PayStation Main
Design Document

CIS 3296 Section 05
Spring 2022

Team Members:
- Arthur Kozhevnik
- Mary Kate

Repository URL:
- https://github.com/cis3296s22/paystationmain-06-durnan-kozhevnik

Table of Contents

Document Overview

This Design Document describes the software architecture and how the requirements are mapped into the design. This document will be a combination of diagrams and text that is describing what the diagrams are showing. The Design Document also specify the complete design of the software implementation using Javadoc.

Architecture

This section describes the different components and their interfaces using UML. For example: client, server, database. For each component provide class diagrams showing the classes to be developed (or used) and their relationship.

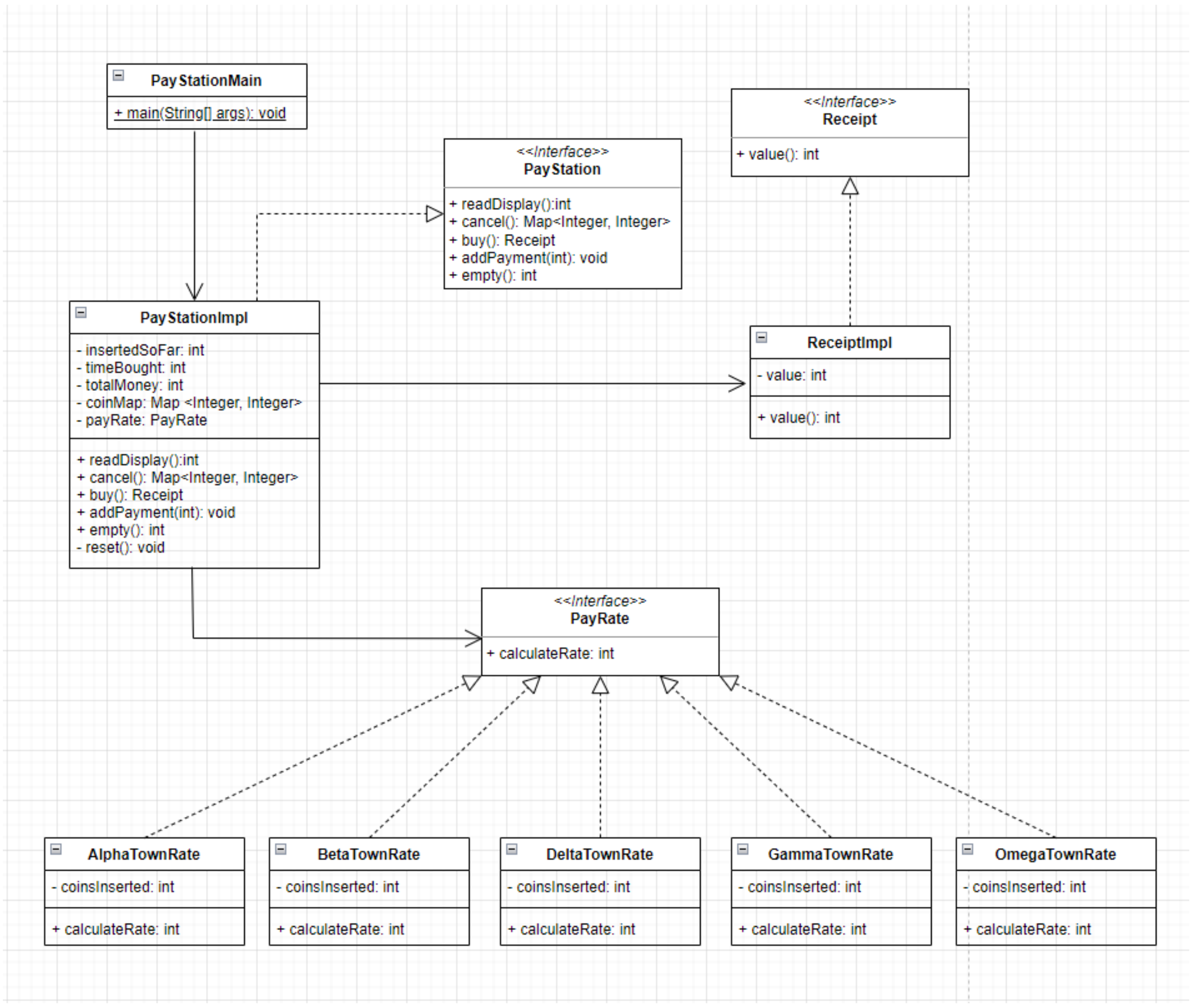


Figure 1 UML Example Class Diagram for PayStation TDD

Detail Design API

For each class define the data fields, methods.

