Package

exceptions

exceptions
Class AddressInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.AddressInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **AddressInvalidException**
extends java.lang.Exception

Constructor Summary

public	`AddressInvalidException()` Initializes Exception
public	`AddressInvalidException(java.lang.String message)` Gets the specific message for the exception
public	`AddressInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
public	`AddressInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

AddressInvalidException

```
public AddressInvalidException()
```

Initializes Exception

AddressInvalidException

```
public AddressInvalidException(java.lang.String message)
```

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

AddressInvalidException

```
public AddressInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

AddressInvalidException

```
public AddressInvalidException(java.lang.String message,
                               java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class BookingDepositInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.BookingDepositInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **BookingDepositInvalidException**
extends java.lang.Exception

Constructor Summary

`public`	`BookingDepositInvalidException()` Initializes Exception
`public`	`BookingDepositInvalidException(java.lang.String message)` Gets the specific message for the exception
`public`	`BookingDepositInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
`public`	`BookingDepositInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

BookingDepositInvalidException

`public BookingDepositInvalidException()`

Initializes Exception

BookingDepositInvalidException

`public BookingDepositInvalidException(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

BookingDepositInvalidException

```
public BookingDepositInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

BookingDepositInvalidException

```
public BookingDepositInvalidException(java.lang.String message,
                                      java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class BooleanInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.BooleanInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **BooleanInvalidException**
extends java.lang.Exception

Constructor Summary

`public`	`BooleanInvalidException()` Initializes Exception
`public`	`BooleanInvalidException(java.lang.String message)` Gets the specific message for the exception
`public`	`BooleanInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
`public`	`BooleanInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

BooleanInvalidException

`public **BooleanInvalidException**()`

Initializes Exception

BooleanInvalidException

`public **BooleanInvalidException**(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

BooleanInvalidException

```
public BooleanInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

BooleanInvalidException

```
public BooleanInvalidException(java.lang.String message,
                               java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class CellPhoneInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.CellPhoneInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **CellPhoneInvalidException**
extends java.lang.Exception

Constructor Summary

public	`CellPhoneInvalidException()` Initializes Exception
public	`CellPhoneInvalidException(java.lang.String message)` Gets the specific message for the exception
public	`CellPhoneInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
public	`CellPhoneInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

CellPhoneInvalidException

`public CellPhoneInvalidException()`

Initializes Exception

CellPhoneInvalidException

`public CellPhoneInvalidException(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

CellPhoneInvalidException

```
public CellPhoneInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

CellPhoneInvalidException

```
public CellPhoneInvalidException(java.lang.String message,
                                 java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class CityInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.CityInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **CityInvalidException**
extends java.lang.Exception

Constructor Summary

`public`	`CityInvalidException()` Initializes Exception
`public`	`CityInvalidException(java.lang.String message)` Gets the specific message for the exception
`public`	`CityInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
`public`	`CityInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`
`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

CityInvalidException

`public` **`CityInvalidException`**`()`

Initializes Exception

CityInvalidException

`public` **`CityInvalidException`**`(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

CityInvalidException

```
public CityInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

CityInvalidException

```
public CityInvalidException(java.lang.String message,
                            java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class ClientDoesNotExistException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.ClientDoesNotExistException
```

All Implemented Interfaces:
java.io.Serializable

public class **ClientDoesNotExistException**
extends java.lang.Exception

Constructor Summary

`public`	`ClientDoesNotExistException()` Initializes Exception
`public`	`ClientDoesNotExistException(java.lang.String message)` Gets the specific message for the exception
`public`	`ClientDoesNotExistException(java.lang.Throwable cause)` Gets the specific cause for the exception
`public`	`ClientDoesNotExistException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

ClientDoesNotExistException

public **ClientDoesNotExistException**()

Initializes Exception

ClientDoesNotExistException

public **ClientDoesNotExistException**(java.lang.String message)

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

ClientDoesNotExistException

```
public ClientDoesNotExistException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

ClientDoesNotExistException

```
public ClientDoesNotExistException(java.lang.String message,
                                   java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception

`cause` - is the cause for the reason an exception is thrown

exceptions
Class ClientNameNotUniqueException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.ClientNameNotUniqueException
```

All Implemented Interfaces:
java.io.Serializable

public class **ClientNameNotUniqueException**
extends java.lang.Exception

Constructor Summary

`public`	`ClientNameNotUniqueException()` Initializes Exception
`public`	`ClientNameNotUniqueException(java.lang.String message)` Gets the specific message for the exception
`public`	`ClientNameNotUniqueException(java.lang.Throwable cause)` Gets the specific cause for the exception
`public`	`ClientNameNotUniqueException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

ClientNameNotUniqueException

`public` **`ClientNameNotUniqueException`**`()`

Initializes Exception

ClientNameNotUniqueException

`public` **`ClientNameNotUniqueException`**`(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

ClientNameNotUniqueException

```
public ClientNameNotUniqueException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

ClientNameNotUniqueException

```
public ClientNameNotUniqueException(java.lang.String message,
                                    java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class CloseTimeInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.CloseTimeInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **CloseTimeInvalidException**
extends java.lang.Exception

Constructor Summary

public	CloseTimeInvalidException()
public	CloseTimeInvalidException(java.lang.String message)
public	CloseTimeInvalidException(java.lang.Throwable cause)
public	CloseTimeInvalidException(java.lang.String message, java.lang.Throwable cause)

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

CloseTimeInvalidException

```
public CloseTimeInvalidException()
```

CloseTimeInvalidException

```
public CloseTimeInvalidException(java.lang.String message)
```

CloseTimeInvalidException

```
public CloseTimeInvalidException(java.lang.Throwable cause)
```

CloseTimeInvalidException

```
public CloseTimeInvalidException(java.lang.String message,
                                 java.lang.Throwable cause)
```

exceptions
Class CostInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.CostInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **CostInvalidException**
extends java.lang.Exception

Constructor Summary

`public`	`CostInvalidException()` Initializes Exception
`public`	`CostInvalidException(java.lang.String message)` Gets the specific message for the exception
`public`	`CostInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
`public`	`CostInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

CostInvalidException

`public CostInvalidException()`

Initializes Exception

CostInvalidException

`public CostInvalidException(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

CostInvalidException

```
public CostInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

CostInvalidException

```
public CostInvalidException(java.lang.String message,
                            java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class CountryInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.CountryInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **CountryInvalidException**
extends java.lang.Exception

Constructor Summary

`public`	`CountryInvalidException()` Initializes Exception
`public`	`CountryInvalidException(java.lang.String message)` Gets the specific message for the exception
`public`	`CountryInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
`public`	`CountryInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

CountryInvalidException

`public CountryInvalidException()`

Initializes Exception

CountryInvalidException

`public CountryInvalidException(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

CountryInvalidException

```
public CountryInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

CountryInvalidException

```
public CountryInvalidException(java.lang.String message,
                               java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class DamageDepositInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.DamageDepositInvalidException
```

All Implemented Interfaces:
java.io.Serializable

---

public class **DamageDepositInvalidException**
extends java.lang.Exception

Constructor Summary

`public`	`DamageDepositInvalidException()` Initializes Exception
`public`	`DamageDepositInvalidException(java.lang.String message)` Gets the specific message for the exception
`public`	`DamageDepositInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
`public`	`DamageDepositInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

DamageDepositInvalidException

`public `**`DamageDepositInvalidException`**`()`

Initializes Exception

DamageDepositInvalidException

`public `**`DamageDepositInvalidException`**`(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

DamageDepositInvalidException

```
public DamageDepositInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

DamageDepositInvalidException

```
public DamageDepositInvalidException(java.lang.String message,
                                     java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class DatabaseConnectionException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.DatabaseConnectionException
```

All Implemented Interfaces:
java.io.Serializable

public class **DatabaseConnectionException**
extends java.lang.Exception

Constructor Summary

`public`	`DatabaseConnectionException()` Initializes Exception
`public`	`DatabaseConnectionException(java.lang.String message)` Gets the specific message for the exception
`public`	`DatabaseConnectionException(java.lang.Throwable cause)` Gets the specific cause for the exception
`public`	`DatabaseConnectionException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`
`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

DatabaseConnectionException

`public` **`DatabaseConnectionException`**`()`

Initializes Exception

DatabaseConnectionException

`public` **`DatabaseConnectionException`**`(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

DatabaseConnectionException

```
public DatabaseConnectionException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

DatabaseConnectionException

```
public DatabaseConnectionException(java.lang.String message,
                                   java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class DateInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.DateInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **DateInvalidException**
extends java.lang.Exception

Constructor Summary

`public`	`DateInvalidException()` Initializes Exception
`public`	`DateInvalidException(java.lang.String message)` Gets the specific message for the exception
`public`	`DateInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
`public`	`DateInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`
`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

DateInvalidException

`public DateInvalidException()`

Initializes Exception

DateInvalidException

`public DateInvalidException(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

DateInvalidException

```
public DateInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

DateInvalidException

```
public DateInvalidException(java.lang.String message,
                            java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class DescriptionInvalidException

```
java.lang.Object
   |
   +-java.lang.Throwable
      |
      +-java.lang.Exception
         |
         +-exceptions.DescriptionInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **DescriptionInvalidException**
extends java.lang.Exception

Constructor Summary

`public`	`DescriptionInvalidException()` Initializes Exception
`public`	`DescriptionInvalidException(java.lang.String message)` Gets the specific message for the exception
`public`	`DescriptionInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
`public`	`DescriptionInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

DescriptionInvalidException

```
public DescriptionInvalidException()
```

Initializes Exception

DescriptionInvalidException

```
public DescriptionInvalidException(java.lang.String message)
```

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

DescriptionInvalidException

```
public DescriptionInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

DescriptionInvalidException

```
public DescriptionInvalidException(java.lang.String message,
                                   java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class DiscountInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.DiscountInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **DiscountInvalidException**
extends java.lang.Exception

Constructor Summary

`public`	`DiscountInvalidException()` Initializes Exception
`public`	`DiscountInvalidException(java.lang.String message)` Gets the specific message for the exception
`public`	`DiscountInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
`public`	`DiscountInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

DiscountInvalidException

`public DiscountInvalidException()`

Initializes Exception

DiscountInvalidException

`public DiscountInvalidException(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

DiscountInvalidException

```
public DiscountInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

DiscountInvalidException

```
public DiscountInvalidException(java.lang.String message,
                                java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class EmailInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.EmailInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **EmailInvalidException**
extends java.lang.Exception

Constructor Summary

public	`EmailInvalidException()` Initializes Exception
public	`EmailInvalidException(java.lang.String message)` Gets the specific message for the exception
public	`EmailInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
public	`EmailInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

EmailInvalidException

public **EmailInvalidException**()

Initializes Exception

EmailInvalidException

public **EmailInvalidException**(java.lang.String message)

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

EmailInvalidException

```
public EmailInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

EmailInvalidException

```
public EmailInvalidException(java.lang.String message,
                             java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class GivenNameInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.GivenNameInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **GivenNameInvalidException**
extends java.lang.Exception

Constructor Summary

public	`GivenNameInvalidException()` Initializes Exception
public	`GivenNameInvalidException(java.lang.String message)` Gets the specific message for the exception
public	`GivenNameInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
public	`GivenNameInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

GivenNameInvalidException

`public GivenNameInvalidException()`

Initializes Exception

GivenNameInvalidException

`public GivenNameInvalidException(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

GivenNameInvalidException

```
public GivenNameInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

GivenNameInvalidException

```
public GivenNameInvalidException(java.lang.String message,
                                 java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class HomePhoneInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.HomePhoneInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **HomePhoneInvalidException**
extends java.lang.Exception

Constructor Summary

public	HomePhoneInvalidException() Initializes Exception
public	HomePhoneInvalidException(java.lang.String message) Gets the specific message for the exception
public	HomePhoneInvalidException(java.lang.Throwable cause) Gets the specific cause for the exception
public	HomePhoneInvalidException(java.lang.String message, java.lang.Throwable cause) Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

HomePhoneInvalidException

`public HomePhoneInvalidException()`

Initializes Exception

HomePhoneInvalidException

`public HomePhoneInvalidException(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

HomePhoneInvalidException

```
public HomePhoneInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

HomePhoneInvalidException

```
public HomePhoneInvalidException(java.lang.String message,
                                 java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class MaxCapacityInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.MaxCapacityInvalidException
```

All Implemented Interfaces:
java.io.Serializable

---

public class **MaxCapacityInvalidException**
extends java.lang.Exception

Constructor Summary

`public`	`MaxCapacityInvalidException()` Initializes Exception
`public`	`MaxCapacityInvalidException(java.lang.String message)` Gets the specific message for the exception
`public`	`MaxCapacityInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
`public`	`MaxCapacityInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

MaxCapacityInvalidException

`public` **`MaxCapacityInvalidException`**`()`

Initializes Exception

MaxCapacityInvalidException

`public` **`MaxCapacityInvalidException`**`(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

MaxCapacityInvalidException

```
public MaxCapacityInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

MaxCapacityInvalidException

```
public MaxCapacityInvalidException(java.lang.String message,
                                   java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class MinBookingIntervalInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.MinBookingIntervalInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **MinBookingIntervalInvalidException**
extends java.lang.Exception

Constructor Summary

`public`	`MinBookingIntervalInvalidException()` Initializes Exception
`public`	`MinBookingIntervalInvalidException(java.lang.String message)` Gets the specific message for the exception
`public`	`MinBookingIntervalInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
`public`	`MinBookingIntervalInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

MinBookingIntervalInvalidException

`public` **`MinBookingIntervalInvalidException`**`()`

Initializes Exception

MinBookingIntervalInvalidException

`public` **`MinBookingIntervalInvalidException`**`(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

MinBookingIntervalInvalidException

```
public MinBookingIntervalInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

MinBookingIntervalInvalidException

```
public MinBookingIntervalInvalidException(java.lang.String message,
                                          java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class MinBookingTimeInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.MinBookingTimeInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **MinBookingTimeInvalidException**
extends java.lang.Exception

Constructor Summary

`public`	`MinBookingTimeInvalidException()` Initializes Exception
`public`	`MinBookingTimeInvalidException(java.lang.String message)` Gets the specific message for the exception
`public`	`MinBookingTimeInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
`public`	`MinBookingTimeInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

MinBookingTimeInvalidException

public **MinBookingTimeInvalidException**()

Initializes Exception

MinBookingTimeInvalidException

public **MinBookingTimeInvalidException**(java.lang.String message)

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

MinBookingTimeInvalidException

```
public MinBookingTimeInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

MinBookingTimeInvalidException

```
public MinBookingTimeInvalidException(java.lang.String message,
                                      java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class NameInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.NameInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **NameInvalidException**
extends java.lang.Exception

Constructor Summary

`public`	`NameInvalidException()` Initializes Exception
`public`	`NameInvalidException(java.lang.String message)` Gets the specific message for the exception
`public`	`NameInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
`public`	`NameInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

NameInvalidException

`public` **`NameInvalidException`**`()`

Initializes Exception

NameInvalidException

`public` **`NameInvalidException`**`(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

NameInvalidException

```
public NameInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

NameInvalidException

```
public NameInvalidException(java.lang.String message,
                            java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class NumberInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.NumberInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **NumberInvalidException**
extends java.lang.Exception

Constructor Summary

public	`NumberInvalidException()` Initializes Exception
public	`NumberInvalidException(java.lang.String message)` Gets the specific message for the exception
public	`NumberInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
public	`NumberInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`
`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

NumberInvalidException

`public NumberInvalidException()`

Initializes Exception

NumberInvalidException

`public NumberInvalidException(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

NumberInvalidException

```
public NumberInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

NumberInvalidException

```
public NumberInvalidException(java.lang.String message,
                              java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class OpenTimeInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.OpenTimeInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **OpenTimeInvalidException**
extends java.lang.Exception

Constructor Summary

public	OpenTimeInvalidException()
public	OpenTimeInvalidException(java.lang.String message)
public	OpenTimeInvalidException(java.lang.Throwable cause)
public	OpenTimeInvalidException(java.lang.String message, java.lang.Throwable cause)

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

OpenTimeInvalidException

```
public OpenTimeInvalidException()
```

OpenTimeInvalidException

```
public OpenTimeInvalidException(java.lang.String message)
```

OpenTimeInvalidException

```
public OpenTimeInvalidException(java.lang.Throwable cause)
```

OpenTimeInvalidException

```
public OpenTimeInvalidException(java.lang.String message,
                                java.lang.Throwable cause)
```

exceptions
Class PasswordInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.PasswordInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **PasswordInvalidException**
extends java.lang.Exception

Constructor Summary

`public`	`PasswordInvalidException()` Initializes Exception
`public`	`PasswordInvalidException(java.lang.String message)` Gets the specific message for the exception
`public`	`PasswordInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
`public`	`PasswordInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

PasswordInvalidException

`public PasswordInvalidException()`

Initializes Exception

PasswordInvalidException

`public PasswordInvalidException(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

PasswordInvalidException

```
public PasswordInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

PasswordInvalidException

```
public PasswordInvalidException(java.lang.String message,
                                java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class PostalCodeInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.PostalCodeInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **PostalCodeInvalidException**
extends java.lang.Exception

Constructor Summary

public	`PostalCodeInvalidException()` Initializes Exception
public	`PostalCodeInvalidException(java.lang.String message)` Gets the specific message for the exception
public	`PostalCodeInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
public	`PostalCodeInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

PostalCodeInvalidException

```
public PostalCodeInvalidException()
```

Initializes Exception

PostalCodeInvalidException

```
public PostalCodeInvalidException(java.lang.String message)
```

