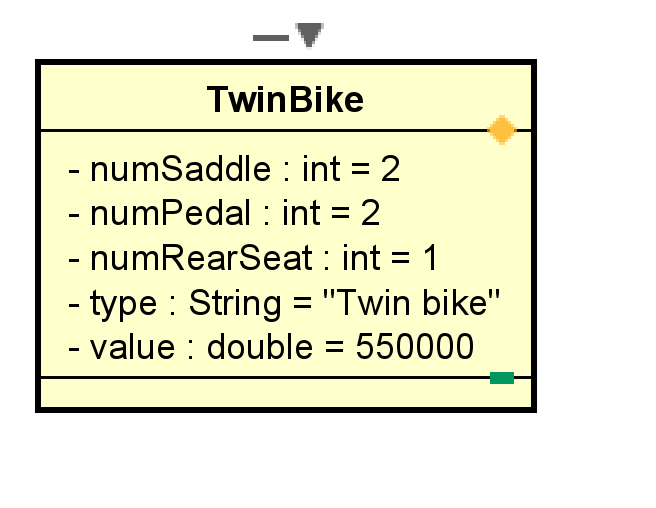
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | numSaddle | int | 1 | Unchanged value |
| 2 | numPedal | int | 1 | Constant value |
| 3 | numRearSeat | int | 1 | Constant value |
| 4 | value | double | 400000 | Constant value |
| 5 | type | String | “Standard bike” | Constant type |

**Operation** inherit

**Method** none

**State** none

1. **Class “TwinBike”**



***Attribute***

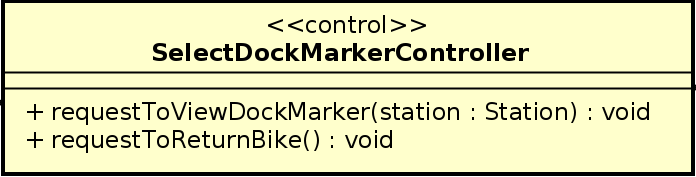
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | numSaddle | int | 2 | Unchanged value |
| 2 | numPedal | int | 2 | Constant value |
| 3 | numRearSeat | int | 1 | Constant value |
| 4 | value | double | 550000 | Constant value |
| 5 | type | String | “Twin bike” | Constant type |

**Operation** inherit

**Method** none

**State** none

1. **Class “SelectDockMarkerController”**



**Attribute** none

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description (purpose) |
| 1 | requestToViewDockMarker | void | Process request to view the dock marker |
| 2 | RequestToReturnBike | void | Process to request to return the bike |

Parameter

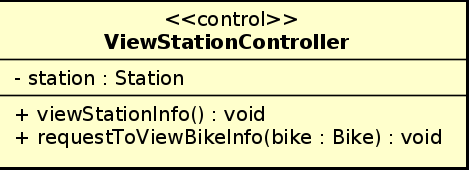
* station: station that want to view information

Exception : ViewStationException

**Method** none

**State** none

1. **Class “ViewStationController”**



**Attribute** station: station that user want to view information

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description (purpose) |
| 1 | viewStationInfo | void | View station information |
| 2 | requestToViewBikeInfo | Void | Process to request to view bike information |

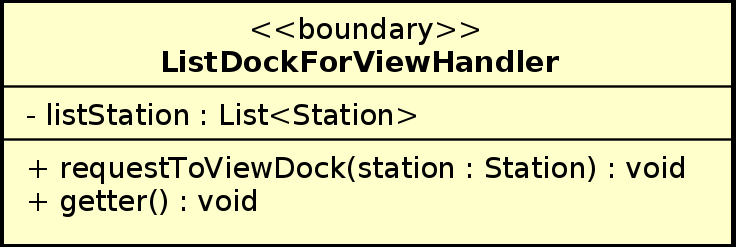
Parameter

* bike: bike that user want to rent

**Method** none

**State** none

1. **Class “ListDockForViewHandler”**



**Attribute** listStation: a list of stations

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description (purpose) |
| 1 | requestToViewDock | void | Process to request to view dock |
| 2 | getter | Void | get method |

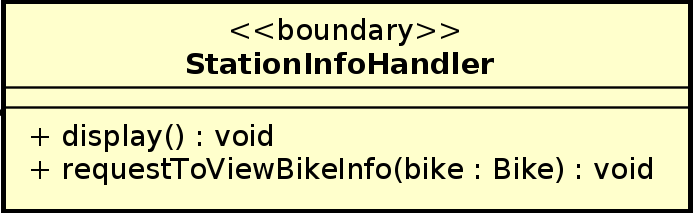
Parameter

* station: station that user want to view information

**Method** none

**State** none

1. **Class “StationInfoHandler”**



**Attribute** none

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description (purpose) |
| 1 | display | void | Display station information |
| 2 | requestToViewBikeInfo | Void | Process to request to view bike information |

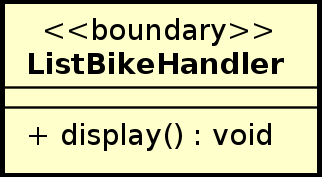
Parameter

* bike: bike that user want to view information

**Method** none

**State** none

1. **Class “ListBikeHandler”**



**Attribute** none

**Operation**

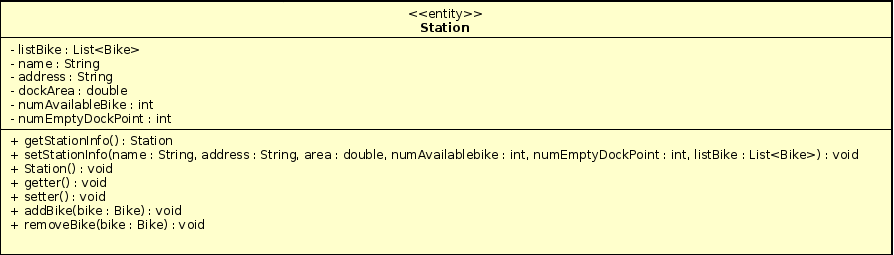
|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description (purpose) |
| 1 | display | void | Display list of bikes |

Parameter none

**Method** none

**State** none

1. **Class “Station”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | listBike | List<Bike> | NULL | List the bikes in the station |
| 2 | name | String | NULL | Name of the station |
| 3 | address | String | NULL | Address of the station |
| 4 | dockArea | double | NULL | Area of the dock |
| 5 | numAvailableBike | int | NULL | Number of available bikes |
| 6 | numEmptyDockPoint | int | NULL | Number of empty dock points |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description (purpose) |
| 1 | getStationInfo | Station | Get station information for display |
| 2 | setStationInfo | void | Set station information |
| 3 | Station | void | Constructor |
| 4 | getter | void | Get all attribute in acronym |
| 5 | setter | void | Set value for each attribute in acronym |
| 6 | addBike | void | Add bike to the station if user return bike |
| 7 | removeBike | void | Remove bike in the station if user rent bike |

Parameter : same like attribute

**Method** none

**State** none