- The purpose of the class.
- The purpose of each data field.
- The purpose of each method
- Pre-conditions if any.
- Post-conditions if any.
- Parameters and data types
- Return value and output variables
- Exceptions thrown*.

This information should be in structured comments (e.g. Javadoc) in the source files. A documentation generation tool (e.g. Javadoc) may be used to generate the document as a draft.

Package edu.temple.cis.paystation

```
package edu.temple.cis.paystation
```

All Classes and Interfaces **Interfaces** **Classes** **Exceptions**

Class	Description
IllegalCoinException	
PayStation	
PayStationImpl	Implementation of the pay station.
Receipt	
ReceiptImpl	

Class IllegalCoinException

java.lang.Object java.lang.Throwable java.lang.Exception
edu.temple.cis.paystation.IllegalCoinException

All Implemented Interfaces:

Serializable

```
public class IllegalCoinException
extends Exception
```

See Also:

Serialized Form

Constructor Summary

Constructors

Constructor	Description
`IllegalCoinException(String e)`	

Method Summary

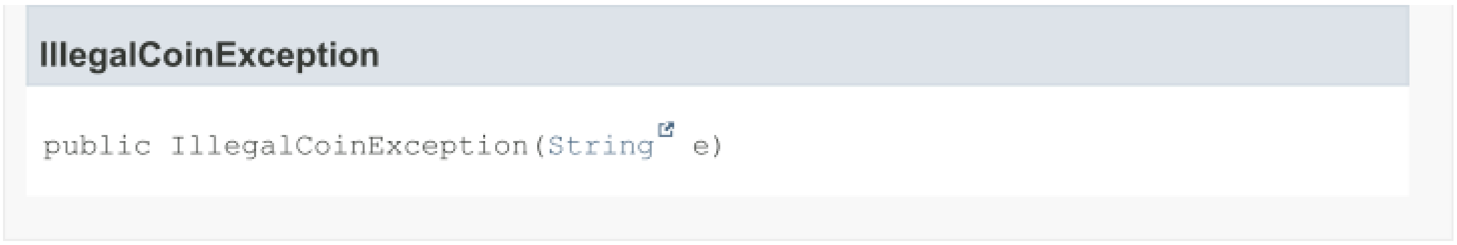
Methods inherited from class java.lang.Throwable

addSuppressed, fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, getSuppressed, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Details



IllegalCoinException

```
public IllegalCoinException(String e)
```

Interface PayStation

All Known Implementing Classes:

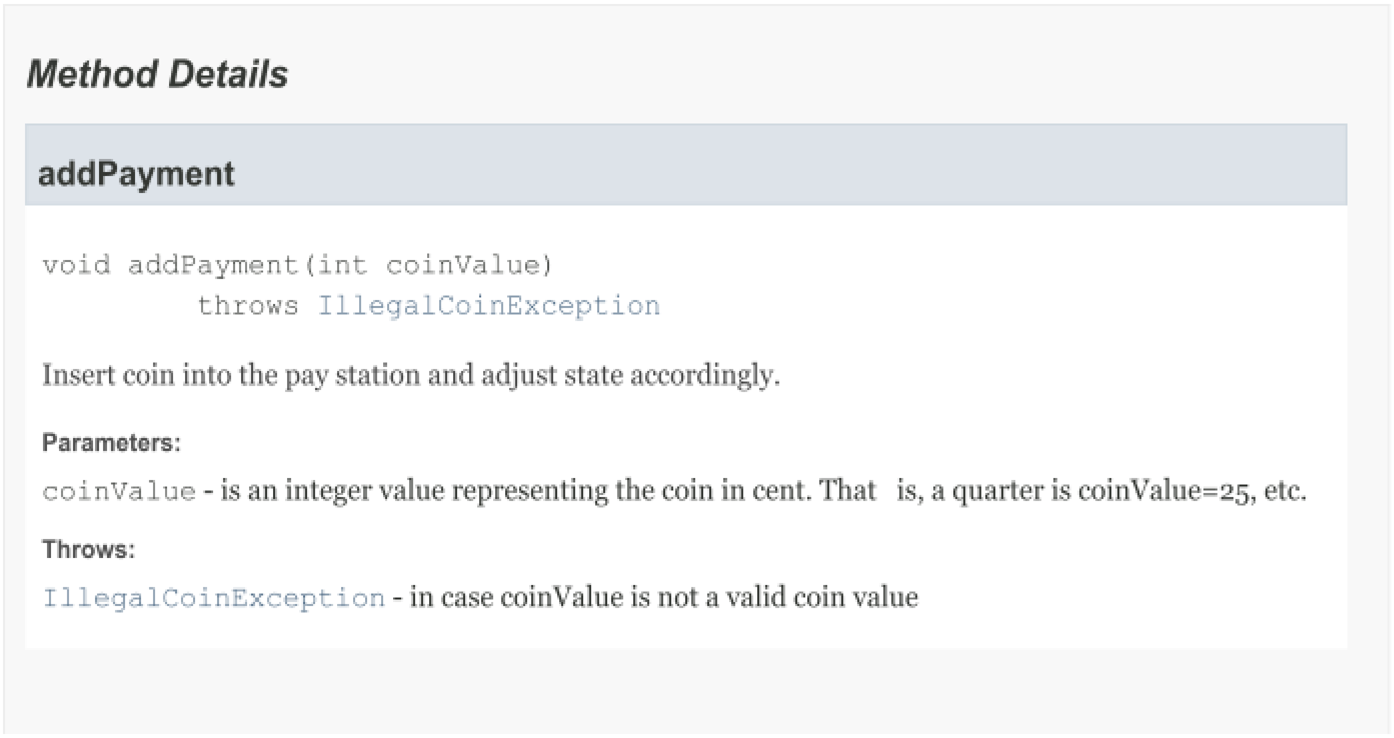
`PayStationImpl`

`public interface **PayStation**`

Method Summary

All Methods | **Instance Methods** | **Abstract Methods**

Modifier and Type	Method	Description
`void`	`addPayment(int coinValue)`	Insert coin into the pay station and adjust state accordingly.
`Receipt`	`buy()`	Buy parking time.
`Map<Integer,Integer>`	`cancel()`	Cancel the present transaction.
`int`	`empty()`	Reset money collected.
`int`	`readDisplay()`	Read the machine's display.



Method Details

addPayment

```
void addPayment(int coinValue)
         throws IllegalCoinException
```

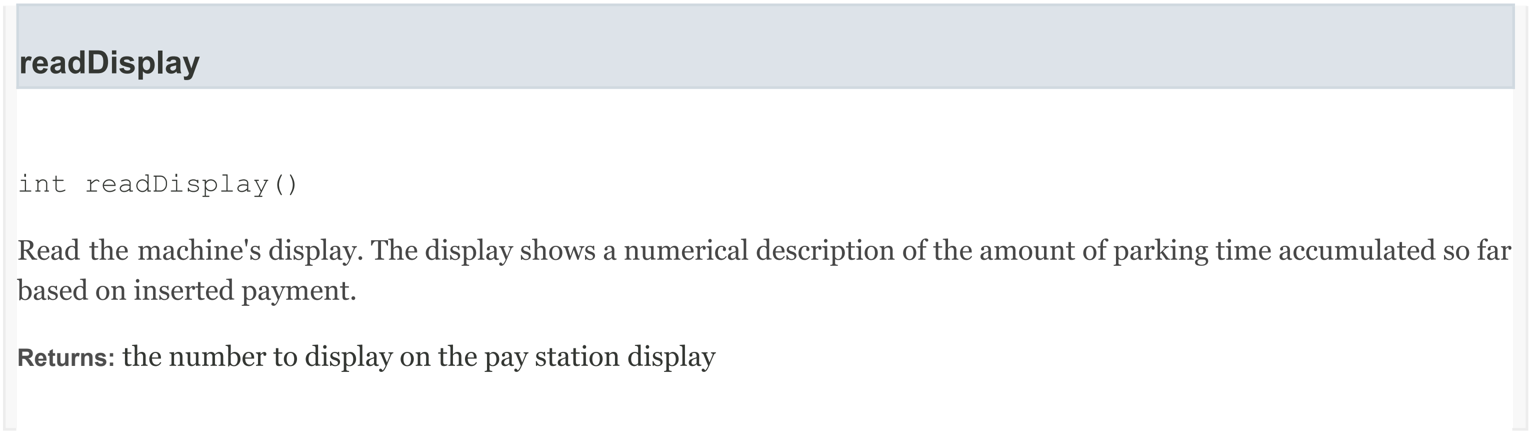
Insert coin into the pay station and adjust state accordingly.

Parameters:

`coinValue` - is an integer value representing the coin in cent. That is, a quarter is coinValue=25, etc.