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

PostalCodeInvalidException

```
public PostalCodeInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

PostalCodeInvalidException

```
public PostalCodeInvalidException(java.lang.String message,
                                  java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class ProvinceInvalidException

```
java.lang.Object
   |
   +-java.lang.Throwable
      |
      +-java.lang.Exception
         |
         +-exceptions.ProvinceInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **ProvinceInvalidException**
extends java.lang.Exception

Constructor Summary

`public`	`ProvinceInvalidException()` Initializes Exception
`public`	`ProvinceInvalidException(java.lang.String message)` Gets the specific message for the exception
`public`	`ProvinceInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
`public`	`ProvinceInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`
`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

ProvinceInvalidException

`public ProvinceInvalidException()`

Initializes Exception

ProvinceInvalidException

`public ProvinceInvalidException(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

ProvinceInvalidException

```
public ProvinceInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

ProvinceInvalidException

```
public ProvinceInvalidException(java.lang.String message,
                                java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class ScheduleInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.ScheduleInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **ScheduleInvalidException**
extends java.lang.Exception

Constructor Summary

`public`	`ScheduleInvalidException()` Initializes Exception
`public`	`ScheduleInvalidException(java.lang.String message)` Gets the specific message for the exception
`public`	`ScheduleInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
`public`	`ScheduleInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

ScheduleInvalidException

`public` **`ScheduleInvalidException`**`()`

Initializes Exception

ScheduleInvalidException

`public` **`ScheduleInvalidException`**`(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

ScheduleInvalidException

```
public ScheduleInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

ScheduleInvalidException

```
public ScheduleInvalidException(java.lang.String message,
                                java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class SetupTimeInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.SetupTimeInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **SetupTimeInvalidException**
extends java.lang.Exception

Constructor Summary

public	`SetupTimeInvalidException()` Initializes Exception
public	`SetupTimeInvalidException(java.lang.String message)` Gets the specific message for the exception
public	`SetupTimeInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
public	`SetupTimeInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

SetupTimeInvalidException

`public SetupTimeInvalidException()`

Initializes Exception

SetupTimeInvalidException

`public SetupTimeInvalidException(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

SetupTimeInvalidException

```
public SetupTimeInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

SetupTimeInvalidException

```
public SetupTimeInvalidException(java.lang.String message,
                                 java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class SurnameInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.SurnameInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **SurnameInvalidException**
extends java.lang.Exception

Constructor Summary

`public`	`SurnameInvalidException()` Initializes Exception
`public`	`SurnameInvalidException(java.lang.String message)` Gets the specific message for the exception
`public`	`SurnameInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
`public`	`SurnameInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

SurnameInvalidException

`public` **`SurnameInvalidException`**`()`

Initializes Exception

SurnameInvalidException

`public` **`SurnameInvalidException`**`(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

SurnameInvalidException

```
public SurnameInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

SurnameInvalidException

```
public SurnameInvalidException(java.lang.String message,
                               java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class TearDownTimeInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.TearDownTimeInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **TearDownTimeInvalidException**
extends java.lang.Exception

Constructor Summary

public	`TearDownTimeInvalidException()` Initializes Exception
public	`TearDownTimeInvalidException(java.lang.String message)` Gets the specific message for the exception
public	`TearDownTimeInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
public	`TearDownTimeInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`
`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

TearDownTimeInvalidException

`public TearDownTimeInvalidException()`

Initializes Exception

TearDownTimeInvalidException

`public TearDownTimeInvalidException(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:

`message` - is the message that is thrown for the exception

TearDownTimeInvalidException

```
public TearDownTimeInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:

`cause` - is the cause for the reason an exception is thrown

TearDownTimeInvalidException

```
public TearDownTimeInvalidException(java.lang.String message,
                                    java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:

`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class UsernameInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.UsernameInvalidException
```

All Implemented Interfaces:

java.io.Serializable

public class **UsernameInvalidException**
extends java.lang.Exception

Constructor Summary

`public`	`UsernameInvalidException()` Initializes Exception
`public`	`UsernameInvalidException(java.lang.String message)` Gets the specific message for the exception
`public`	`UsernameInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
`public`	`UsernameInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

UsernameInvalidException

`public` **`UsernameInvalidException`**`()`

Initializes Exception

UsernameInvalidException

`public` **`UsernameInvalidException`**`(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:
`message` - is the message that is thrown for the exception

UsernameInvalidException

```
public UsernameInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:
`cause` - is the cause for the reason an exception is thrown

UsernameInvalidException

```
public UsernameInvalidException(java.lang.String message,
                                java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:
`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

exceptions
Class WorkPhoneInvalidException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-exceptions.WorkPhoneInvalidException
```

All Implemented Interfaces:
java.io.Serializable

public class **WorkPhoneInvalidException**
extends java.lang.Exception

Constructor Summary

`public`	`WorkPhoneInvalidException()` Initializes Exception
`public`	`WorkPhoneInvalidException(java.lang.String message)` Gets the specific message for the exception
`public`	`WorkPhoneInvalidException(java.lang.Throwable cause)` Gets the specific cause for the exception
`public`	`WorkPhoneInvalidException(java.lang.String message, java.lang.Throwable cause)` Gets the specific message and cause for the exception that is thrown

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

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

Constructors

WorkPhoneInvalidException

`public` **`WorkPhoneInvalidException`**`()`

Initializes Exception

WorkPhoneInvalidException

`public` **`WorkPhoneInvalidException`**`(java.lang.String message)`

(continued on next page)

(continued from last page)

Gets the specific message for the exception

Parameters:
`message` - is the message that is thrown for the exception

WorkPhoneInvalidException

```
public WorkPhoneInvalidException(java.lang.Throwable cause)
```

Gets the specific cause for the exception

Parameters:
`cause` - is the cause for the reason an exception is thrown

WorkPhoneInvalidException

```
public WorkPhoneInvalidException(java.lang.String message,
                                 java.lang.Throwable cause)
```

Gets the specific message and cause for the exception that is thrown

Parameters:
`message` - is the message that is thrown for the exception
`cause` - is the cause for the reason an exception is thrown

Package
logic

logic
Class AdditionalChargeManager

```
java.lang.Object
  |
  +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
          |
          +-logic.AdditionalChargeManager
```

All Implemented Interfaces:
java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable

public class **AdditionalChargeManager**
extends javax.servlet.http.HttpServlet

AdditionalChargeManager provides an interface for the GUI to interact with the AdditionalChargeBroker. It is also an implementation of a HttpServlet that will process any get and post requests from the GUI.

Constructor Summary

public	`AdditionalChargeManager()` Constructor for the AdditionalChargeManager.

Method Summary

void	`close()` Closes the current AdditionalChargeBroker instance
AdditionalCharge	`getAdditionalChargeInfo(int additional_charge_id)` Performs a query through the AdditionalChargeBroker to determine all related AdditionalCharge information related to the passed additional_charge_id.
java.util.List	`getAdditionalChargeList()` Retrieves all AdditionalCharges that exist in the database through the AdditionalChargeBroker.
boolean	`remove(AdditionalCharge ac)` Removes the AdditionalCharge passed from the database through the AdditionalChargeBroker.
boolean	`save(int id, java.lang.String acName, java.lang.String acCost)` Validates, then persists the AdditionalCharge information passed to the database through the AdditionalChargeBroker.
java.util.List	`search(java.lang.String query)` Searches the database for any AdditionalCharges that match the query provided.
boolean	`validate(AdditionalCharge ac)` Validates all information contained in an AdditionalCharge to ensure it properly conforms with all business requirements and database storage limitations.
boolean	`validate(java.lang.String acName, java.lang.String cost)` Validates all information contained passed to the method to ensure it properly conforms with all business requirements and database storage limitations.

Methods inherited from class `javax.servlet.http.HttpServlet`
`service`

Methods inherited from class `javax.servlet.GenericServlet`
`destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service`

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `javax.servlet.Servlet`
`destroy, getServletConfig, getServletInfo, init, service`

Methods inherited from interface `javax.servlet.ServletConfig`
`getInitParameter, getInitParameterNames, getServletContext, getServletName`

Constructors

AdditionalChargeManager

```
public AdditionalChargeManager()
```

Constructor for the AdditionalChargeManager. Gets the current instance of the AdditionalChargeBroker class for use in any database interactions.

Methods

validate

```
public boolean validate(java.lang.String acName,
          java.lang.String cost)
   throws CostInvalidException,
          NameInvalidException
```

Validates all information contained passed to the method to ensure it properly conforms with all business requirements and database storage limitations.

Parameters:

`acName` - - the name of the AdditionalCharge to validate

`cost` - - the cost of the AddtionalCharge to validate

Returns:

true if validation is successful. An appropriate exception is thrown otherwise

Throws:

`CostInvalidException` - - when the provided AddtionalCharges cost cannot be parsed to a double, is less than 0.0, or is not specified.

`NameInvalidException` - - when the provided AddtionalCharges name is longer than 25 characters, or is not provided

(continued on next page)

(continued from last page)

validate

```
public boolean validate(AdditionalCharge ac)
  throws CostInvalidException,
         NameInvalidException
```

Validates all information contained in an AdditionalCharge to ensure it properly conforms with all business requirements and database storage limitations.

Parameters:

`ac` - - the AdditionalCharge containing the information to validate.

Returns:

true if validation is successful. An appropriate exception is thrown otherwise

Throws:

`CostInvalidException` - - when the provided AddtionalCharges cost cannot be parsed to a double, is less than 0.0, or is not specified.

`NameInvalidException` - - when the provided AddtionalCharges name is longer than 25 characters, or is not provided

search

```
public java.util.List search(java.lang.String query)
  throws DatabaseConnectionException
```

Searches the database for any AdditionalCharges that match the query provided. The search is executed through the AdditionalChargeBroker

Parameters:

`query` - the string to search all AdditionalCharges for.

Returns:

a List containing all AdditionalCharges that match the search query provided.

Throws:

`DatabaseConnectionException` - if a connection to the database cannot be established

save

```
public boolean save(int id,
         java.lang.String acName,
         java.lang.String acCost)
  throws CostInvalidException,
         NameInvalidException,
         DatabaseConnectionException
```

Validates, then persists the AdditionalCharge information passed to the database through the AdditionalChargeBroker.

Parameters:

`id` - - the id of the AdditionalCharge. If creating a new AdditonalCharge this should be 0, otherwise it is set to the id of the AdditionalCharge to update.

`acName` - - the name of the AdditionalCharge

`acCost` - - the cost of the AdditionalCharge

Returns:

true if the AddtionCharge is successfully saved to the database. Throws an appropriate exception otherwise.

Throws:

`CostInvalidException` - - when the provided acCost cannot be parsed to a double, is less than 0.0, or is not specified.

`NameInvalidException` - - when the provided acName is longer than 25 characters, or is not provided

(continued on next page)

(continued from last page)

DatabaseConnectionException - - if a connection to the database cannot be established

remove

```
public boolean remove(AdditionalCharge ac)
  throws DatabaseConnectionException
```

Removes the AdditionalCharge passed from the database through the AdditionalChargeBroker.

Parameters:

ac - - the AdditionalCharge that is to be removed from the database. Only the id is needed to be set in order to perform this operation.

Returns:

true if the AdditionalCharge is successfully removed from the database. If the AdditionalCharge cannot be removed from the database an exception will be thrown.

Throws:

DatabaseConnectionException - - if a connection to the database cannot be established

close

```
public void close()
```

Closes the current AdditionalChargeBroker instance

getAdditionalChargeList

```
public java.util.List getAdditionalChargeList()
  throws DatabaseConnectionException
```

Retrieves all AdditionalCharges that exist in the database through the AdditionalChargeBroker.

Returns:

a list of all AdditionalCharges that exist

Throws:

DatabaseConnectionException - if a connection to the database cannot be established

getAdditionalChargeInfo

```
public AdditionalCharge getAdditionalChargeInfo(int additional_charge_id)
  throws DatabaseConnectionException
```

Performs a query through the AdditionalChargeBroker to determine all related AdditionalCharge information related to the passed additional_charge_id. If the passed additional_charge_id does not exist in the database, a null value will be returned.

Parameters:

additional_charge_id - - The id of the AdditionalCharge to look up

Returns:

an AdditionalCharge containing any information related to the additional_charge_id passed
null if no AdditionalCharge exists with the passed additional_charge_id

Throws:

DatabaseConnectionException - - if a connection to the database cannot be established

logic
Class BookingManager

```
java.lang.Object
  |
  +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
          |
          +-logic.BookingManager
```

All Implemented Interfaces:

java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable

public class **BookingManager**
extends javax.servlet.http.HttpServlet

Servlet implementation class BookingManager

Constructor Summary

`public`	`BookingManager()` Constructor for the BookingManager.

Method Summary

`void`	`close()` Closes the current BookingBroker instance
`java.lang.String`	`generateSchedule(Facility facility, java.util.Calendar start, java.util.Calendar end, java.util.List bookings)` Generates a series of HTML tables to represent a number of Bookings on a schedule.
`java.util.List`	`getAllBookingsForClient(int client_id)` Gets a List of all Bookings in the database that have been made for the Client specified by the passed client_id
`java.util.List`	`getAllBookingsForInvoice(int invoice_no)` Gets a List of all Bookings in the database that are included on an Invoice with the specified invoice_no
`Booking`	`getBookingInformation(int id)` Performs a query through the BookingBroker to determine all related Booking information related to the passed id.
`boolean`	`remove(Booking booking)` Removed the passed Booking from the database through the BookingBroker
`boolean`	`save(Booking booking)` Persists a booking into the database

boolean	`save(int id, java.lang.String eventTitle, BookingType eventType, java.util.Date startTime, java.util.Date endTime, java.lang.String setupTime, java.lang.String tearDownTime, java.lang.String client, Catering catering, Employee creator, java.lang.String numberOfPeople, Rate rate, java.util.ArrayList additionalCharges, Facility facility, java.lang.String invoice_no)` Validates, then persists the Booking information passed to the database through the Booking.
java.util.List	`search(java.util.Date start, java.util.Date end)` Finds all Bookings during the specified time period
java.util.List	`search(java.lang.String search, java.util.Date start, java.util.Date end, Facility facility)` Finds all bookings for a specified faciltiy during a specifed timeframe that matches the specified search string
java.util.List	`search(java.lang.String search, Facility facility, java.util.Date startTime, java.util.Date endTime)` Sends a message to the database to search for requested information, and returns information in a list format.
boolean	`validate(Booking booking)` Validates all fields of an booking object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field.
boolean	`validate(java.lang.String eventTitle, BookingType eventType, java.util.Date startTime, java.util.Date endTime, java.lang.String setupTime, java.lang.String tearDownTime, java.lang.String client, Catering catering, Employee creator, java.lang.String numberOfPeople, Rate rate, java.util.ArrayList additionalCharges, Facility facility, java.lang.String invoice_no)` Validates all fields of an booking object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field.

Methods inherited from class `javax.servlet.http.HttpServlet`
`service`

Methods inherited from class `javax.servlet.GenericServlet`
`destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service`

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `javax.servlet.Servlet`
`destroy, getServletConfig, getServletInfo, init, service`

Methods inherited from interface `javax.servlet.ServletConfig`
`getInitParameter, getInitParameterNames, getServletContext, getServletName`

Constructors

(continued from last page)

BookingManager

```
public BookingManager()
```

Constructor for the BookingManager. Gets the current instance of the BookingBroker class for use in any database interactions.

Methods

generateSchedule

```
public java.lang.String generateSchedule(Facility facility,
         java.util.Calendar start,
         java.util.Calendar end,
         java.util.List bookings)
  throws DatabaseConnectionException
```

Generates a series of HTML tables to represent a number of Bookings on a schedule.

Parameters:

`facility` - The facility to generate a schedule for.
`start` - the date to Start generating a schedule for
`end` - the date to finish generating the schedule on
`bookings` - the bookings to include on the schedule

Returns:

a String containing HTML to display the schedule

Throws:

`DatabaseConnectionException` - if a connection to the database cannot be established

validate

```
public boolean validate(Booking booking)
  throws NameInvalidException,
         ScheduleInvalidException,
         SetupTimeInvalidException,
         TearDownTimeInvalidException,
         NumberInvalidException,
         DateInvalidException,
         DatabaseConnectionException,
         ClientNameNotUniqueException,
         ClientDoesNotExistException,
         java.sql.SQLException
```

Validates all fields of an booking object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field. If a field is not valid a appropriate exception will be thrown

Parameters:

`booking` - contains all information for booking

Returns:

true/false if the object is valid.

Throws:

`NameInvalidException` - is thrown if it is null or <30 characters
`ScheduleInvalidException` - is thrown if a booking already exists for that specific time frame in the schedule
`SetupTimeInvalidException` - is thrown if it is <0 minutes
`TearDownTimeInvalidException` - is thrown if it is <0 minutes
`NumberInvalidException` - is thrown if setupTime or tearDownTime are not in a numerical format

(continued from last page)

`DateInvalidException` - is thrown if it is null,end time before start time or end time == start time.
`ClientDoesNotExistException`
`ClientNameNotUniqueException`
`DatabaseConnectionException`
`SQLException`

validate