Throws:

`IllegalCoinException` - in case coinValue is not a valid coin value

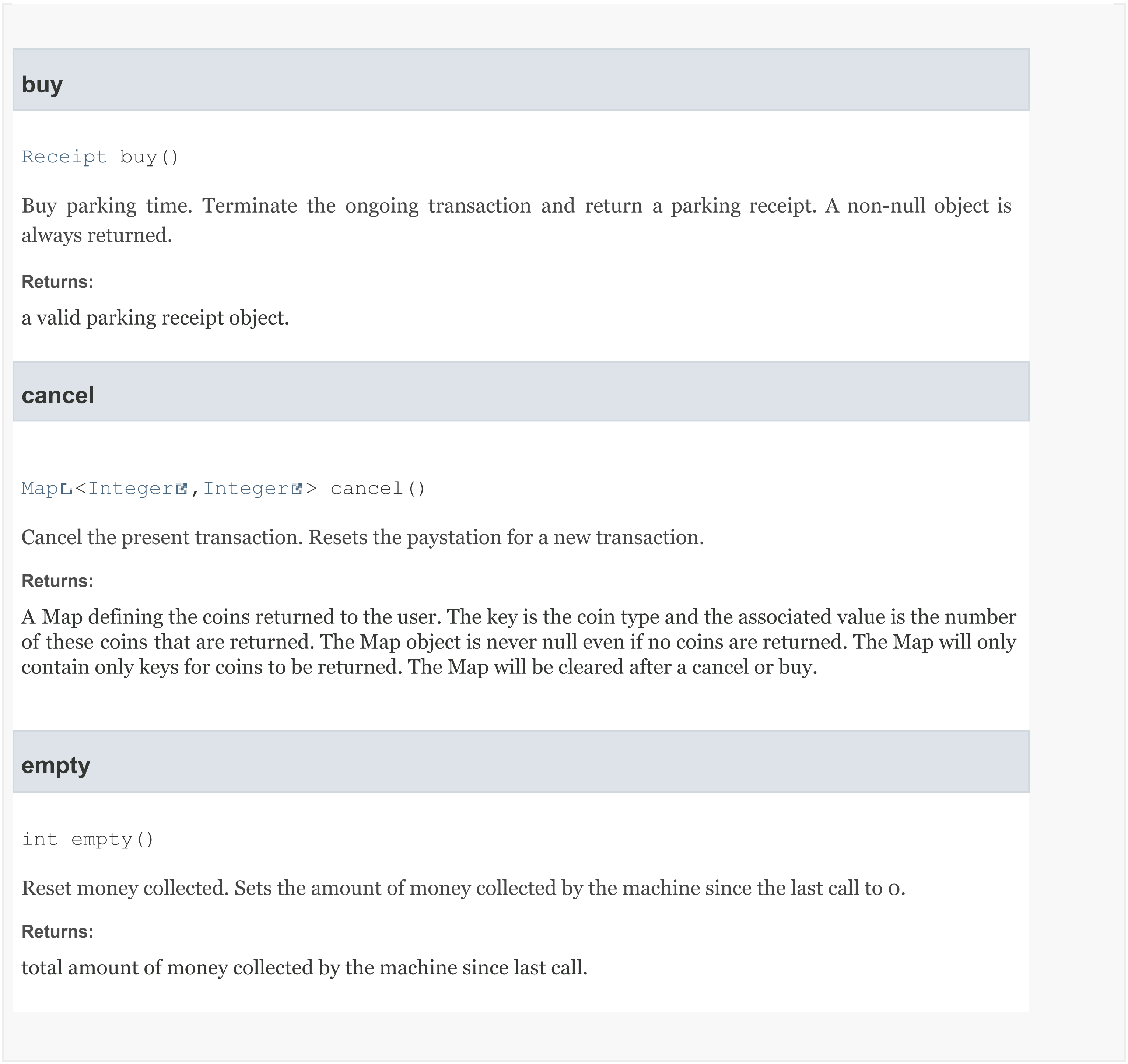


readDisplay

```
int readDisplay()
```

Read the machine's display. The display shows a numerical description of the amount of parking time accumulated so far based on inserted payment.

Returns: the number to display on the pay station display



buy

```
Receipt buy()
```

Buy parking time. Terminate the ongoing transaction and return a parking receipt. A non-null object is always returned.

Returns:

a valid parking receipt object.

cancel

```
Map<Integer,Integer> cancel()
```

Cancel the present transaction. Resets the paystation for a new transaction.

Returns:

A Map defining the coins returned to the user. The key is the coin type and the associated value is the number of these coins that are returned. The Map object is never null even if no coins are returned. The Map will only contain only keys for coins to be returned. The Map will be cleared after a cancel or buy.

empty

```
int empty()
```

Reset money collected. Sets the amount of money collected by the machine since the last call to 0.

Returns:

total amount of money collected by the machine since last call.