```
public boolean validate(java.lang.String eventTitle,
         BookingType eventType,
         java.util.Date startTime,
         java.util.Date endTime,
         java.lang.String setupTime,
         java.lang.String tearDownTime,
         java.lang.String client,
         Catering catering,
         Employee creator,
         java.lang.String numberOfPeople,
         Rate rate,
         java.util.ArrayList additionalCharges,
         Facility facility,
         java.lang.String invoice_no)
  throws NameInvalidException,
         ScheduleInvalidException,
         SetupTimeInvalidException,
         TearDownTimeInvalidException,
         NumberInvalidException,
         DateInvalidException,
         DatabaseConnectionException,
         ClientNameNotUniqueException,
         ClientDoesNotExistException,
         java.sql.SQLException
```

Validates all fields of an booking object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field. If a field is not valid a appropriate exception will be thrown

Parameters:

`eventTitle` - for the booking
`eventType` - for the booking
`startTime` - for the booking
`endTime` - for the booking
`setupTime` - for the booking
`tearDownTime` - for the booking
`client` - for the booking
`creator` - for the booking
`rate` - for the booking
`additionalCharges` - for the booking
`facility` - for the booking
`invoice_no` - for the booking

Returns:

true/false if the object is valid.

Throws:

`NameInvalidException` - is thrown if it is null or <30 characters
`ScheduleInvalidException` - is thrown if a booking already exists for that specific time frame in the schedule
`SetupTimeInvalidException` - is thrown if it is <0 minutes
`TearDownTimeInvalidException` - is thrown if it is <0 minutes
`NumberInvalidException` - is thrown if setupTime or tearDownTime are not in a numerical format
`DateInvalidException` - is thrown if it is null,end time before start time or end time == start time.
`DatabaseConnectionException`
`ClientNameNotUniqueException`
`ClientDoesNotExistException`

(continued from last page)

`SQLException`

search

```
public java.util.List search(java.lang.String search,
         Facility facility,
         java.util.Date startTime,
         java.util.Date endTime)
  throws DatabaseConnectionException
```

Sends a message to the database to search for requested information, and returns information in a list format.

Parameters:

`search` - passes message along to database to search than returns a list

Returns:

List with all requested information from database

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

search

```
public java.util.List search(java.lang.String search,
         java.util.Date start,
         java.util.Date end,
         Facility facility)
  throws DatabaseConnectionException
```

Finds all bookings for a specified faciltiy during a specifed timeframe that matches the specified search string

Parameters:

`search` - String to search booking for
`start` - The date to begin including Bookings, non-Inclusive
`end` - The date to end including bookings, non-Inclusive

Returns:

a List containing all Bookings between the start and end dates provided for the specified facility, non-Inclusive

Throws:

`DatabaseConnectionException` - if a connection to the database cannot be established

search

```
public java.util.List search(java.util.Date start,
         java.util.Date end)
```

Finds all Bookings during the specified time period

Parameters:

`start` - The date to begin including Bookings, non-Inclusive
`end` - The date to end including bookings, non-Inclusive

Returns:

a List containing all Bookings between the start and end dates provided, non-Inclusive

save

```
public boolean save(Booking booking)
  throws DatabaseConnectionException
```

(continued from last page)

Persists a booking into the database

Parameters:

`booking` - the Booking to save in the database

Returns:

true if the Booking is sucsessfully persisted into the database

Throws:

`DatabaseConnectionException` - if a connection to the database cannot be established

save

```
public boolean save(int id,
         java.lang.String eventTitle,
         BookingType eventType,
         java.util.Date startTime,
         java.util.Date endTime,
         java.lang.String setupTime,
         java.lang.String tearDownTime,
         java.lang.String client,
         Catering catering,
         Employee creator,
         java.lang.String numberOfPeople,
         Rate rate,
         java.util.ArrayList additionalCharges,
         Facility facility,
         java.lang.String invoice_no)
  throws NameInvalidException,
         ScheduleInvalidException,
         SetupTimeInvalidException,
         TearDownTimeInvalidException,
         NumberInvalidException,
         DatabaseConnectionException,
         ClientNameNotUniqueException,
         ClientDoesNotExistException,
         java.sql.SQLException,
         DateInvalidException
```

Validates, then persists the Booking information passed to the database through the Booking.

Parameters:

`id` - the id of the Booking to persist. If creating a new Booking this should be 0, otherwise it is set to the id of the Booking to update.

`eventTitle` - the title of the booking

`eventType` - the BookingType of the Booking.

`startTime` - the time the Booking starts at.

`endTime` - the time the Booking ends at.

`setupTime` - the amount of time in minutes that is needed for setup.

`tearDownTime` - the amoun of time in minutes that is needed for teardown.

`client` - the Client name that the booking is made for.

`catering` - the Catering object related to this Booking

`creator` - the Employee that created this Booking

`numberOfPeople` - the number of people that are attending this Booking

`rate` - the Rate that this event is being charged at

`additionalCharges` - A List of any AdditionalCharges that are charged for this Booking

`facility` - the Facility this booking is booked for

`invoice_no` - the number of the Invoice this booking is included on

Returns:

true if Booking infomation is valid and saved to the database. Throws an appropriate exception otherwise.

Throws:

(continued from last page)

`NameInvalidException` - if eventTitle is null or longer than 25 characters
`ScheduleInvalidException` - if a Booking is previously booked in the same timeframe
`SetupTimeInvalidException` - if setupTime is not a number or is less than 0
`TearDownTimeInvalidException` - if tearDownTime is not a number or is less than 0
`NumberInvalidException` - if numberOfPeople is not a number or is less than 0
`DatabaseConnectionException` - if a connection to the database cannot be established
`DateInvalidException` - if the start or end date provided is not a date
`ClientNameNotUniqueException` - if the Client name provided is not unique
`ClientDoesNotExistException` - if the Client name provided does not exist in the database
`SQLException` - if there is an sql error

remove

```
public boolean remove(Booking booking)
  throws DatabaseConnectionException
```

Removed the passed Booking from the database through the BookingBroker

Parameters:

`booking` - the Booking that is to be removed from the database. Only the id is needed to be set in order to perform this operation.

Returns:

true if the Booking is succsesfully removed from the database. If the Booking cannot be removed from the database an exception will be thrown.

Throws:

`DatabaseConnectionException` - if a connection to the database cannot be established

getAllBookingsForClient

```
public java.util.List getAllBookingsForClient(int client_id)
  throws DatabaseConnectionException
```

Gets a List of all Bookings in the database that have been made for the Client specified by the passed client_id

Parameters:

`client_id` - the id of the Client that related bookings will be found for

Returns:

a List containing all Booking that are associated with the Client specifed by the passed client id

Throws:

`DatabaseConnectionException` - if a connection to the database cannot be established

getAllBookingsForInvoice

```
public java.util.List getAllBookingsForInvoice(int invoice_no)
  throws DatabaseConnectionException
```

Gets a List of all Bookings in the database that are included on an Invoice with the specified invoice_no

Parameters:

`invoice_no` - the invoice number that related bookings will be found for

Returns:

a List containing all Bookings that are associated with the passed invoice_no

Throws:

`DatabaseConnectionException` - if a connection to the database cannot be established

getBookingInformation

```
public Booking getBookingInformation(int id)
  throws DatabaseConnectionException
```

Performs a query through the BookingBroker to determine all related Booking information related to the passed id. If the passed id does not exist in the database, a null value will be returned.

Parameters:

`id` - the id of the Booking to look up

Returns:

a Booking containing any information related to the id passed
null if no Booking exists with the passed id

Throws:

`DatabaseConnectionException` - if a connection to the database cannot be established

close

```
public void close()
```

Closes the current BookingBroker instance

logic
Class BookingTypeManager

```
java.lang.Object
  |
  +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
          |
          +-logic.BookingTypeManager
```

All Implemented Interfaces:
java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable

public class **BookingTypeManager**
extends javax.servlet.http.HttpServlet

Servlet implementation class BookingTypeManager

Constructor Summary

public	`BookingTypeManager()` Gets the broker instance

Method Summary

void	`close()` Closes database connection
BookingType	`getBookingTypeInfo(int bookingType_id)`
java.util.List	`getBookingTypeList()` Searches Database for required information and returns it in a list format
boolean	`remove(BookingType bt)` Removes the supplied object from the database
boolean	`save(BookingType bt)` Validates all fields for bt to ensure that all fields are valid.
boolean	`save(int id, java.lang.String name, java.lang.String setupTime, java.lang.String tearDownTime)`
java.util.List	`search(java.lang.String query)` Sends a message to the database to search for requested information based on query and returns information in a list format.
boolean	`validate(BookingType bt)` Validates all fields of an bookingType object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field.
boolean	`validate(java.lang.String name, java.lang.String setupTime, java.lang.String tearDownTime)` Validates all fields of an bookingType object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field.

Methods inherited from class `javax.servlet.http.HttpServlet`
`service`

Methods inherited from class `javax.servlet.GenericServlet`
`destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service`

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `javax.servlet.Servlet`
`destroy, getServletConfig, getServletInfo, init, service`

Methods inherited from interface `javax.servlet.ServletConfig`
`getInitParameter, getInitParameterNames, getServletContext, getServletName`

Constructors

BookingTypeManager

```
public BookingTypeManager()
```

Gets the broker instance

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

See Also:

`javax.servlet.http.HttpServlet()`

Methods

validate

```
public boolean validate(BookingType bt)
  throws NameInvalidException,
         SetupTimeInvalidException,
         TearDownTimeInvalidException
```

Validates all fields of an bookingType object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field. If a field is not valid a appropriate exception will be thrown

Parameters:

`bt` - BookingType that is validated

Returns:

true/false if the object is valid.

Throws:

`NameInvalidException` - is thrown if it is null or >45 characters

`SetupTimeInvalidException` - is thrown if it is <0 minutes

(continued from last page)

TearDownTimeInvalidException - is thrown if it is <0 minutes

validate

```
public boolean validate(java.lang.String name,
         java.lang.String setupTime,
         java.lang.String tearDownTime)
  throws TearDownTimeInvalidException,
         SetupTimeInvalidException,
         NameInvalidException
```

Validates all fields of an bookingType object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field. If a field is not valid a appropriate exception will be thrown

Parameters:

`name` - for the bookingType

`setupTime` - for the bookingType

`tearDownTime` - for the bookingType

Returns:

true/false if the object is valid.

Throws:

TearDownTimeInvalidException - is thrown if it is <0 minutes

SetupTimeInvalidException - is thrown if it is <0 minutes

NameInvalidException - is thrown if it is null or >45 characters

search

```
public java.util.List search(java.lang.String query)
  throws DatabaseConnectionException
```

Sends a message to the database to search for requested information based on query and returns information in a list format.

Parameters:

`query` - is the query used to select proper information from requested table

Returns:

list of all requested information from database

Throws:

DatabaseConnectionException - is thrown if DB connection fails

save

```
public boolean save(BookingType bt)
  throws NameInvalidException,
         SetupTimeInvalidException,
         TearDownTimeInvalidException,
         DatabaseConnectionException
```

Validates all fields for bt to ensure that all fields are valid. If a field is not valid a appropriate exception will be thrown If name exists, information in bookingType table is updated, otherwise information is inserted into table

Parameters:

`bt` - bookingType contains all information for a bookingType

Returns:

boolean

(continued from last page)

Throws:

`TearDownTimeInvalidException` - is thrown if it is <0 minutes
`SetupTimeInvalidException` - is thrown if it is <0 minutes
`NameInvalidException` - is thrown if it is null or >45 characters
`DatabaseConnectionException` - is thrown if DB connection fails

save

```
public boolean save(int id,
         java.lang.String name,
         java.lang.String setupTime,
         java.lang.String tearDownTime)
  throws java.lang.NumberFormatException,
         TearDownTimeInvalidException,
         SetupTimeInvalidException,
         NameInvalidException,
         DatabaseConnectionException
```

Parameters:

`name` - for bookingType
`setupTime` - for bookingType
`tearDownTime` - for bookingType

Returns:

updated list for bookingType

Throws:

`NameInvalidException` - is thrown if it is null or >45 characters
`SetupTimeInvalidException` - is thrown if it is <0 minutes
`TearDownTimeInvalidException` - is thrown if it is <0 minutes
`NumberFormatException` - is thrown if setupTime or tearDownTime are in a non-numeric format
`DatabaseConnectionException` - is thrown if DB connection fails

remove

```
public boolean remove(BookingType bt)
  throws DatabaseConnectionException
```

Removes the supplied object from the database

Parameters:

`bt` - bookingType contains all information for a bookingType

Returns:

true/false if the object is valid.

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

close

```
public void close()
```

Closes database connection

(continued from last page)

getBookingTypeList

```
public java.util.List getBookingTypeList()
  throws DatabaseConnectionException
```

Searches Database for required information and returns it in a list format

Returns:

list of all requested information from database

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

getBookingTypeInfo

```
public BookingType getBookingTypeInfo(int bookingType_id)
  throws DatabaseConnectionException
```

Parameters:

`bookingType_id`

Returns:

bookingType

Throws:

`DatabaseConnectionException`

logic
Class ClientManager

```
java.lang.Object
  |
  +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
          |
          +-logic.ClientManager
```

All Implemented Interfaces:
java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable

public class **ClientManager**
extends javax.servlet.http.HttpServlet

Servlet implementation class ClientManager

Constructor Summary

`public`	`ClientManager()` Gets the broker instance

Method Summary

`void`	`close()` Closes Database connection
`Client`	`getClientInfo(int id)` Gets organization broker and adds a new client to organization
`java.util.List`	`getClientList()` Gets needed information from Client table in Database and creates a new client to add to a client table
`boolean`	`remove(Client client)` Removes the supplied object from the database
`boolean`	`save(int id, java.lang.String givenName, java.lang.String surname, java.lang.String email, java.lang.String address, java.lang.String city, java.lang.String province, java.lang.String country, java.lang.String postalCode, java.lang.String discount, java.lang.String password, java.lang.String homePhone, java.lang.String workPhone, java.lang.String cellPhone, java.util.ArrayList organizations)` Validates all fields for id, givenName, surname, email, address, city, province, country, postalCode, discount, password, homePhone, workPhone, cellPhone, organizations to ensure that all fields are valid.
`java.util.List`	`search(java.lang.String query)` Sends a message to the database to search for requested information, and returns information in a list format.
`boolean`	`validate(Client c)` Validates all fields of an client object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field.

boolean	`validate(java.lang.String givenName, java.lang.String surname, java.lang.String email, java.lang.String address, java.lang.String city, java.lang.String province, java.lang.String country, java.lang.String postalCode, java.lang.String discount, java.lang.String password, java.lang.String homePhone, java.lang.String workPhone, java.lang.String cellPhone, java.util.ArrayList organizations)` Validates all fields of an client object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field.

Methods inherited from class `javax.servlet.http.HttpServlet`
`service`

Methods inherited from class `javax.servlet.GenericServlet`
`destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service`

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `javax.servlet.Servlet`
`destroy, getServletConfig, getServletInfo, init, service`

Methods inherited from interface `javax.servlet.ServletConfig`
`getInitParameter, getInitParameterNames, getServletContext, getServletName`

Constructors

ClientManager

```
public ClientManager()
```

Gets the broker instance

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

See Also:

`javax.servlet.http.HttpServlet()`

Methods

(continued from last page)

validate

```
public boolean validate(Client c)
  throws PostalCodeInvalidException,
         GivenNameInvalidException,
         SurnameInvalidException,
         AddressInvalidException,
         EmailInvalidException,
         CityInvalidException,
         CountryInvalidException,
         ProvinceInvalidException,
         DiscountInvalidException,
         HomePhoneInvalidException,
         WorkPhoneInvalidException,
         CellPhoneInvalidException
```

Validates all fields of an client object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field. If a field is not valid a appropriate exception will be thrown

Parameters:

`c` - client this is a client to validate

Returns:

true/false if the object is valid.

Throws:

`PostalCodeInvalidException` - is thrown if >6 characters, ==6 & !valid or ==5 digits
`GivenNameInvalidException` - is thrown if it is null or has >65 characters
`SurnameInvalidException` - is thrown if it is null or has >63 characters
`AddressInvalidException` - is thrown if it is null or has >75 characters
`EmailInvalidException` - is thrown if it is null or has >50 characters
`CityInvalidException` - is thrown if it is null or has >75 characters
`CountryInvalidException` - is thrown if it is null or has >25 characters
`ProvinceInvalidException` - is thrown if it is null or has >2 characters
`DiscountInvalidException` - is thrown if it is null, >3 characters, > 100% discount or <0% discount
`HomePhoneInvalidException` - is thrown if it is null or has >10 characters
`WorkPhoneInvalidException` - is thrown if it is null or has >10 characters
`CellPhoneInvalidException` - is thrown if it is null or has >10 characters

(continued from last page)

validate

```
public boolean validate(java.lang.String givenName,
         java.lang.String surname,
         java.lang.String email,
         java.lang.String address,
         java.lang.String city,
         java.lang.String province,
         java.lang.String country,
         java.lang.String postalCode,
         java.lang.String discount,
         java.lang.String password,
         java.lang.String homePhone,
         java.lang.String workPhone,
         java.lang.String cellPhone,
         java.util.ArrayList organizations)
  throws PostalCodeInvalidException,
         GivenNameInvalidException,
         SurnameInvalidException,
         AddressInvalidException,
         EmailInvalidException,
         CityInvalidException,
         CountryInvalidException,
         ProvinceInvalidException,
         DiscountInvalidException,
         HomePhoneInvalidException,
         WorkPhoneInvalidException,
         CellPhoneInvalidException
```

Validates all fields of an client object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field. If a field is not valid a appropriate exception will be thrown

Parameters:

`givenName` - is a givenName for the client
`surname` - is a surname for the client
`email` - is a email for the client
`address` - is a email for the client
`city` - is a id city the client
`province` - is a province for the client
`country` - is a country for the client
`postalCode` - is a postalCode for the client
`discount` - is a discount for the client
`password` - is a password for the client
`homePhone` - is a homePhone for the client
`workPhone` - is a workPhone for the client
`cellPhone` - is a cellPhone for the client
`organizations` - is a organizations for the client

Returns:

true/false if the object is valid.

Throws:

`PostalCodeInvalidException` - is thrown if >6 characters, ==6 & !valid or ==5 digits
`GivenNameInvalidException` - is thrown if it is null or has >65 characters
`SurnameInvalidException` - is thrown if it is null or has >63 characters
`AddressInvalidException` - is thrown if it is null or has >75 characters
`EmailInvalidException` - is thrown if it is null or has >50 characters
`CityInvalidException` - is thrown if it is null or has >75 characters
`CountryInvalidException` - is thrown if it is null or has >25 characters
`ProvinceInvalidException` - is thrown if it is null or has >2 characters
`DiscountInvalidException` - is thrown if it is null, >3 characters, > 100% discount or <0% discount
`HomePhoneInvalidException` - is thrown if it is null or has >10 characters
`WorkPhoneInvalidException` - is thrown if it is null or has >10 characters

(continued from last page)

`CellPhoneInvalidException` - is thrown if it is null or has >10 characters

getClientList