Class PayStationImpl

java.lang.Object edu.temple.cis.paystation.PayStationImpl

All Implemented Interfaces:

PayStation

```
public class PayStationImpl
extends Object
implements PayStation
```

Implementation of the pay station. Responsibilities: 1) Accept payment; 2) Calculate parking time based on payment; 3) Know earning, parking time bought; 4) Issue receipts; 5) Handle buy and cancel events. This source code is from the book "Flexible, Reliable Software: Using Patterns and Agile Development" published 2010 by CRC Press. Author: Henrik B Christensen Computer Science Department Aarhus University This source code is provided WITHOUT ANY WARRANTY either expressed or implied. You may study, use, modify, and distribute it for non-commercial purposes. For any commercial use, see http://www.baerbak.com/

Constructor Summary

Constructors

Constructor	Description
`PayStationImpl()`	

Method Summary

All Methods **Instance Methods** **Concrete Methods**

Modifier and Type	Method	Description
`void`	`addPayment(int coinValue)`	Insert coin into the pay station and adjust state accordingly.
`Receipt`	`buy()`	Buy parking time.
`Map<Integer,Integer>`	`cancel()`	Cancel the present transaction.
`int`	`empty()`	Reset money collected.

int	**readDisplay**()	Read the machine's display.

Methods inherited from class java.lang.Object

`clone , equals , finalize , getClass, hashCode, notify, notifyAll, toString , wait , wait , wait`

Constructor Details

PayStationImpl

```
public PayStationImpl()
```

Method Details

addPayment

```
public void addPayment(int coinValue)
throws IllegalCoinException
```

Description copied from interface: `PayStation`

Insert coin into the pay station and adjust state accordingly.

Specified by: `addPayment` in interface `PayStation`

Parameters:

`coinValue` - is an integer value representing the coin in cent. That is, a quarter is coinValue=25, etc.

Throws:

`IllegalCoinException` - in case coinValue is not a valid coin value

readDisplay

```
public int readDisplay()
```

Description copied from interface: `PayStation`

Read the machine's display. The display shows a numerical description of the amount of parking time accumulated so far based on inserted payment.

Specified by:

`readDisplay` in interface `PayStation`

Returns: the number to display on the pay station display

buy

```
public Receipt buy()
```

Description copied from interface: `PayStation`

Buy parking time. Terminate the ongoing transaction and return a parking receipt. A non-null object is always returned.

Specified by:

`buy` in interface `PayStation`

Returns:

a valid parking receipt object.

cancel

```
public Map<Integer,Integer> cancel()
```

Description copied from interface: `PayStation`

Cancel the present transaction. Resets the paystation for a new transaction.

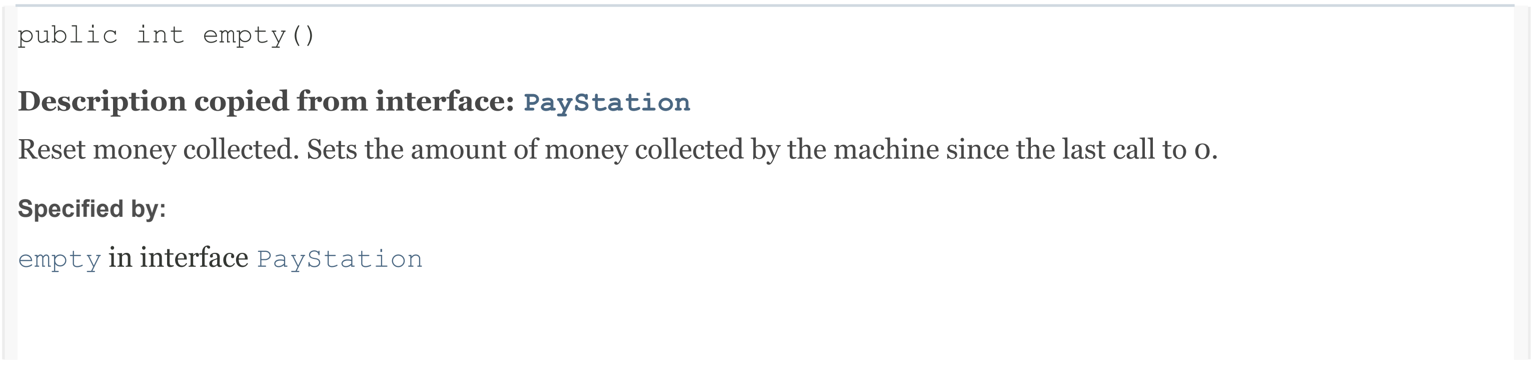
Specified by:

`cancel` in interface `PayStation`

Returns:

A Map defining the coins returned to the user. The key is the coin type and the associated value is the number of these coins that are returned. The Map object is never null even if no coins are returned. The Map will only contain only keys for coins to be returned. The Map will be cleared after a cancel or buy.