```
public java.util.List getClientList()
  throws DatabaseConnectionException
```

Gets needed information from Client table in Database and creates a new client to add to a client table

Returns:

ClientList returns selected clientList

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

close

```
public void close()
  throws java.sql.SQLException
```

Closes Database connection

Throws:

`SQLException`

save

```
public boolean save(int id,
         java.lang.String givenName,
         java.lang.String surname,
         java.lang.String email,
         java.lang.String address,
         java.lang.String city,
         java.lang.String province,
         java.lang.String country,
         java.lang.String postalCode,
         java.lang.String discount,
         java.lang.String password,
         java.lang.String homePhone,
         java.lang.String workPhone,
         java.lang.String cellPhone,
         java.util.ArrayList organizations)
  throws PostalCodeInvalidException,
         GivenNameInvalidException,
         SurnameInvalidException,
         AddressInvalidException,
         EmailInvalidException,
         CityInvalidException,
         CountryInvalidException,
         ProvinceInvalidException,
         DiscountInvalidException,
         DatabaseConnectionException,
         HomePhoneInvalidException,
         WorkPhoneInvalidException,
         CellPhoneInvalidException
```

Validates all fields for id, givenName, surname, email, address, city, province, country, postalCode, discount, password, homePhone, workPhone, cellPhone, organizations to ensure that all fields are valid. If a field is not valid a appropriate exception will be thrown If id exists, Client table is updated, if id != exist than information is inserted into Client table

Parameters:

`id` - is the id for the client

`givenName` - is the givenName for the client

`surname` - is the surname for the client

(continued from last page)

`email` - is the email for the client
`address` - is the email for the client
`city` - is the city for the client
`province` - is the province for the client
`country` - is the country for the client
`postalCode` - is the postalCode for the client
`discount` - is the discount for the client
`password` - is the password for the client
`homePhone` - is the homePhone for the client
`workPhone` - is the workPhone for the client
`cellPhone` - is the cellPhone for the client
`organizations` - is the organizations for the client

Returns:

updated list for client

Throws:

`PostalCodeInvalidException` - is thrown if >6 characters, ==6 & !valid or ==5 digits
`GivenNameInvalidException` - is thrown if it is null or has >65 characters
`SurnameInvalidException` - is thrown if it is null or has >63 characters
`AddressInvalidException` - is thrown if it is null or has >75 characters
`EmailInvalidException` - is thrown if it is null or has >50 characters
`CityInvalidException` - is thrown if it is null or has >75 characters
`CountryInvalidException` - is thrown if it is null or has >25 characters
`ProvinceInvalidException` - is thrown if it is null or has >2 characters
`DiscountInvalidException` - is thrown if it is null, >3 characters, > 100% discount or <0% discount
`HomePhoneInvalidException` - is thrown if it is null or has >10 characters
`WorkPhoneInvalidException` - is thrown if it is null or has >10 characters
`CellPhoneInvalidException` - is thrown if it is null or has >10 characters

remove

```
public boolean remove(Client client)
  throws DatabaseConnectionException
```

Removes the supplied object from the database

Parameters:

`client` - this is a variable that holds client information to remove from the database

Returns:

true/false if the object is valid.

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

getClientInfo

```
public Client getClientInfo(int id)
  throws DatabaseConnectionException
```

Gets organization broker and adds a new client to organization

Parameters:

`id` - is a id for the client

Returns:

clientInformation based on id

(continued from last page)

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

search

```
public java.util.List search(java.lang.String query)
  throws DatabaseConnectionException,
         java.sql.SQLException
```

Sends a message to the database to search for requested information, and returns information in a list format.

Parameters:

`query` - passes message along to database to search than returns a list

Returns:

List with all requested information from database

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails
`SQLException`

logic
Class EmployeeManager

```
java.lang.Object
  |
  +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
          |
          +-logic.EmployeeManager
```

All Implemented Interfaces:

java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable

---

public class **EmployeeManager**
extends javax.servlet.http.HttpServlet

Servlet implementation class EmployeeManager

Constructor Summary

public	EmployeeManager() Gets the broker instance

Method Summary

void	close() Closes Database connection
Employee	getEmployeeInfo(int employee_id)
java.util.List	getEmployeeList() Gets all information from table for that table
boolean	remove(Employee emp) Removes an employee object.
boolean	save(Employee emp) Saves/Updates an employee object if valid.
boolean	save(int id, java.lang.String username, java.lang.String password, java.lang.String givenName, java.lang.String surname, java.lang.String employeeLevel) Validates all fields for username, password, givenName, surname to ensure that all fields are not null.
java.util.List	search(java.lang.String search) Creates a list of employee objects that matched the string passed in.
boolean	validate(Employee emp) Validates all fields of an employee object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field.
boolean	validate(java.lang.String username, java.lang.String password, java.lang.String givenName, java.lang.String surname) Validates all fields of an employee object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field.

Methods inherited from class `javax.servlet.http.HttpServlet`
`service`

Methods inherited from class `javax.servlet.GenericServlet`
`destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service`

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `javax.servlet.Servlet`
`destroy, getServletConfig, getServletInfo, init, service`

Methods inherited from interface `javax.servlet.ServletConfig`
`getInitParameter, getInitParameterNames, getServletContext, getServletName`

Constructors

EmployeeManager

```
public EmployeeManager()
```

Gets the broker instance

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

See Also:

`javax.servlet.http.HttpServlet()`

Methods

validate

```
public boolean validate(Employee emp)
  throws GivenNameInvalidException,
         PasswordInvalidException,
         SurnameInvalidException,
         UsernameInvalidException
```

Validates all fields of an employee object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field. If a field is not valid a appropriate exception will be thrown

Parameters:

`emp` - the employee object to validate.

Returns:

true/false if the object is valid.

Throws:

`GivenNameInvalidException` - is thrown if it is null or has >25 characters

(continued from last page)

PasswordInvalidException - is thrown if it is <8 characters
SurnameInvalidException - is thrown if it is null or has >25 characters
UsernameInvalidException - is thrown if it is null or has >25 characters

validate

```
public boolean validate(java.lang.String username,
         java.lang.String password,
         java.lang.String givenName,
         java.lang.String surname)
  throws PasswordInvalidException,
         GivenNameInvalidException,
         SurnameInvalidException,
         UsernameInvalidException
```

Validates all fields of an employee object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field. If a field is not valid a appropriate exception will be thrown

Parameters:

`username` - is the username for the employee
`password` - is the password for the employee
`givenName` - is the givenName for the employee
`surname` - is the surname for the employee

Returns:

true/false if the object is valid.

Throws:

PasswordInvalidException - is thrown if it is <8 characters
GivenNameInvalidException - is thrown if it is null or has >25 characters
SurnameInvalidException - is thrown if it is null or has >25 characters
UsernameInvalidException - is thrown if it is null or has >25 characters

search

```
public java.util.List search(java.lang.String search)
  throws DatabaseConnectionException
```

Creates a list of employee objects that matched the string passed in. Searches through all employee in the database for any matches found.

Parameters:

`search` - string containing the information to look for.

Returns:

a List of employee objects that contain the search string.

Throws:

DatabaseConnectionException - is thrown if DB connection fails

save

```
public boolean save(Employee emp)
  throws DatabaseConnectionException
```

Saves/Updates an employee object if valid.

Parameters:

`emp` - employee object to commit.

(continued from last page)

Returns:

true/false if the object was saved.

Throws:

DatabaseConnectionException - is thrown if DB connection fails

save

```
public boolean save(int id,
         java.lang.String username,
         java.lang.String password,
         java.lang.String givenName,
         java.lang.String surname,
         java.lang.String employeeLevel)
  throws PasswordInvalidException,
         GivenNameInvalidException,
         SurnameInvalidException,
         UsernameInvalidException,
         DatabaseConnectionException
```

Validates all fields for username, password, givenName, surname to ensure that all fields are not null. If a field is null a appropriate exception will be thrown If id exists information in Employee table will be updated If id != exist information is inserted into Employee table

Parameters:

`username` - is the username for the employee

`password` - is the password for the employee

`givenName` - is the givenName for the employee

`surname` - is the surname for the employee

`employeeLevel` - is the level of access the employee has to the system

Returns:

updated list for Employee

Throws:

UsernameInvalidException - is thrown if it is null or has >25 characters

SurnameInvalidException - is thrown if it is null or has >25 characters

GivenNameInvalidException - is thrown if it is null or has >25 characters

PasswordInvalidException - is thrown if it is <8 characters

DatabaseConnectionException - is thrown if DB connection fails

remove

```
public boolean remove(Employee emp)
  throws DatabaseConnectionException
```

Removes an employee object.

Parameters:

`emp` - the employee object to commit.

Returns:

true/false if the object was removed.

Throws:

DatabaseConnectionException - is thrown if DB connection fails

close

```
public void close()
```

(continued from last page)

Closes Database connection

getEmployeeList

```
public java.util.List getEmployeeList()
  throws DatabaseConnectionException
```

Gets all information from table for that table

Returns:
List with all requested information from database

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails

getEmployeeInfo

```
public Employee getEmployeeInfo(int employee_id)
  throws DatabaseConnectionException
```

Parameters:
`employee_id`

Returns:
Employee

Throws:
`DatabaseConnectionException`

logic
Class FacilityManager

```
java.lang.Object
  |
  +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
          |
          +-logic.FacilityManager
```

All Implemented Interfaces:
java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable

---

public class **FacilityManager**
extends javax.servlet.http.HttpServlet

Servlet implementation class FacilityManager

Constructor Summary

`public`	`FacilityManager()` Gets the current Broker instance Throws appropriate exception if a error occurs

Method Summary

`void`	`close()` Closes Database connection
`Facility`	`getFacilityInfo(int facility_id)` Gets all information related to the passed facility id
`java.util.List`	`getFacilityList()` Updates resultSet, returns facilities
`boolean`	`remove(Facility facility)` Deletes information from facility table where facility_id = facility_getId If Id does not exist a exception will be thrown
`boolean`	`save(int id, java.lang.String openTime, java.lang.String closeTime, java.lang.String setupTime, java.lang.String tearDownTime, java.lang.String name, java.util.ArrayList rates, java.util.ArrayList additionalCharges, java.lang.String maxCapacity, java.lang.String minBookingInterval, java.lang.String minBookingTime)` Validates all fields for facility to ensure that all fields are not null.
`java.util.List`	`search(java.lang.String query)` Searches for specific information and returns facilities
`boolean`	`validate(Facility facility)` Validates all fields of an facility object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field.

boolean	validate(java.lang.String openTime, java.lang.String closeTime, java.lang.String setupTime, java.lang.String tearDownTime, java.lang.String name, java.util.ArrayList rates, java.util.ArrayList additionalCharges, java.lang.String maxCapacity, java.lang.String minBookingInterval, java.lang.String minBookingTime) Validates all fields of an facility object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field.

Methods inherited from class `javax.servlet.http.HttpServlet`
service

Methods inherited from class `javax.servlet.GenericServlet`
destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service

Methods inherited from class `java.lang.Object`
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface `javax.servlet.Servlet`
destroy, getServletConfig, getServletInfo, init, service

Methods inherited from interface `javax.servlet.ServletConfig`
getInitParameter, getInitParameterNames, getServletContext, getServletName

Constructors

FacilityManager

`public` **`FacilityManager`**`()`

Gets the current Broker instance Throws appropriate exception if a error occurs

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

See Also:

`javax.servlet.http.HttpServlet()`

Methods

(continued from last page)

validate

```
public boolean validate(java.lang.String openTime,
         java.lang.String closeTime,
         java.lang.String setupTime,
         java.lang.String tearDownTime,
         java.lang.String name,
         java.util.ArrayList rates,
         java.util.ArrayList additionalCharges,
         java.lang.String maxCapacity,
         java.lang.String minBookingInterval,
         java.lang.String minBookingTime)
  throws NameInvalidException,
         DatabaseConnectionException,
         OpenTimeInvalidException,
         CloseTimeInvalidException,
         SetupTimeInvalidException,
         TearDownTimeInvalidException,
         MaxCapacityInvalidException,
         MinBookingIntervalInvalidException,
         MinBookingTimeInvalidException
```

Validates all fields of an facility object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field. If a field is not valid a appropriate exception will be thrown

Parameters:

`openTime` - is the openTime for the facility
`closeTime` - is the closeTime for the facility
`setupTime` - is the setupTime for the facility
`tearDownTime` - is the tearDownTime for the facility
`name` - is the name for the facility
`rates` - is the rates for the facility
`additionalCharges` - is the additionalCharges for the facility
`maxCapacity` - is the maxCapacity for the facility
`minBookingInterval` - is the minBookingInterval for the facility
`minBookingTime` - is the minBookingTime for the facility

Returns:

true/false if the object is valid.

Throws:

`CostInvalidException` - is thrown if it not a number,or != double
`NameInvalidException` - is thrown if it is null or has >30 characters
`DatabaseConnectionException` - is thrown if DB connection fails
`DescriptionInvalidException` - is thrown if it is null or has >400 characters
`MinBookingTimeInvalidException` - is thrown if it is null or is not in a minute format
`MinBookingIntervalInvalidException` - is thrown if it is not in the proper number format
`MaxCapacityInvalidException` - is thrown if it is null or if it is not a positive number
`TearDownTimeInvalidException` - is thrown if it is null, is a negative number or is not in the proper number format
`SetupTimeInvalidException` - is thrown if it is null, is a negative number or is not in the proper number format
`CloseTimeInvalidException` - is thrown if it is null, is <0, >24 or not in the proper number format
`OpenTimeInvalidException` - is thrown if it is null, is <0, >24 or not in the proper number format

(continued from last page)

validate

```
public boolean validate(Facility facility)
  throws NameInvalidException,
         DatabaseConnectionException,
         OpenTimeInvalidException,
         CloseTimeInvalidException,
         SetupTimeInvalidException,
         TearDownTimeInvalidException,
         MaxCapacityInvalidException,
         MinBookingIntervalInvalidException,
         MinBookingTimeInvalidException
```

Validates all fields of an facility object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field. If a field is not valid a appropriate exception will be thrown

Parameters:

`facility` - the facility object to validate

Returns:

true/false if the object is valid.

Throws:

`CostInvalidException` - is thrown if it is <0 number
`NameInvalidException` - is thrown if it is null or has >30 characters
`DatabaseConnectionException` - is thrown if DB connection fails
`DescriptionInvalidException` - is thrown if it is null or has >400 characters
`MinBookingTimeInvalidException` - is thrown if it is null or is not in a minute format
`MinBookingIntervalInvalidException` - is thrown if it is not in the proper number format
`MaxCapacityInvalidException` - is thrown if it is null or if it is not a positive number
`TearDownTimeInvalidException` - is thrown if it is null, is a negative number or is not in the proper number format
`SetupTimeInvalidException` - is thrown if it is null, is a negative number or is not in the proper number format
`CloseTimeInvalidException` - is thrown if it is null, is <0, >24 or not in the proper number format
`OpenTimeInvalidException` - is thrown if it is null, is <0, >24 or not in the proper number format

search

```
public java.util.List search(java.lang.String query)
  throws DatabaseConnectionException
```

Searches for specific information and returns facilities

Parameters:

`query` - searches database based on query and returns specified information

Returns:

a List of facility objects that contain the search string.

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

(continued from last page)

save

```
public boolean save(int id,
           java.lang.String openTime,
           java.lang.String closeTime,
           java.lang.String setupTime,
           java.lang.String tearDownTime,
           java.lang.String name,
           java.util.ArrayList rates,
           java.util.ArrayList additionalCharges,
           java.lang.String maxCapacity,
           java.lang.String minBookingInterval,
           java.lang.String minBookingTime)
  throws NameInvalidException,
           DatabaseConnectionException,
           OpenTimeInvalidException,
           CloseTimeInvalidException,
           SetupTimeInvalidException,
           TearDownTimeInvalidException,
           MaxCapacityInvalidException,
           MinBookingIntervalInvalidException,
           MinBookingTimeInvalidException
```

Validates all fields for facility to ensure that all fields are not null. If a field is null a appropriate exception will be thrown If id exists, invoice table is updated, if id != exist than information is inserted into Invoice table

Returns:
true/false if the object was saved.

Throws:
`CostInvalidException` - is thrown if it is <0 number
`NumberFormatException` - is thrown if it is not in the proper numberFormat
`NameInvalidException` - is thrown if it is null or has >30 characters
`DescriptionInvalidException` - is thrown if it is null or has >400 characters
`DatabaseConnectionException` - is thrown if DB connection fails
`MinBookingTimeInvalidException` - is thrown if it is null or is not in a minute format
`MinBookingIntervalInvalidException` - is thrown if it is not in the proper number format
`MaxCapacityInvalidException` - is thrown if it is null or if it is not a positive number
`TearDownTimeInvalidException` - is thrown if it is null, is a negative number or is not in the proper number format
`SetupTimeInvalidException` - is thrown if it is null, is a negative number or is not in the proper number format
`CloseTimeInvalidException` - is thrown if it is null, is <0, >24 or not in the proper number format
`OpenTimeInvalidException` - is thrown if it is null, is <0, >24 or not in the proper number format

remove

```
public boolean remove(Facility facility)
  throws DatabaseConnectionException
```

Deletes information from facility table where facility_id = facility_getId If Id does not exist a exception will be thrown

Parameters:
`facility` - the facility object to validate and remove from database

Returns:
true/false if the object is valid.

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails

(continued from last page)

close

```
public void close()
```

Closes Database connection

getFacilityList

```
public java.util.List getFacilityList()
  throws DatabaseConnectionException
```

Updates resultSet, returns facilities

Returns:

updated resultSet

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

getFacilityInfo

```
public Facility getFacilityInfo(int facility_id)
  throws DatabaseConnectionException
```

Gets all information related to the passed facility id

Parameters:

`facility_id` - the id of the facility to retrieve information for

Returns:

a Facility containing all information found

Throws:

`DatabaseConnectionException`

logic
Class InvoiceManager

```
java.lang.Object
  |
  +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
          |
          +-logic.InvoiceManager
```

All Implemented Interfaces:
java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable

public class **InvoiceManager**
extends javax.servlet.http.HttpServlet

Servlet implementation class InvoiceManager

Constructor Summary	
public	InvoiceManager() Gets the broker instance

Method Summary	
void	close() Closes Database connection
java.util.List	getAllUnPaidInvoices()
Invoice	getInvoiceInfo(int invoice_no)
java.util.List	getInvoicesForClient(int client_id)
java.util.List	getPayments(int invoice_id)
boolean	isPaid(int invoice_id)
void	payInvoice(Invoice i, Payment p)
double	paymentOwed(int invoice_id)
boolean	remove(Invoice invoice) Deletes information from invoice table where invoice_id = invoice_getId If Id does not exist a exception will be thrown
boolean	save(int id, java.util.Date date, Client client, java.util.ArrayList bookings, java.lang.String description) Validates all fields for date, client, bookings, paid, paymentDue, description dueDate and payments to ensure that all fields are not null.

`boolean`	`save(Invoice invoice)` Validates all fields for invoice to ensure that all fields are not null.
`java.util.List`	`search(java.lang.String search)` Searches the Database for matching Invoices
`boolean`	`validate(java.util.Date date, Client client, java.util.ArrayList bookings, java.lang.String description)` Validates all fields of an facility object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field.
`boolean`	`validate(Invoice invoice)` Validates all fields of an facility object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field.

Methods inherited from class `javax.servlet.http.HttpServlet`

`service`

Methods inherited from class `javax.servlet.GenericServlet`

`destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service`

Methods inherited from class `java.lang.Object`

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

Methods inherited from interface `javax.servlet.Servlet`

`destroy, getServletConfig, getServletInfo, init, service`

Methods inherited from interface `javax.servlet.ServletConfig`

`getInitParameter, getInitParameterNames, getServletContext, getServletName`

Constructors

InvoiceManager

`public **InvoiceManager**()`

Gets the broker instance

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

See Also:

`javax.servlet.http.HttpServlet()`

Methods

(continued from last page)

validate

```
public boolean validate(java.util.Date date,
         Client client,
         java.util.ArrayList bookings,
         java.lang.String description)
  throws DescriptionInvalidException,
         DatabaseConnectionException,
         DateInvalidException
```

Validates all fields of an facility object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field. If a field is not valid a appropriate exception will be thrown If a field is null a appropriate exception will be thrown

Parameters:

`date` - is the date for the invoice

`client` - is the client for the invoice

`bookings` - is the bookings for the invoice

`description` - is the description for the invoice

Returns:

true/false if the object is valid.

Throws:

`DescriptionInvalidException` - is thrown if it is null or has >400 characters

`DatabaseConnectionException` - is thrown if DB connection fails

`DateInvalidException` - is thrown if date is null or the invoice date is before current date

validate

```
public boolean validate(Invoice invoice)
  throws CostInvalidException,
         DescriptionInvalidException,
         BooleanInvalidException,
         DatabaseConnectionException,
         GivenNameInvalidException,
         SurnameInvalidException,
         AddressInvalidException,
         EmailInvalidException,
         CityInvalidException,
         CountryInvalidException,
         PostalCodeInvalidException,
         ProvinceInvalidException,
         NameInvalidException,
         ScheduleInvalidException,
         SetupTimeInvalidException,
         TearDownTimeInvalidException,
         NumberInvalidException,
         DateInvalidException
```

Validates all fields of an facility object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field. If a field is not valid a appropriate exception will be thrown

Parameters:

`invoice`

Returns:

true/false if the object is valid.

Throws:

`CostInvalidException` - is thrown if it not a number or !=double

`DescriptionInvalidException` - is thrown if it is null or has >400 characters

(continued from last page)

`BooleanInvalidException` - is thrown if boolean not true/false
`DatabaseConnectionException` - is thrown if DB connection fails
`AddressInvalidException` - is thrown if it is null or has >75 characters
`SurnameInvalidException` - is thrown if it is null or has >63 characters
`GivenNameInvalidException` - is thrown if it is null or has >65 characters
`ProvinceInvalidException` - is thrown if it is null or has >2 characters
`PostalCodeInvalidException` - is thrown if >6 characters, ==6 & !valid or ==5 digits
`CountryInvalidException` - is thrown if it is null or has >25 characters
`CityInvalidException` - is thrown if it is null or has >75 characters
`EmailInvalidException` - is thrown if it is null or has >50 characters
`NumberInvalidException` - is thrown if setupTime or tearDownTime are not in a numerical format
`TearDownTimeInvalidException` - is thrown if it is <0 minutes
`SetupTimeInvalidException` - is thrown if it is <0 minutes
`ScheduleInvalidException` - Checks Schedule to see if it exists, that time is free,and that it is not to big. if any of these are not true than the exception is thrown.
`NameInvalidException` - is thrown if name is null or >25 characters
`DateInvalidException` - is thrown if date is null or the invoice date is before current date

search

```
public java.util.List search(java.lang.String search)
  throws DatabaseConnectionException,
         java.sql.SQLException
```

Searches the Database for matching Invoices

Parameters:

`search` - searches database based on query and returns specified information

Returns:

a List containing all

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails
`SQLException`

save

```
public boolean save(int id,
         java.util.Date date,
         Client client,
         java.util.ArrayList bookings,
         java.lang.String description)
  throws DescriptionInvalidException,
         DatabaseConnectionException,
         DateInvalidException
```

Validates all fields for date, client, bookings, paid, paymentDue, description dueDate and payments to ensure that all fields are not null. If a field is null a appropriate exception will be thrown If id exists, invoice table is updated, if id != exist than information is inserted into Invoice table

Parameters:

`date` - is the date for the invoice
`client` - is the client for the invoice
`bookings` - is the bookings for the invoice
`description` - is the description for the invoice

Returns:

true/false if the object is valid.

Throws:

(continued from last page)

DescriptionInvalidException - is thrown if it is null or has >400 characters
DatabaseConnectionException - is thrown if DB connection fails
DateInvalidException - is thrown if date is null or the invoice date is before current date

save

```
public boolean save(Invoice invoice)
  throws CostInvalidException,
         DescriptionInvalidException,
         BooleanInvalidException,
         DatabaseConnectionException,
         GivenNameInvalidException,
         SurnameInvalidException,
         AddressInvalidException,
         EmailInvalidException,
         CityInvalidException,
         CountryInvalidException,
         PostalCodeInvalidException,
         ProvinceInvalidException,
         NameInvalidException,
         ScheduleInvalidException,
         SetupTimeInvalidException,
         TearDownTimeInvalidException,
         NumberInvalidException,
         DateInvalidException
```

Validates all fields for invoice to ensure that all fields are not null. If a field is null a appropriate exception will be thrown If id exists, invoice table is updated, if id != exist than information is inserted into Invoice table

Parameters:

invoice - the invoice object to validate and persist

Returns:

true/false if the object was saved.

Throws:

CostInvalidException - is thrown if it not a number or !=double
DescriptionInvalidException - is thrown if it is null or has >400 characters
BooleanInvalidException - is thrown if boolean not true/false
DatabaseConnectionException - is thrown if DB connection fails
AddressInvalidException - is thrown if it is null or has >75 characters
SurnameInvalidException - is thrown if it is null or has >63 characters
GivenNameInvalidException - is thrown if it is null or has >65 characters
ProvinceInvalidException - is thrown if it is null or has >2 characters
PostalCodeInvalidException - is thrown if >6 characters, ==6 & !valid or ==5 digits
CountryInvalidException - is thrown if it is null or has >25 characters
CityInvalidException - is thrown if it is null or has >75 characters
EmailInvalidException - is thrown if it is null or has >50 characters
NumberInvalidException - is thrown if setupTime or tearDownTime are not in a numerical format
TearDownTimeInvalidException - is thrown if it is <0 minutes
SetupTimeInvalidException - is thrown if it is <0 minutes
ScheduleInvalidException - Checks Schedule to see if it exists, that time is free,and that it is not to big. if any of these are not true than the exception is thrown.
NameInvalidException - is thrown if name is null or >25 characters
DateInvalidException - is thrown if date is null or the invoice date is before current date

remove

```
public boolean remove(Invoice invoice)
  throws DatabaseConnectionException
```

Deletes information from invoice table where invoice_id = invoice_getId If Id does not exist a exception will be thrown

(continued from last page)

Parameters:
`invoice` - the invoice object to remove

Returns:
true/false if the object is valid.

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails

isPaid

```
public boolean isPaid(int invoice_id)
  throws DatabaseConnectionException
```

Parameters:
`invoice_id` - is id for specific id

Returns:
boolean for ispaid

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails

paymentOwed

```
public double paymentOwed(int invoice_id)
  throws DatabaseConnectionException
```

getPayments

```
public java.util.List getPayments(int invoice_id)
  throws DatabaseConnectionException
```

getInvoiceInfo

```
public Invoice getInvoiceInfo(int invoice_no)
  throws DatabaseConnectionException,
         java.sql.SQLException
```

getInvoicesForClient

```
public java.util.List getInvoicesForClient(int client_id)
  throws DatabaseConnectionException,
         java.sql.SQLException
```

close

```
public void close()
```

(continued from last page)

Closes Database connection

payInvoice

```
public void payInvoice(Invoice i,
         Payment p)
```

getAllUnPaidInvoices

```
public java.util.List getAllUnPaidInvoices()
  throws DatabaseConnectionException,
         java.sql.SQLException
```

logic
Class LoginManager

```
java.lang.Object
  |
  +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
          |
          +-logic.LoginManager
```

All Implemented Interfaces:
java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable

public class **LoginManager**
extends javax.servlet.http.HttpServlet

Constructor Summary

public	LoginManager()

Methods inherited from class javax.servlet.http.HttpServlet

service

Methods inherited from class javax.servlet.GenericServlet

destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface javax.servlet.Servlet

destroy, getServletConfig, getServletInfo, init, service

Methods inherited from interface javax.servlet.ServletConfig

getInitParameter, getInitParameterNames, getServletContext, getServletName

Constructors

LoginManager

public **LoginManager**()

logic
Class OrganizationManager

```
java.lang.Object
  |
  +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
          |
          +-logic.OrganizationManager
```

All Implemented Interfaces:
java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable

public class **OrganizationManager**
extends javax.servlet.http.HttpServlet

Servlet implementation class OrganizationManager

Constructor Summary

`public`	`OrganizationManager()` Gets the current broker instance

Method Summary

`void`	`close()` closes Database Connection
`Organization`	`getOrgInfo(int id)`
`java.util.List`	`getOrgList()` Gets organization list from organization table
`boolean`	`remove(Organization org)` Deletes information from organization table where organization_id = organization_getId If id does not exist a exception will be thrown
`boolean`	`save(int id, java.lang.String name, java.lang.String discount, java.lang.String contact, java.lang.String description)` Checks each of the name, discount, contact and description fields to ensure they are valid, if any field is null, a appropriate exception will be thrown.
`java.util.List`	`search(java.lang.String query)` Searches the Database for matching organizations
`boolean`	`validate(Organization org)` Validates all fields of an organization object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field.
`boolean`	`validate(java.lang.String name, java.lang.String discount, java.lang.String contact, java.lang.String description)` Validates all fields of an organization object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field.

Methods inherited from class `javax.servlet.http.HttpServlet`

`service`

Methods inherited from class `javax.servlet.GenericServlet`
`destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service`

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `javax.servlet.Servlet`
`destroy, getServletConfig, getServletInfo, init, service`

Methods inherited from interface `javax.servlet.ServletConfig`
`getInitParameter, getInitParameterNames, getServletContext, getServletName`

Constructors

OrganizationManager

```
public OrganizationManager()
```

Gets the current broker instance

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

See Also:

`javax.servlet.http.HttpServlet()`

Methods

validate

```
public boolean validate(java.lang.String name,
         java.lang.String discount,
         java.lang.String contact,
         java.lang.String description)
  throws NameInvalidException,
         DiscountInvalidException,
         DescriptionInvalidException,
         DatabaseConnectionException,
         ClientNameNotUniqueException,
         ClientDoesNotExistException,
         java.sql.SQLException
```

Validates all fields of an organization object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field. If a field is not valid a appropriate exception will be thrown

Parameters:

`name` - is the name for the organization
`discount` - is the discount for the organization
`contact` - is the contact for the organization
`description` - is the description for the organization

(continued from last page)

Returns:

true/false if the object is valid.

Throws:

NameInvalidException - is thrown if it is null or has >30 characters
DiscountInvalidException - is thrown if it is null, has <0 number or is not in the proper number format
DescriptionInvalidException - is thrown if it is null or has >400 characters
DatabaseConnectionException - is thrown if DB connection fails
GivenNameInvalidException - is thrown if it is null or has >65 characters
SurnameInvalidException - is thrown if it is null or has >63 characters
AddressInvalidException - is thrown if it is null or has >75 characters
EmailInvalidException - is thrown if it is null or has >50 characters
CityInvalidException - is thrown if it is null or has >75 characters
CountryInvalidException - is thrown if it is null or has >25 characters
PostalCodeInvalidException - is thrown if >6 characters, ==6 & !valid or ==5 digits
ProvinceInvalidException - is thrown if it is null or has >2 characters
ClientNameNotUniqueException
ClientDoesNotExistException
SQLException

validate

```
public boolean validate(Organization org)
  throws NameInvalidException,
         DiscountInvalidException,
         DescriptionInvalidException,
         DatabaseConnectionException,
         ClientNameNotUniqueException,
         ClientDoesNotExistException,
         java.sql.SQLException
```

Validates all fields of an organization object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field. If a field is not valid a appropriate exception will be thrown

Parameters:

org - is the organization object that holds information about the organization

Returns:

true/false if the object is valid.

Throws:

NameInvalidException - is thrown if it is null or has >30 characters
DiscountInvalidException - is thrown if it is null, has <0 number or is not in the proper number format
DescriptionInvalidException - is thrown if it is null or has >400 characters
DatabaseConnectionException - is thrown if DB connection fails
GivenNameInvalidException - is thrown if it is null or has >65 characters
SurnameInvalidException - is thrown if it is null or has >63 characters
AddressInvalidException - is thrown if it is null or has >75 characters
EmailInvalidException - is thrown if it is null or has >50 characters
CityInvalidException - is thrown if it is null or has >75 characters
CountryInvalidException - is thrown if it is null or has >25 characters
PostalCodeInvalidException - is thrown if >6 characters, ==6 & !valid or ==5 digits
ProvinceInvalidException - is thrown if it is null or has >2 characters
ClientDoesNotExistException
ClientNameNotUniqueException
SQLException

(continued from last page)

search

```
public java.util.List search(java.lang.String query)
  throws DatabaseConnectionException,
         java.sql.SQLException
```

Searches the Database for matching organizations

Parameters:

`query` - searches database based on query and returns specified information

Returns:

a List of Organization objects that contain the search string.

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

`SQLException`

save

```
public boolean save(int id,
         java.lang.String name,
         java.lang.String discount,
         java.lang.String contact,
         java.lang.String description)
  throws NameInvalidException,
         DiscountInvalidException,
         DescriptionInvalidException,
         DatabaseConnectionException,
         ClientNameNotUniqueException,
         ClientDoesNotExistException,
         java.sql.SQLException
```

Checks each of the name, discount, contact and description fields to ensure they are valid, if any field is null, a appropriate exception will be thrown.

Parameters:

`name` - is the name for the organization

`discount` - is the discount for the organization

`contact` - is the contact for the organization

`description` - is the description for the organization

Returns:

true/false if the object is valid.

Throws:

`NumberFormatException` - is thrown if it is not in the proper number format

`NameInvalidException` - is thrown if it is null or has >30 characters

`DiscountInvalidException` - is thrown if it is null, has <0 number or is not in the proper number format

`DescriptionInvalidException` - is thrown if it is null or has >400 characters

`DatabaseConnectionException` - is thrown if DB connection fails

`GivenNameInvalidException` - is thrown if it is null or has >65 characters

`SurnameInvalidException` - is thrown if it is null or has >63 characters

`AddressInvalidException` - is thrown if it is null or has >75 characters

`CityInvalidException` - is thrown if it is null or has >75 characters

`CountryInvalidException` - is thrown if it is null or has >25 characters

`PostalCodeInvalidException` - is thrown if >6 characters, ==6 & !valid or ==5 digits

`ProvinceInvalidException` - is thrown if it is null or has >2 characters

`ClientDoesNotExistException`

`ClientNameNotUniqueException`

`SQLException`

remove

```
public boolean remove(Organization org)
  throws DatabaseConnectionException
```

Deletes information from organization table where organization_id = organization_getId If id does not exist a exception will be thrown

Parameters:

`org` - is the organization object that holds information about the organization

Returns:

true/false if the object is valid.

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

close