empty



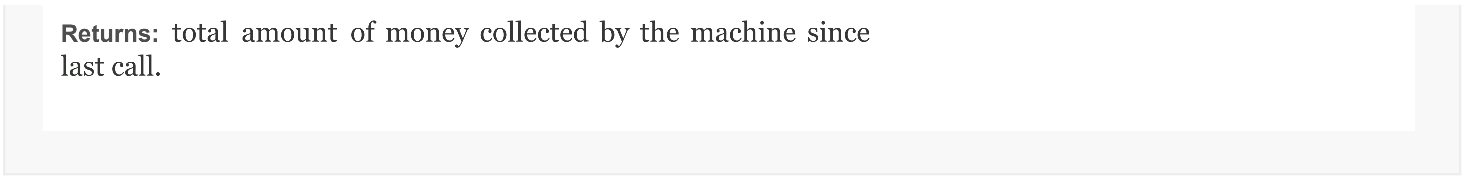
```
public int empty()
```

Description copied from interface: `PayStation`

Reset money collected. Sets the amount of money collected by the machine since the last call to 0.

Specified by:

`empty` in interface `PayStation`



Returns: total amount of money collected by the machine since last call.

Package edu.temple.cis.paystation

Interface Receipt

All Known Implementing Classes:

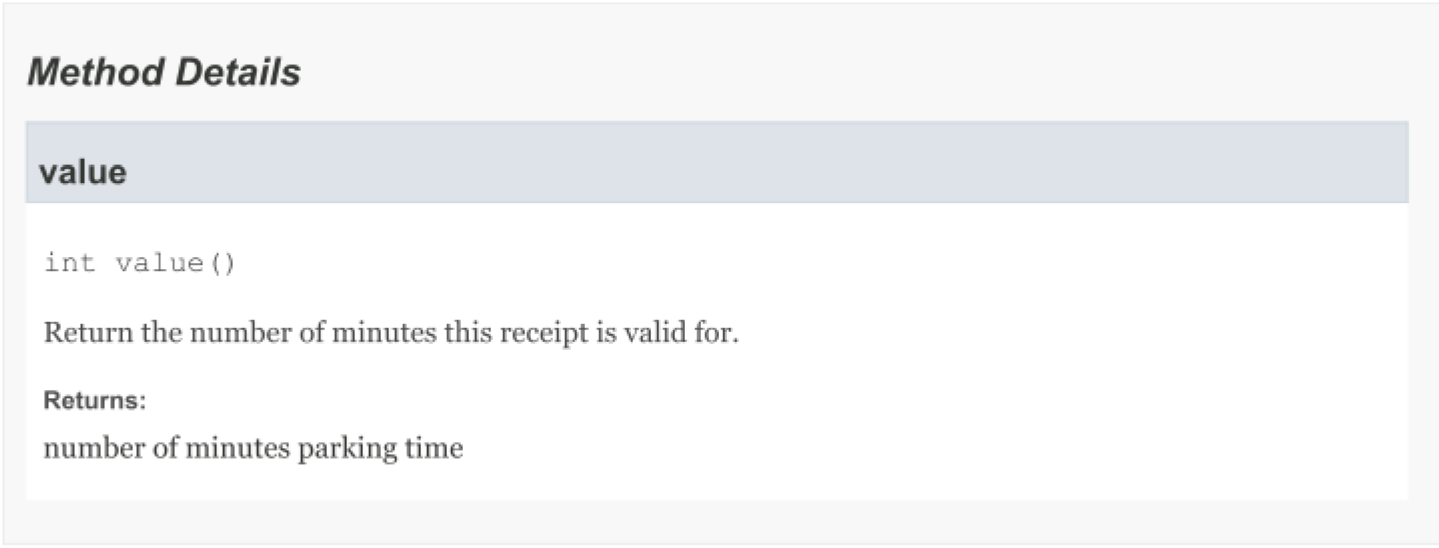
`ReceiptImpl`

`public interface Receipt`

Method Summary

All Methods | **Instance Methods** | **Abstract Methods**

Modifier and Type	Method	Description
`int`	`value()`	Return the number of minutes this receipt is valid for.