```
public void close()
```

closes Database Connection

getOrgList

```
public java.util.List getOrgList()
  throws DatabaseConnectionException,
         java.sql.SQLException
```

Gets organization list from organization table

Returns:

requested information found in the database in a list

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

`SQLException`

getOrgInfo

```
public Organization getOrgInfo(int id)
  throws DatabaseConnectionException,
         java.sql.SQLException
```

logic
Class PublicLoginManager

```
java.lang.Object
  |
  +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
          |
          +-logic.PublicLoginManager
```

All Implemented Interfaces:
java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable

public class **PublicLoginManager**
extends javax.servlet.http.HttpServlet

Constructor Summary

public	PublicLoginManager()

Methods inherited from class `javax.servlet.http.HttpServlet`

`service`

Methods inherited from class `javax.servlet.GenericServlet`

`destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service`

Methods inherited from class `java.lang.Object`

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

Methods inherited from interface `javax.servlet.Servlet`

`destroy, getServletConfig, getServletInfo, init, service`

Methods inherited from interface `javax.servlet.ServletConfig`

`getInitParameter, getInitParameterNames, getServletContext, getServletName`

Constructors

PublicLoginManager

`public PublicLoginManager()`

logic
Class RateManager

```
java.lang.Object
  |
  +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
          |
          +-logic.RateManager
```

All Implemented Interfaces:
java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable

public class **RateManager**
extends javax.servlet.http.HttpServlet

Constructor Summary

`public`	`RateManager()` Gets rateBroker instance, if an occurs exception will be thrown.

Method Summary

`void`	`close()` Closes the RateBroker
`Rate`	`getRateInfo(int rates_id)`
`java.util.List`	`getRateList()` Gets all information from rate table in database
`boolean`	`remove(Rate rate)` Deletes information from rate table, where rate_id= rate.getId if the id does not exist a exception will be thrown.
`boolean`	`save(int id, java.lang.String name, java.lang.String description, java.lang.String rateCost, java.lang.String damageDeposit, java.lang.String bookingDeposit, java.lang.String isHourly, java.lang.String validStartTime, java.lang.String validEndTime, java.lang.String sunday, java.lang.String monday, java.lang.String tuesday, java.lang.String wednesday, java.lang.String thursday, java.lang.String friday, java.lang.String saturday)` If id exists the rate table is updated, otherwise a new rate is inserted into the table
`boolean`	`save(Rate rate)` If id exists, rate table is updated, if it does not exist it is added to rate table.
`java.util.List`	`search(java.lang.String query)` Searches database for matching rates
`boolean`	`validate(Rate rate)` Validates all fields of an rate object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field.

boolean	`validate(java.lang.String name, java.lang.String description, java.lang.String rateCost, java.lang.String damageDeposit, java.lang.String bookingDeposit)` Validates all fields of an rate object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field.

Methods inherited from class `javax.servlet.http.HttpServlet`
`service`

Methods inherited from class `javax.servlet.GenericServlet`
`destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service`

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `javax.servlet.Servlet`
`destroy, getServletConfig, getServletInfo, init, service`

Methods inherited from interface `javax.servlet.ServletConfig`
`getInitParameter, getInitParameterNames, getServletContext, getServletName`

Constructors

RateManager

```
public RateManager()
```

Gets rateBroker instance, if an occurs exception will be thrown.

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

Methods

validate

```
public boolean validate(Rate rate)
  throws NameInvalidException,
         DescriptionInvalidException,
         CostInvalidException,
         DamageDepositInvalidException,
         BookingDepositInvalidException
```

Validates all fields of an rate object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field. If a field is not valid a appropriate exception will be thrown

Parameters:

`rate` - holds all information required by the rate

Returns:

true/false if the object is valid.

(continued from last page)

Throws:

`NameInvalidException` - is thrown if it is null or has >30 characters
`DescriptionInvalidException` - is thrown if it is null or has >400 characters
`CostInvalidException` - is thrown if it not a number or !=double
`DamageDepositInvalidException` - is thrown if it is null, <0 or not in the proper number format
`BookingDepositInvalidException` - is thrown if it is null, <0 or not in the proper number format

validate

```
public boolean validate(java.lang.String name,
         java.lang.String description,
         java.lang.String rateCost,
         java.lang.String damageDeposit,
         java.lang.String bookingDeposit)
  throws CostInvalidException,
         NameInvalidException,
         DescriptionInvalidException,
         DamageDepositInvalidException,
         BookingDepositInvalidException
```

Validates all fields of an rate object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field. If a field is not valid a appropriate exception will be thrown

Parameters:

`name` - is the name for the Rate
`description` - is the description for the Rate
`rateCost` - is the rateCost for the Rate
`damageDeposit` - is the damageDeposit for the Rate
`bookingDeposit` - is the bookingDeposit for the Rate

Returns:

true/false if the object is valid.

Throws:

`NameInvalidException` - is thrown if it is null or has >30 characters
`DescriptionInvalidException` - is thrown if it is null or has >400 characters
`CostInvalidException` - is thrown if it not a number or !=double
`DamageDepositInvalidException` - is thrown if it is null, <0 or not in the proper number format
`BookingDepositInvalidException` - is thrown if it is null, <0 or not in the proper number format

search

```
public java.util.List search(java.lang.String query)
  throws DatabaseConnectionException
```

Searches database for matching rates

Parameters:

`query` - searches database based on query and returns specified information

Returns:

a List of facility objects that contain the search string.

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

(continued from last page)

save

```
public boolean save(int id,
         java.lang.String name,
         java.lang.String description,
         java.lang.String rateCost,
         java.lang.String damageDeposit,
         java.lang.String bookingDeposit,
         java.lang.String isHourly,
         java.lang.String validStartTime,
         java.lang.String validEndTime,
         java.lang.String sunday,
         java.lang.String monday,
         java.lang.String tuesday,
         java.lang.String wednesday,
         java.lang.String thursday,
         java.lang.String friday,
         java.lang.String saturday)
  throws CostInvalidException,
         NameInvalidException,
         DescriptionInvalidException,
         DatabaseConnectionException,
         DamageDepositInvalidException,
         BookingDepositInvalidException,
         java.text.ParseException
```

If id exists the rate table is updated, otherwise a new rate is inserted into the table

Parameters:

`id` - is the id for the Rate
`name` - is the name for the Rate
`description` - is the description for the Rate
`rateCost` - is the rateCost for the Rate
`damageDeposit` - is the damageDeposit for the Rate
`bookingDeposit` - is the bookingDeposit for the Rate

Returns:

true/false if the object is valid.

Throws:

`CostInvalidException` - is thrown if it not a number or !=double
`NameInvalidException` - is thrown if it is null or has >30 characters
`DescriptionInvalidException` - is thrown if it is null or has >400 characters
`NumberFormatException` - is thrown if field is not entered in a proper number format
`DatabaseConnectionException` - is thrown if DB connection fails
`DamageDepositInvalidException` - is thrown if it is null, <0 or not in the proper number format
`NumberFormatException` - is thrown if a field is not in the proper number format
`BookingDepositInvalidException` - is thrown if it is null, <0 or not in the proper number format
`ParseException`

save

```
public boolean save(Rate rate)
  throws NameInvalidException,
         DescriptionInvalidException,
         CostInvalidException,
         DatabaseConnectionException,
         DamageDepositInvalidException,
         BookingDepositInvalidException
```

If id exists, rate table is updated, if it does not exist it is added to rate table. appropriate exceptions are thrown as needed

Parameters:

(continued from last page)

`rate` - holds all information required by the rate

Returns:

true/false if the object is valid.

Throws:

`NameInvalidException` - is thrown if it is null or has >30 characters
`DescriptionInvalidException` - is thrown if it is null or has >400 characters
`CostInvalidException` - is thrown if it not a number or !=double
`DatabaseConnectionException` - is thrown if DB connection fails
`DamageDepositInvalidException` - is thrown if it is null, <0 or not in the proper number format
`BookingDepositInvalidException` - is thrown if it is null, <0 or not in the proper number format

remove

```
public boolean remove(Rate rate)
  throws DatabaseConnectionException
```

Deletes information from rate table, where rate_id= rate.getId if the id does not exist a exception will be thrown.

Parameters:

`rate` - holds all information required by the rate

Returns:

true/false if the object is valid.

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

close

```
public void close()
```

Closes the RateBroker

getRateList

```
public java.util.List getRateList()
  throws DatabaseConnectionException
```

Gets all information from rate table in database

Returns:

all information for rate table in a list format

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

getRateInfo

```
public Rate getRateInfo(int rates_id)
  throws DatabaseConnectionException
```

logic
Class ToDoItemManager

```
java.lang.Object
  |
  +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
          |
          +-logic.ToDoItemManager
```

All Implemented Interfaces:
java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable

public class **ToDoItemManager**
extends javax.servlet.http.HttpServlet

Constructor Summary

`public`	`ToDoItemManager()` Gets toDoItemBroker instance, if an occurs exception will be thrown.

Method Summary

`void`	`close()` Closes the database connection
`java.util.List`	`getAll()` Gets all information from toDoItem table in database
`boolean`	`remove(ToDoItem todo)` Deletes information from toDoItem table where toDoItem_id = toDoItem_getId If id does not exist a exception will be thrown
`boolean`	`save(int id, java.lang.String ToDoName, java.lang.String ToDoDescription, java.lang.String ToDoDate)` Validates toDoName, toDoDescription to ensure they meet all criteria Checks toDoDate to ensure that it is not null, if it is a exception will be thrown Persist: will persist the supplied item into the DB
`boolean`	`save(ToDoItem todo)` HttpServletRequest and HttpServletResponse checks toDoName and toDoDescription to ensure they meet all criteria, checks toDoDate to ensure it is not null.
`java.util.List`	`search(java.lang.String query)` Searches the DB for matching toDoItems
`boolean`	`validate(java.lang.String ToDoName, java.lang.String ToDoDescription, java.lang.String ToDoDate)` Validates all fields of an toDoItem object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field.
`boolean`	`validate(ToDoItem todo)` Validates all fields of an toDoitem object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field.

Methods inherited from class `javax.servlet.http.HttpServlet`

service

Methods inherited from class `javax.servlet.GenericServlet`
`destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service`

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `javax.servlet.Servlet`
`destroy, getServletConfig, getServletInfo, init, service`

Methods inherited from interface `javax.servlet.ServletConfig`
`getInitParameter, getInitParameterNames, getServletContext, getServletName`

Constructors

ToDoItemManager

```
public ToDoItemManager()
```

Gets toDoItemBroker instance, if an occurs exception will be thrown.

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

Methods

validate

```
public boolean validate(java.lang.String ToDoName,
                        java.lang.String ToDoDescription,
                        java.lang.String ToDoDate)
                 throws NameInvalidException,
                        DescriptionInvalidException,
                        DateInvalidException
```

Validates all fields of an toDoItem object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field. If a field is not valid a appropriate exception will be thrown

Parameters:

`ToDoName` - is the toDoName for the ToDoItem
`ToDoDescription` - is the toDoDescription for the ToDoItem
`ToDoDate` - is the ToDoDate for the ToDoItem

Returns:

true/false if the object is valid.

Throws:

`NameInvalidException` - is thrown if it is null or has >30 characters
`DescriptionInvalidException` - is thrown if it is null or has >400 characters
`DateInvalidException` - is thrown if date is null or is in the past

validate

```
public boolean validate(ToDoItem todo)
  throws NameInvalidException,
         DescriptionInvalidException,
         DateInvalidException
```

Validates all fields of an toDoitem object to ensure that the fields can be committed to the database without error and users correctly inputed information for each field. If a field is not valid a appropriate exception will be thrown

Parameters:
`todo` - holds all information required by the toDo

Returns:
true/false if the object is valid.

Throws:
`NameInvalidException` - is thrown if it is null or has >30 characters
`DescriptionInvalidException` - is thrown if it is null or has >400 characters
`DateInvalidException` - is thrown if date is null or is in the past

search

```
public java.util.List search(java.lang.String query)
  throws DatabaseConnectionException
```

Searches the DB for matching toDoItems

Parameters:
`query` - searches database based on query and returns specified information

Returns:
a List of facility objects that contain the search string.

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails

save

```
public boolean save(int id,
         java.lang.String ToDoName,
         java.lang.String ToDoDescription,
         java.lang.String ToDoDate)
  throws NameInvalidException,
         DescriptionInvalidException,
         DatabaseConnectionException,
         DateInvalidException
```

Validates toDoName, toDoDescription to ensure they meet all criteria Checks toDoDate to ensure that it is not null, if it is a exception will be thrown Persist: will persist the supplied item into the DB

Parameters:
`ToDoName` - is the toDoName for the ToDoItem
`ToDoDescription` - is the toDoDescription for the ToDoItem
`ToDoDate` - is the ToDoDate for the ToDoItem

Returns:
true/false if the object is valid.

Throws:

(continued from last page)

`NameInvalidException` - is thrown if it is null or has >30 characters
`DescriptionInvalidException` - is thrown if it is null or has >400 characters
`DateInvalidException` - is thrown if date is null or is in the past
`DatabaseConnectionException` - is thrown if DB connection fails

save

```
public boolean save(ToDoItem todo)
  throws NameInvalidException,
         DescriptionInvalidException,
         DatabaseConnectionException,
         DateInvalidException
```

HttpServletRequest and HttpServletResponse checks toDoName and toDoDescription to ensure they meet all criteria, checks toDoDate to ensure it is not null. If toDoDate is null a exception will be thrown. persists the supplied object into the DB

Parameters:
`todo` - holds all information required by the toDo

Returns:
true/false if the object is valid.

Throws:
`NameInvalidException` - is thrown if it is null or has >30 characters
`DescriptionInvalidException` - is thrown if it is null or has >400 characters
`DateInvalidException` - is thrown if date is null or is in the past
`DatabaseConnectionException` - is thrown if DB connection fails

remove

```
public boolean remove(ToDoItem todo)
  throws DatabaseConnectionException
```

Deletes information from toDoItem table where toDoItem_id = toDoItem_getId If id does not exist a exception will be thrown

Parameters:
`todo` - holds all information required by the toDo

Returns:
true/false if the object is valid.

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails

close

```
public void close()
```

Closes the database connection

getAll

```
public java.util.List getAll()
  throws DatabaseConnectionException
```

Gets all information from toDoItem table in database

Returns:

(continued from last page)

all information for toDoItem table in a list format

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

Package
logicTest

logicTest
Class additionalChargeManagerTest

```
java.lang.Object
  |
  +-junit.framework.Assert
      |
      +-junit.framework.TestCase
          |
          +-logicTest.additionalChargeManagerTest
```

All Implemented Interfaces:
junit.framework.Test

public class **additionalChargeManagerTest**
extends junit.framework.TestCase

Constructor Summary

public	additionalChargeManagerTest()

Method Summary

void	testShouldBeAbleToCreateAAdditionalCharge()
void	testShouldBeAbleToDeleteAdditionalCharge()
void	testShouldNotBeAbleToCreateAAdditionalChargeWithInvalidCost()
void	testShouldNotBeAbleToCreateAAdditionalChargeWithInvalidName()

Methods inherited from class `junit.framework.TestCase`

countTestCases, getName, run, run, runBare, setName, toString

Methods inherited from class `junit.framework.Assert`

assertEquals, assertFalse, assertFalse, assertNotNull, assertNotNull, assertNotSame, assertNotSame, assertNull, assertNull, assertSame, assertSame, assertTrue, assertTrue, fail, fail, failNotEquals, failNotSame, failSame

Methods inherited from class `java.lang.Object`

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface `junit.framework.Test`

countTestCases, run

Constructors

additionalChargeManagerTest

```
public additionalChargeManagerTest()
```

Methods

testShouldBeAbleToCreateAAdditionalCharge

```
public void testShouldBeAbleToCreateAAdditionalCharge()
```

testShouldNotBeAbleToCreateAAdditionalChargeWithInvalidName

```
public void testShouldNotBeAbleToCreateAAdditionalChargeWithInvalidName()
```

testShouldNotBeAbleToCreateAAdditionalChargeWithInvalidCost

```
public void testShouldNotBeAbleToCreateAAdditionalChargeWithInvalidCost()
```

testShouldBeAbleToDeleteAdditionalCharge

```
public void testShouldBeAbleToDeleteAdditionalCharge()
```

logicTest
Class AdditionalChargeTest2

```
java.lang.Object
   |
   +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
         |
         +-logicTest.AdditionalChargeTest2
```

All Implemented Interfaces:
java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable

public class **AdditionalChargeTest2**
extends javax.servlet.http.HttpServlet

Servlet implementation class AdditionalChargeTest2

Constructor Summary

public	AdditionalChargeTest2()

Methods inherited from class javax.servlet.http.HttpServlet

service

Methods inherited from class javax.servlet.GenericServlet

destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface javax.servlet.Servlet

destroy, getServletConfig, getServletInfo, init, service

Methods inherited from interface javax.servlet.ServletConfig

getInitParameter, getInitParameterNames, getServletContext, getServletName

Constructors

AdditionalChargeTest2

public **AdditionalChargeTest2**()

See Also:

(continued from last page)

```
javax.servlet.http.HttpServlet()
```

logicTest
Class bookingManagerTest

```
java.lang.Object
   |
   +-junit.framework.Assert
      |
      +-junit.framework.TestCase
         |
         +-logicTest.bookingManagerTest
```

All Implemented Interfaces:
junit.framework.Test

public class **bookingManagerTest**
extends junit.framework.TestCase

Constructor Summary

public	bookingManagerTest()

Method Summary

void	testShouldAddBookingToSchedule()
void	testShouldBeAbleToCreateABooking()
void	testShouldBeAbleToDeleteBooking()
void	testShouldNotBeAbleToCreateABookingWithInvalidBookingName()
void	testShouldNotBeAbleToCreateABookingWithInvalidClient()
void	testShouldNotBeAbleToCreateABookingWithInvalidName()
void	testShouldNotBeCreatedIfEndDateAfterStartDate()
void	testShouldNotToAddBookingToScheduleIfBookingExistsInSpecifiedTimeFrame()

Methods inherited from class `junit.framework.TestCase`

`countTestCases, getName, run, run, runBare, setName, toString`

Methods inherited from class `junit.framework.Assert` `assertEquals, assertFalse, assertFalse, assertNotNull, assertNotNull, assertNotSame, assertNotSame, assertNull, assertNull, assertSame, assertSame, assertTrue, assertTrue, fail, fail, failNotEquals, failNotSame, failSame`

Methods inherited from class `java.lang.Object`

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

Methods inherited from interface `junit.framework.Test`

`countTestCases, run`

Constructors

bookingManagerTest

`public bookingManagerTest()`

Methods

testShouldBeAbleToCreateABooking

`public void testShouldBeAbleToCreateABooking()`

testShouldNotBeAbleToCreateABookingWithInvalidName

`public void testShouldNotBeAbleToCreateABookingWithInvalidName()`

testShouldNotBeAbleToCreateABookingWithInvalidClient

`public void testShouldNotBeAbleToCreateABookingWithInvalidClient()`

testShouldNotBeAbleToCreateABookingWithInvalidBookingName

`public void testShouldNotBeAbleToCreateABookingWithInvalidBookingName()`

testShouldNotBeCreatedIfEndDateAfterStartDate

`public void testShouldNotBeCreatedIfEndDateAfterStartDate()`

testShouldBeAbleToDeleteBooking

```
public void testShouldBeAbleToDeleteBooking()
```

testShouldAddBookingToSchedule

```
public void testShouldAddBookingToSchedule()
```

testShouldNotToAddBookingToScheduleIfBookingExistsInSpecifiedTimeFrame

```
public void testShouldNotToAddBookingToScheduleIfBookingExistsInSpecifiedTimeFrame()
```

logicTest
Class bookingManagerTest2

```
java.lang.Object
  |
  +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
          |
          +-logicTest.bookingManagerTest2
```

All Implemented Interfaces:
java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable

public class **bookingManagerTest2**
extends javax.servlet.http.HttpServlet

Servlet implementation class bookingManagerTest2

Constructor Summary

public	bookingManagerTest2()

Methods inherited from class `javax.servlet.http.HttpServlet`
`service`

Methods inherited from class `javax.servlet.GenericServlet`
`destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service`

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `javax.servlet.Servlet`
`destroy, getServletConfig, getServletInfo, init, service`

Methods inherited from interface `javax.servlet.ServletConfig`
`getInitParameter, getInitParameterNames, getServletContext, getServletName`

Constructors

bookingManagerTest2

`public bookingManagerTest2()`

See Also:

(continued from last page)

```
javax.servlet.http.HttpServlet()
```

logicTest
Class BookingTypeManagerTest

```
java.lang.Object
  |
  +-junit.framework.Assert
      |
      +-junit.framework.TestCase
          |
          +-logicTest.BookingTypeManagerTest
```

All Implemented Interfaces:
junit.framework.Test

public class **BookingTypeManagerTest**
extends junit.framework.TestCase

Constructor Summary

public	BookingTypeManagerTest()

Method Summary

void	testShouldBeAbleToCreateABookingType()
void	testShouldBeAbleToDeleteBookingType()
void	testShouldNotBeAbleToCreateABookingTypeWithInvalidname()
void	testShouldNotBeAbleToCreateABookingTypeWithInvalidSetupTime()
void	testShouldNotBeAbleToCreateABookingTypeWithInvalidTearDownTime()

Methods inherited from class `junit.framework.TestCase`

countTestCases, getName, run, run, runBare, setName, toString

Methods inherited from class `junit.framework.Assert`

assertEquals, assertFalse, assertFalse, assertNotNull, assertNotNull, assertNotSame, assertNotSame, assertNull, assertNull, assertSame, assertSame, assertTrue, assertTrue, fail, fail, failNotEquals, failNotSame, failSame

Methods inherited from class `java.lang.Object`

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface `junit.framework.Test`

```
countTestCases, run
```

Constructors

BookingTypeManagerTest

```
public BookingTypeManagerTest()
```

Methods

testShouldBeAbleToCreateABookingType

```
public void testShouldBeAbleToCreateABookingType()
```

testShouldNotBeAbleToCreateABookingTypeWithInvalidname

```
public void testShouldNotBeAbleToCreateABookingTypeWithInvalidname()
```

testShouldNotBeAbleToCreateABookingTypeWithInvalidSetupTime

```
public void testShouldNotBeAbleToCreateABookingTypeWithInvalidSetupTime()
```

testShouldNotBeAbleToCreateABookingTypeWithInvalidTearDownTime

```
public void testShouldNotBeAbleToCreateABookingTypeWithInvalidTearDownTime()
```

testShouldBeAbleToDeleteBookingType

```
public void testShouldBeAbleToDeleteBookingType()
```

logicTest
Class BookingTypeManagerTest2

```
java.lang.Object
  |
  +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
          |
          +-logicTest.BookingTypeManagerTest2
```

All Implemented Interfaces:
java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable

public class **BookingTypeManagerTest2**
extends javax.servlet.http.HttpServlet

Servlet implementation class BookingTypeManagerTest2

Constructor Summary

public	BookingTypeManagerTest2()

Methods inherited from class `javax.servlet.http.HttpServlet`

`service`

Methods inherited from class `javax.servlet.GenericServlet`

`destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service`

Methods inherited from class `java.lang.Object`

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

Methods inherited from interface `javax.servlet.Servlet`

`destroy, getServletConfig, getServletInfo, init, service`

Methods inherited from interface `javax.servlet.ServletConfig`

`getInitParameter, getInitParameterNames, getServletContext, getServletName`

Constructors

BookingTypeManagerTest2

`public BookingTypeManagerTest2()`

See Also:

(continued from last page)

```
javax.servlet.http.HttpServlet()
```

logicTest
Class ClientManagerTest

```
java.lang.Object
  |
  +-junit.framework.Assert
      |
      +-junit.framework.TestCase
          |
          +-logicTest.ClientManagerTest
```

All Implemented Interfaces:
junit.framework.Test

public class **ClientManagerTest**
extends junit.framework.TestCase

Constructor Summary

public	ClientManagerTest()

Method Summary

void	testShouldBeAbleToCreateAClient()
void	testShouldBeAbleToDeleteClient()
void	testShouldNotBeAbleToCreateAClientWithInvalidAddress()
void	testShouldNotBeAbleToCreateAClientWithInvalidCellPhone()
void	testShouldNotBeAbleToCreateAClientWithInvalidCity()
void	testShouldNotBeAbleToCreateAClientWithInvalidCountry()
void	testShouldNotBeAbleToCreateAClientWithInvalidDisount()
void	testShouldNotBeAbleToCreateAClientWithInvalidEmail()
void	testShouldNotBeAbleToCreateAClientWithInvalidGivenName()
void	testShouldNotBeAbleToCreateAClientWithInvalidHomePhoneNumber()
void	testShouldNotBeAbleToCreateAClientWithInvalidPostalCode()
void	testShouldNotBeAbleToCreateAClientWithInvalidProvince()
void	testShouldNotBeAbleToCreateAClientWithInvalidSurname()

void	testShouldNotBeAbleToCreateAClientWithInvalidWorkPhone()

Methods inherited from class `junit.framework.TestCase`

`countTestCases, getName, run, run, runBare, setName, toString`

Methods inherited from class `junit.framework.Assert`

`assertEquals, assertFalse, assertFalse, assertNotNull, assertNotNull, assertNotSame, assertNotSame, assertNull, assertNull, assertSame, assertSame, assertTrue, assertTrue, fail, fail, failNotEquals, failNotSame, failSame`

Methods inherited from class `java.lang.Object`

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

Methods inherited from interface `junit.framework.Test`

`countTestCases, run`

Constructors

ClientManagerTest

public **ClientManagerTest**()

Methods

testShouldBeAbleToCreateAClient

public void **testShouldBeAbleToCreateAClient**()

testShouldNotBeAbleToCreateAClientWithInvalidPostalCode

public void **testShouldNotBeAbleToCreateAClientWithInvalidPostalCode**()

testShouldNotBeAbleToCreateAClientWithInvalidGivenName

public void **testShouldNotBeAbleToCreateAClientWithInvalidGivenName**()

testShouldNotBeAbleToCreateAClientWithInvalidSurname

public void **testShouldNotBeAbleToCreateAClientWithInvalidSurname**()

(continued from last page)

testShouldNotBeAbleToCreateAClientWithInvalidAddress

```
public void testShouldNotBeAbleToCreateAClientWithInvalidAddress()
```

testShouldNotBeAbleToCreateAClientWithInvalidEmail

```
public void testShouldNotBeAbleToCreateAClientWithInvalidEmail()
```

testShouldNotBeAbleToCreateAClientWithInvalidCity

```
public void testShouldNotBeAbleToCreateAClientWithInvalidCity()
```

testShouldNotBeAbleToCreateAClientWithInvalidCountry

```
public void testShouldNotBeAbleToCreateAClientWithInvalidCountry()
```

testShouldNotBeAbleToCreateAClientWithInvalidProvince

```
public void testShouldNotBeAbleToCreateAClientWithInvalidProvince()
```

testShouldNotBeAbleToCreateAClientWithInvalidDisount

```
public void testShouldNotBeAbleToCreateAClientWithInvalidDisount()
```

testShouldNotBeAbleToCreateAClientWithInvalidHomePhoneNumber

```
public void testShouldNotBeAbleToCreateAClientWithInvalidHomePhoneNumber()
```

testShouldNotBeAbleToCreateAClientWithInvalidWorkPhone

```
public void testShouldNotBeAbleToCreateAClientWithInvalidWorkPhone()
```

testShouldNotBeAbleToCreateAClientWithInvalidCellPhone

```
public void testShouldNotBeAbleToCreateAClientWithInvalidCellPhone()
```

(continued from last page)

testShouldBeAbleToDeleteClient

```
public void testShouldBeAbleToDeleteClient()
```

logicTest
Class ClientManagerTest2

```
java.lang.Object
  |
  +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
          |
          +-logicTest.ClientManagerTest2
```

All Implemented Interfaces:
java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable

public class **ClientManagerTest2**
extends javax.servlet.http.HttpServlet

Servlet implementation class ClientManagerTest2

Constructor Summary

public	ClientManagerTest2()

Methods inherited from class `javax.servlet.http.HttpServlet`

`service`

Methods inherited from class `javax.servlet.GenericServlet`

`destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service`

Methods inherited from class `java.lang.Object`

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

Methods inherited from interface `javax.servlet.Servlet`

`destroy, getServletConfig, getServletInfo, init, service`

Methods inherited from interface `javax.servlet.ServletConfig`

`getInitParameter, getInitParameterNames, getServletContext, getServletName`

Constructors

ClientManagerTest2

`public ClientManagerTest2()`

See Also:

(continued from last page)

```
javax.servlet.http.HttpServlet()
```

logicTest
Class employeeManagerTest

```
java.lang.Object
   |
   +-junit.framework.Assert
      |
      +-junit.framework.TestCase
         |
         +-logicTest.employeeManagerTest
```

All Implemented Interfaces:
junit.framework.Test

public class **employeeManagerTest**
extends junit.framework.TestCase

Constructor Summary

public	employeeManagerTest()

Method Summary

void	testShouldBeAbleToCreateAClient()
void	testShouldBeAbleToDeleteEmployee()
void	testShouldNotBeAbleToCreateAEmployeeWithInvalidEmployeeLevel()
void	testShouldNotBeAbleToCreateAEmployeeWithInvalidGivenName()
void	testShouldNotBeAbleToCreateAEmployeeWithInvalidPassword()
void	testShouldNotBeAbleToCreateAEmployeeWithInvalidSurname()
void	testShouldNotBeAbleToCreateAEmployeeWithInvalidUsername()

Methods inherited from class `junit.framework.TestCase`

countTestCases, getName, run, run, runBare, setName, toString

Methods inherited from class `junit.framework.Assert`

assertEquals, assertFalse, assertFalse, assertNotNull, assertNotNull, assertNotSame, assertNotSame, assertNull, assertNull, assertSame, assertSame, assertTrue, assertTrue, fail, fail, failNotEquals, failNotSame, failSame

Methods inherited from class `java.lang.Object`

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

Methods inherited from interface `junit.framework.Test`
`countTestCases, run`

Constructors

employeeManagerTest

```
public employeeManagerTest()
```

Methods

testShouldBeAbleToCreateAClient

```
public void testShouldBeAbleToCreateAClient()
```

testShouldNotBeAbleToCreateAEmployeeWithInvalidUsername

```
public void testShouldNotBeAbleToCreateAEmployeeWithInvalidUsername()
```

testShouldNotBeAbleToCreateAEmployeeWithInvalidPassword

```
public void testShouldNotBeAbleToCreateAEmployeeWithInvalidPassword()
```

testShouldNotBeAbleToCreateAEmployeeWithInvalidGivenName

```
public void testShouldNotBeAbleToCreateAEmployeeWithInvalidGivenName()
```

testShouldNotBeAbleToCreateAEmployeeWithInvalidSurname

```
public void testShouldNotBeAbleToCreateAEmployeeWithInvalidSurname()
```

testShouldNotBeAbleToCreateAEmployeeWithInvalidEmployeeLevel

```
public void testShouldNotBeAbleToCreateAEmployeeWithInvalidEmployeeLevel()
```

(continued from last page)

testShouldBeAbleToDeleteEmployee

```
public void testShouldBeAbleToDeleteEmployee()
```

logicTest
Class employeeManagerTest2

```
java.lang.Object
  |
  +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
          |
          +-logicTest.employeeManagerTest2
```

All Implemented Interfaces:
java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable

---

public class **employeeManagerTest2**
extends javax.servlet.http.HttpServlet

Servlet implementation class employeeManagerTest2

Constructor Summary

public	employeeManagerTest2()

Methods inherited from class `javax.servlet.http.HttpServlet`

`service`

Methods inherited from class `javax.servlet.GenericServlet`

`destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service`

Methods inherited from class `java.lang.Object`

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

Methods inherited from interface `javax.servlet.Servlet`

`destroy, getServletConfig, getServletInfo, init, service`

Methods inherited from interface `javax.servlet.ServletConfig`

`getInitParameter, getInitParameterNames, getServletContext, getServletName`

Constructors

employeeManagerTest2

`public employeeManagerTest2()`

See Also:

(continued from last page)

```
javax.servlet.http.HttpServlet()
```

logicTest
Class FacilityManagerTest

```
java.lang.Object
  |
  +-junit.framework.Assert
      |
      +-junit.framework.TestCase
          |
          +-logicTest.FacilityManagerTest
```

All Implemented Interfaces:
junit.framework.Test

public class **FacilityManagerTest**
extends junit.framework.TestCase

Constructor Summary

public	FacilityManagerTest()

Method Summary

void	testShouldBeAbleToCreateAFacility()
void	testShouldBeAbleToDeleteFacility()
void	testShouldNotBeAbleToCreateAFacilityWithInvalidBookingInterval()
void	testShouldNotBeAbleToCreateAFacilityWithInvalidBookingTime()
void	testShouldNotBeAbleToCreateAFacilityWithInvalidCloseTime()
void	testShouldNotBeAbleToCreateAFacilityWithInvalidMaxCapacity()
void	testShouldNotBeAbleToCreateAFacilityWithInvalidName()
void	testShouldNotBeAbleToCreateAFacilityWithInvalidOpenTime()
void	testShouldNotBeAbleToCreateAFacilityWithInvalidSetupTime()
void	testShouldNotBeAbleToCreateAFacilityWithInvalidTearDownTime()

Methods inherited from class junit.framework.TestCase

countTestCases, getName, run, run, runBare, setName, toString

Methods inherited from class junit.framework.Assert assertEquals, assertFalse, assertFalse, assertNotNull, assertNotNull, assertNotSame, assertNotSame, assertNull, assertNull, assertSame, assertSame, assertTrue, assertTrue, fail, fail, failNotEquals, failNotSame, failSame

Methods inherited from class `java.lang.Object`

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface `junit.framework.Test`

countTestCases, run

Constructors

FacilityManagerTest

public **FacilityManagerTest**()

Methods

testShouldBeAbleToCreateAFacility

public void **testShouldBeAbleToCreateAFacility**()

testShouldNotBeAbleToCreateAFacilityWithInvalidName

public void **testShouldNotBeAbleToCreateAFacilityWithInvalidName**()

testShouldNotBeAbleToCreateAFacilityWithInvalidOpenTime

public void **testShouldNotBeAbleToCreateAFacilityWithInvalidOpenTime**()

testShouldNotBeAbleToCreateAFacilityWithInvalidCloseTime

public void **testShouldNotBeAbleToCreateAFacilityWithInvalidCloseTime**()

testShouldNotBeAbleToCreateAFacilityWithInvalidSetupTime

public void **testShouldNotBeAbleToCreateAFacilityWithInvalidSetupTime**()

testShouldNotBeAbleToCreateAFacilityWithInvalidTearDownTime

```
public void testShouldNotBeAbleToCreateAFacilityWithInvalidTearDownTime()
```

testShouldNotBeAbleToCreateAFacilityWithInvalidMaxCapacity

```
public void testShouldNotBeAbleToCreateAFacilityWithInvalidMaxCapacity()
```

testShouldNotBeAbleToCreateAFacilityWithInvalidBookingInterval

```
public void testShouldNotBeAbleToCreateAFacilityWithInvalidBookingInterval()
```

testShouldNotBeAbleToCreateAFacilityWithInvalidBookingTime

```
public void testShouldNotBeAbleToCreateAFacilityWithInvalidBookingTime()
```

testShouldBeAbleToDeleteFacility