Method Details

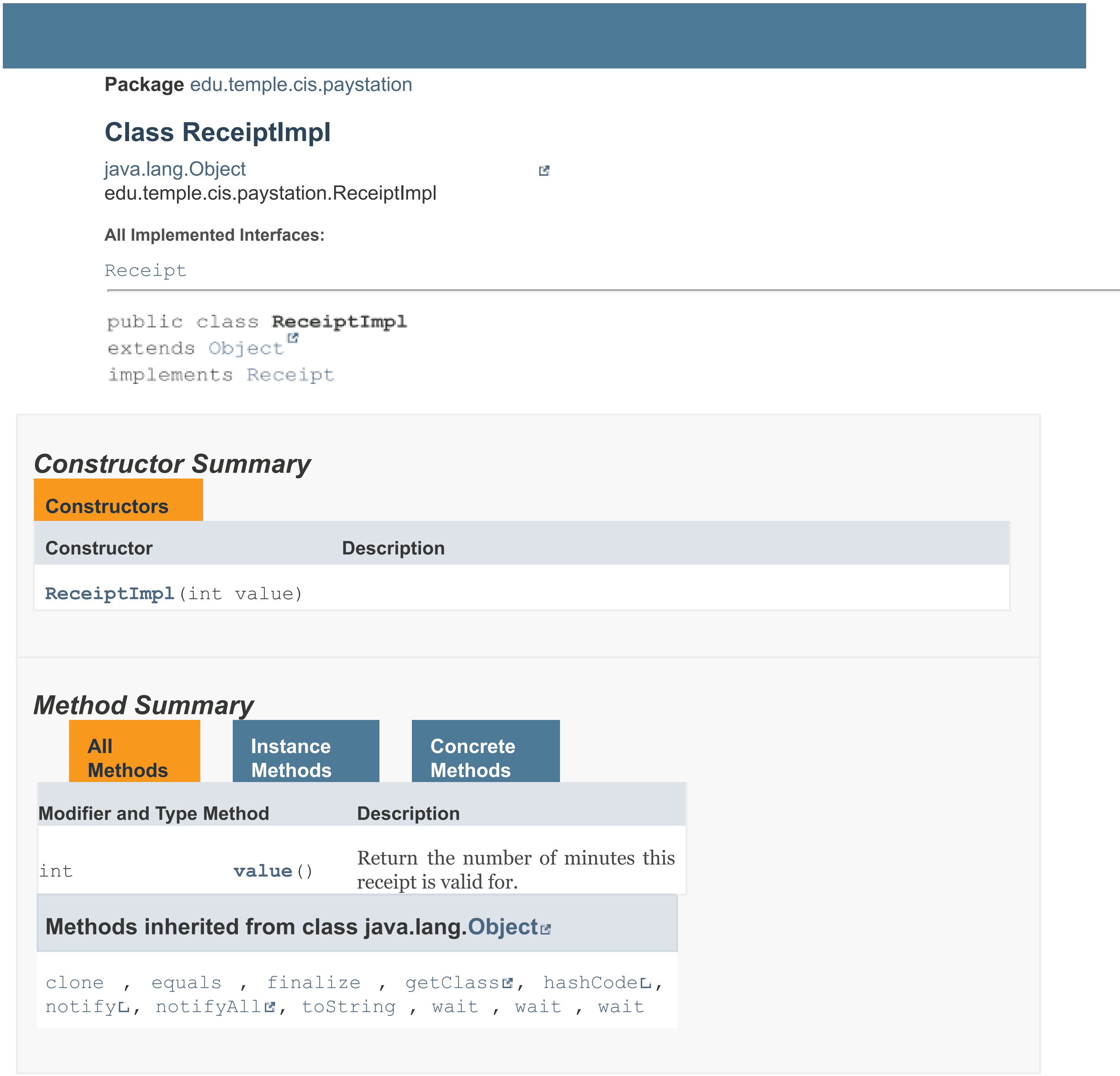
value

`int value()`

Return the number of minutes this receipt is valid for.

Returns:

number of minutes parking time



Package edu.temple.cis.paystation

Class ReceiptImpl

java.lang.Object
edu.temple.cis.paystation.ReceiptImpl

All Implemented Interfaces:

`Receipt`

```
public class ReceiptImpl
extends Object
implements Receipt
```

Constructor Summary

Constructors

Constructor	Description
`ReceiptImpl(int value)`	

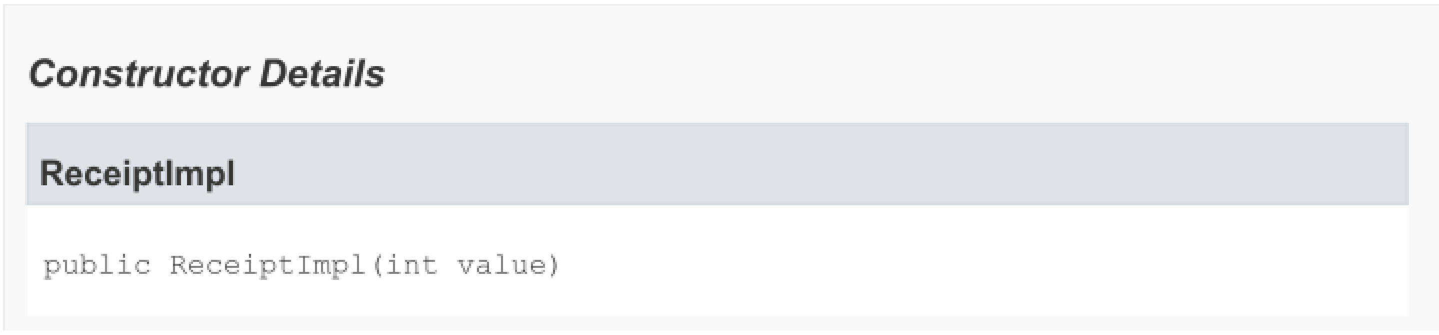
Method Summary

All Methods | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method	Description
`int`	`value()`	Return the number of minutes this receipt is valid for.

Methods inherited from class java.lang.Object

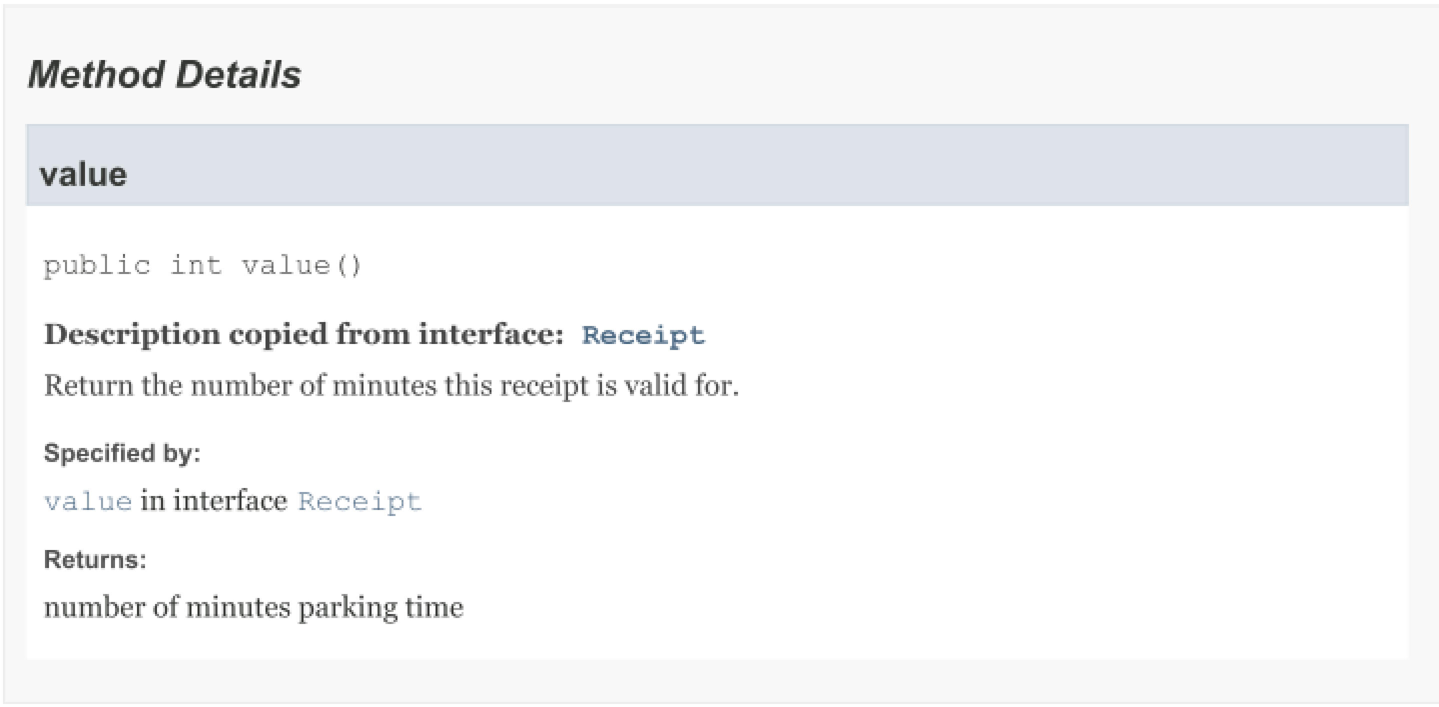
`clone , equals , finalize , getClass, hashCode, notify, notifyAll, toString , wait , wait , wait`



Constructor Details

ReceiptImpl

```
public ReceiptImpl(int value)
```



Method Details

value

```
public int value()
```

Description copied from interface: `Receipt`

Return the number of minutes this receipt is valid for.

Specified by:

`value` in interface `Receipt`

Returns:

number of minutes parking time