```
public void testShouldBeAbleToDeleteFacility()
```

logicTest
Class FacilityManagerTest2

```
java.lang.Object
  |
  +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
          |
          +-logicTest.FacilityManagerTest2
```

All Implemented Interfaces:

java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable

---

public class **FacilityManagerTest2**
extends javax.servlet.http.HttpServlet

Servlet implementation class FacilityManagerTest2

Constructor Summary

public	FacilityManagerTest2()

Methods inherited from class javax.servlet.http.HttpServlet
service

Methods inherited from class javax.servlet.GenericServlet
destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service

Methods inherited from class java.lang.Object
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface javax.servlet.Servlet
destroy, getServletConfig, getServletInfo, init, service

Methods inherited from interface javax.servlet.ServletConfig
getInitParameter, getInitParameterNames, getServletContext, getServletName

---

Constructors

FacilityManagerTest2

`public FacilityManagerTest2()`

See Also:

(continued from last page)

```
javax.servlet.http.HttpServlet()
```

logicTest
Class invoiceManagerTest

```
java.lang.Object
  |
  +-junit.framework.Assert
      |
      +-junit.framework.TestCase
          |
          +-logicTest.invoiceManagerTest
```

All Implemented Interfaces:
junit.framework.Test

public class **invoiceManagerTest**
extends junit.framework.TestCase

Constructor Summary

public	invoiceManagerTest()

Method Summary

void	testShouldBeAbleToSaveInoviceInformation()
void	testShouldNotBeAbleToCreateAInvoiceWithInvalidDate()
void	testShouldNotBeAbleToCreateAInvoiceWithInvalidDescription()
void	testShouldNotBeAbleToCreateAInvoiceWithInvalidId()

Methods inherited from class `junit.framework.TestCase`

`countTestCases, getName, run, run, runBare, setName, toString`

Methods inherited from class `junit.framework.Assert`

```
assertEquals, assertEquals, assertEquals, assertEquals, assertEquals, assertEquals,
assertEquals, assertEquals, assertEquals, assertEquals, assertEquals, assertEquals,
assertEquals, assertEquals, assertEquals, assertEquals, assertEquals, assertEquals,
assertEquals, assertEquals, assertFalse, assertFalse, assertNotNull, assertNotNull,
assertNotSame, assertNotSame, assertNull, assertNull, assertSame, assertSame,
assertTrue, assertTrue, fail, fail, failNotEquals, failNotSame, failSame
```

Methods inherited from class `java.lang.Object`

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

Methods inherited from interface `junit.framework.Test`

`countTestCases, run`

Constructors

invoiceManagerTest

public **invoiceManagerTest**()

Methods

testShouldBeAbleToSaveInoviceInformation

public void **testShouldBeAbleToSaveInoviceInformation**()

testShouldNotBeAbleToCreateAInvoiceWithInvalidId

public void **testShouldNotBeAbleToCreateAInvoiceWithInvalidId**()

testShouldNotBeAbleToCreateAInvoiceWithInvalidDate

public void **testShouldNotBeAbleToCreateAInvoiceWithInvalidDate**()

testShouldNotBeAbleToCreateAInvoiceWithInvalidDescription

public void **testShouldNotBeAbleToCreateAInvoiceWithInvalidDescription**()

logicTest
Class InvoiceManagerTest2

```
java.lang.Object
   |
   +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
         |
         +-logicTest.InvoiceManagerTest2
```

All Implemented Interfaces:

java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable

public class **InvoiceManagerTest2**
extends javax.servlet.http.HttpServlet

Servlet implementation class InvoiceManagerTest2

Constructor Summary

public	InvoiceManagerTest2()

Methods inherited from class `javax.servlet.http.HttpServlet`
service

Methods inherited from class `javax.servlet.GenericServlet`
destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service

Methods inherited from class `java.lang.Object`
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface `javax.servlet.Servlet`
destroy, getServletConfig, getServletInfo, init, service

Methods inherited from interface `javax.servlet.ServletConfig`
getInitParameter, getInitParameterNames, getServletContext, getServletName

Constructors

InvoiceManagerTest2

`public InvoiceManagerTest2()`

See Also:

(continued from last page)

```
javax.servlet.http.HttpServlet()
```

logicTest
Class organizationManagerTest

```
java.lang.Object
  |
  +-junit.framework.Assert
      |
      +-junit.framework.TestCase
          |
          +-logicTest.organizationManagerTest
```

All Implemented Interfaces:
junit.framework.Test

public class **organizationManagerTest**
extends junit.framework.TestCase

Constructor Summary

public	organizationManagerTest()

Method Summary

void	testShouldBeAbleToCreateAOrganization()
void	testShouldBeAbleToDeleteOrganization()
void	testShouldNotBeAbleToCreateAOrganizationWithInvalidContact()
void	testShouldNotBeAbleToCreateAOrganizationWithInvalidDescription()
void	testShouldNotBeAbleToCreateAOrganizationWithInvalidDiscount()
void	testShouldNotBeAbleToCreateAOrganizationWithInvalidId()
void	testShouldNotBeAbleToCreateAOrganizationWithInvalidName()

Methods inherited from class `junit.framework.TestCase`

`countTestCases, getName, run, run, runBare, setName, toString`

Methods inherited from class `junit.framework.Assert`

`assertEquals, assertFalse, assertFalse, assertNotNull, assertNotNull, assertNotSame, assertNotSame, assertNull, assertNull, assertSame, assertSame, assertTrue, assertTrue, fail, fail, failNotEquals, failNotSame, failSame`

Methods inherited from class `java.lang.Object`

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

Methods inherited from interface `junit.framework.Test`
`countTestCases, run`

Constructors

organizationManagerTest

```
public organizationManagerTest()
```

Methods

testShouldBeAbleToCreateAOrganization

```
public void testShouldBeAbleToCreateAOrganization()
```

testShouldNotBeAbleToCreateAOrganizationWithInvalidId

```
public void testShouldNotBeAbleToCreateAOrganizationWithInvalidId()
```

testShouldNotBeAbleToCreateAOrganizationWithInvalidName

```
public void testShouldNotBeAbleToCreateAOrganizationWithInvalidName()
```

testShouldNotBeAbleToCreateAOrganizationWithInvalidDiscount

```
public void testShouldNotBeAbleToCreateAOrganizationWithInvalidDiscount()
```

testShouldNotBeAbleToCreateAOrganizationWithInvalidContact

```
public void testShouldNotBeAbleToCreateAOrganizationWithInvalidContact()
```

testShouldNotBeAbleToCreateAOrganizationWithInvalidDescription

```
public void testShouldNotBeAbleToCreateAOrganizationWithInvalidDescription()
```

(continued from last page)

testShouldBeAbleToDeleteOrganization

```
public void testShouldBeAbleToDeleteOrganization()
```

logicTest
Class OrganizationManagerTest2

```
java.lang.Object
  |
  +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
          |
          +-logicTest.OrganizationManagerTest2
```

All Implemented Interfaces:
java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable

public class **OrganizationManagerTest2**
extends javax.servlet.http.HttpServlet

Servlet implementation class OrganizationManagerTest2

Constructor Summary

public	OrganizationManagerTest2()

Methods inherited from class `javax.servlet.http.HttpServlet`

`service`

Methods inherited from class `javax.servlet.GenericServlet`

`destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service`

Methods inherited from class `java.lang.Object`

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

Methods inherited from interface `javax.servlet.Servlet`

`destroy, getServletConfig, getServletInfo, init, service`

Methods inherited from interface `javax.servlet.ServletConfig`

`getInitParameter, getInitParameterNames, getServletContext, getServletName`

Constructors

OrganizationManagerTest2

```
public OrganizationManagerTest2()
```

See Also:

(continued from last page)

```
javax.servlet.http.HttpServlet()
```

logicTest
Class rateManagerTest

```
java.lang.Object
  |
  +-junit.framework.Assert
      |
      +-junit.framework.TestCase
          |
          +-logicTest.rateManagerTest
```

All Implemented Interfaces:
junit.framework.Test

public class **rateManagerTest**
extends junit.framework.TestCase

Constructor Summary

public	rateManagerTest()

Method Summary

void	testShouldBeAbleToCreateARate()
void	testShouldBeAbleToDeleteRate()
void	testShouldNotBeAbleToCreateARateWithInvalidBookingDeposit()
void	testShouldNotBeAbleToCreateARateWithInvalidDamageDeposit()
void	testShouldNotBeAbleToCreateARateWithInvalidDescription()
void	testShouldNotBeAbleToCreateARateWithInvalidEndTime()
void	testShouldNotBeAbleToCreateARateWithInvalidId()
void	testShouldNotBeAbleToCreateARateWithInvalidIsHourly()
void	testShouldNotBeAbleToCreateARateWithInvalidName()
void	testShouldNotBeAbleToCreateARateWithInvalidRate()
void	testShouldNotBeAbleToCreateARateWithInvalidStartTime()

Methods inherited from class `junit.framework.TestCase`

countTestCases, getName, run, run, runBare, setName, toString

Methods inherited from class `junit.framework.Assert`

`assertEquals, assertFalse, assertFalse, assertNotNull, assertNotNull, assertNotSame, assertNotSame, assertNull, assertNull, assertSame, assertSame, assertTrue, assertTrue, fail, fail, failNotEquals, failNotSame, failSame`

Methods inherited from class `java.lang.Object`

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

Methods inherited from interface `junit.framework.Test`

`countTestCases, run`

Constructors

rateManagerTest

`public rateManagerTest()`

Methods

testShouldBeAbleToCreateARate

`public void testShouldBeAbleToCreateARate()`

testShouldNotBeAbleToCreateARateWithInvalidId

`public void testShouldNotBeAbleToCreateARateWithInvalidId()`

testShouldNotBeAbleToCreateARateWithInvalidName

`public void testShouldNotBeAbleToCreateARateWithInvalidName()`

testShouldNotBeAbleToCreateARateWithInvalidDescription

`public void testShouldNotBeAbleToCreateARateWithInvalidDescription()`

testShouldNotBeAbleToCreateARateWithInvalidRate

`public void testShouldNotBeAbleToCreateARateWithInvalidRate()`

testShouldNotBeAbleToCreateARateWithInvalidDamageDeposit

```
public void testShouldNotBeAbleToCreateARateWithInvalidDamageDeposit()
```

testShouldNotBeAbleToCreateARateWithInvalidBookingDeposit

```
public void testShouldNotBeAbleToCreateARateWithInvalidBookingDeposit()
```

testShouldNotBeAbleToCreateARateWithInvalidIsHourly

```
public void testShouldNotBeAbleToCreateARateWithInvalidIsHourly()
```

testShouldNotBeAbleToCreateARateWithInvalidStartTime

```
public void testShouldNotBeAbleToCreateARateWithInvalidStartTime()
```

testShouldNotBeAbleToCreateARateWithInvalidEndTime

```
public void testShouldNotBeAbleToCreateARateWithInvalidEndTime()
```

testShouldBeAbleToDeleteRate

```
public void testShouldBeAbleToDeleteRate()
```

logicTest
Class RateManagerTest2

```
java.lang.Object
  |
  +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
          |
          +-logicTest.RateManagerTest2
```

All Implemented Interfaces:
java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable

---

public class **RateManagerTest2**
extends javax.servlet.http.HttpServlet

Servlet implementation class RateManagerTest2

Constructor Summary

public	RateManagerTest2()

Methods inherited from class `javax.servlet.http.HttpServlet`

`service`

Methods inherited from class `javax.servlet.GenericServlet`

`destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service`

Methods inherited from class `java.lang.Object`

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

Methods inherited from interface `javax.servlet.Servlet`

`destroy, getServletConfig, getServletInfo, init, service`

Methods inherited from interface `javax.servlet.ServletConfig`

`getInitParameter, getInitParameterNames, getServletContext, getServletName`

Constructors

RateManagerTest2

`public **RateManagerTest2**()`

See Also:

(continued from last page)

```
javax.servlet.http.HttpServlet()
```

logicTest
Class toDoItemManagerTest

```
java.lang.Object
  |
  +-junit.framework.Assert
      |
      +-junit.framework.TestCase
          |
          +-logicTest.toDoItemManagerTest
```

All Implemented Interfaces:
junit.framework.Test

---

public class **toDoItemManagerTest**
extends junit.framework.TestCase

Constructor Summary

public	toDoItemManagerTest()

Method Summary

void	testShouldBeAbleToCreateAToDoItem()
void	testShouldBeAbleToDeleteToDoItem()
void	testShouldNotBeAbleToCreateAToDoItemWithInvalidDate()
void	testShouldNotBeAbleToCreateAToDoItemWithInvalidDescription()
void	testShouldNotBeAbleToCreateAToDoItemWithInvalidId()
void	testShouldNotBeAbleToCreateAToDoItemWithInvalidName()

Methods inherited from class `junit.framework.TestCase`

countTestCases, getName, run, run, runBare, setName, toString

Methods inherited from class `junit.framework.Assert`

assertEquals, assertFalse, assertFalse, assertNotNull, assertNotNull, assertNotSame, assertNotSame, assertNull, assertNull, assertSame, assertSame, assertTrue, assertTrue, fail, fail, failNotEquals, failNotSame, failSame

Methods inherited from class `java.lang.Object`

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface `junit.framework.Test`
`countTestCases, run`

Constructors

toDoItemManagerTest

public **toDoItemManagerTest**()

Methods

testShouldBeAbleToCreateAToDoItem

public void **testShouldBeAbleToCreateAToDoItem**()

testShouldNotBeAbleToCreateAToDoItemWithInvalidId

public void **testShouldNotBeAbleToCreateAToDoItemWithInvalidId**()

testShouldNotBeAbleToCreateAToDoItemWithInvalidName

public void **testShouldNotBeAbleToCreateAToDoItemWithInvalidName**()

testShouldNotBeAbleToCreateAToDoItemWithInvalidDescription

public void **testShouldNotBeAbleToCreateAToDoItemWithInvalidDescription**()

testShouldNotBeAbleToCreateAToDoItemWithInvalidDate

public void **testShouldNotBeAbleToCreateAToDoItemWithInvalidDate**()

testShouldBeAbleToDeleteToDoItem

public void **testShouldBeAbleToDeleteToDoItem**()

logicTest
Class ToDoItemManagerTest2

```
java.lang.Object
  |
  +-javax.servlet.GenericServlet
      |
      +-javax.servlet.http.HttpServlet
          |
          +-logicTest.ToDoItemManagerTest2
```

All Implemented Interfaces:
java.io.Serializable, javax.servlet.ServletConfig, javax.servlet.Servlet, java.io.Serializable

public class **ToDoItemManagerTest2**
extends javax.servlet.http.HttpServlet

Servlet implementation class ToDoItemManagerTest2

Constructor Summary

public	ToDoItemManagerTest2()

Methods inherited from class `javax.servlet.http.HttpServlet`

`service`

Methods inherited from class `javax.servlet.GenericServlet`

`destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, init, log, log, service`

Methods inherited from class `java.lang.Object`

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

Methods inherited from interface `javax.servlet.Servlet`

`destroy, getServletConfig, getServletInfo, init, service`

Methods inherited from interface `javax.servlet.ServletConfig`

`getInitParameter, getInitParameterNames, getServletContext, getServletName`

Constructors

ToDoItemManagerTest2

`public ToDoItemManagerTest2()`

See Also:

(continued from last page)

```
javax.servlet.http.HttpServlet()
```

Package
persistence

persistence
Class AdditionalChargeBroker

```
java.lang.Object
  |
  +-persistence.AdditionalChargeBroker
```

public class **AdditionalChargeBroker**
extends java.lang.Object

Method Summary

void	`close()` Closes broker and any open Database connections
`AdditionalCharge`	`getAdditionalChargeInformation(int id)` Selects all data from the additional charge table As long as the additional charge is not empty it creates a new entry in the table with all user input data
`java.util.List`	`getAdditionalChargeList()` Searches Database for id, name and cost in additional_charges
`static AdditionalChargeBroker`	`getBroker()` Gets the broker instance
`boolean`	`persist(java.lang.Object o)` If the id for additional_charge exists it is updated with the new values If the id does not exist, a new additional_charge is created and stored in the additional_charge table in the database
`boolean`	`remove(java.lang.Object o)` Removes the requested additional charge from the database
`java.util.ArrayList`	`search(java.lang.String searchText)` Searches database for requested additional charge

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods

getBroker

`public static AdditionalChargeBroker getBroker()`

Gets the broker instance

Returns:
the current broker instance

Throws:

(continued from last page)

DatabaseConnectionException - - is thrown if Database connection fails

close

`public void close()`

Closes broker and any open Database connections

getAdditionalChargeList

```
public java.util.List getAdditionalChargeList()
  throws DatabaseConnectionException
```

Searches Database for id, name and cost in additional_charges

Returns:

a list of additional charges

Throws:

DatabaseConnectionException - - is thrown if Database connection fails

persist

```
public boolean persist(java.lang.Object o)
  throws DatabaseConnectionException
```

If the id for additional_charge exists it is updated with the new values If the id does not exist, a new additional_charge is created and stored in the additional_charge table in the database

Parameters:

o - - the additional_charge to search for

Returns:

true/false - if object is valid

Throws:

DatabaseConnectionException - - is thrown if Database connection fails

remove

```
public boolean remove(java.lang.Object o)
  throws DatabaseConnectionException
```

Removes the requested additional charge from the database

Parameters:

o - - the additional charge to be removed

Returns:

true/false - if the object is valid

Throws:

DatabaseConnectionException - - is thrown if Database connection fails

getAdditionalChargeInformation

```
public AdditionalCharge getAdditionalChargeInformation(int id)
  throws DatabaseConnectionException
```

(continued from last page)

Selects all data from the additional charge table As long as the additional charge is not empty it creates a new entry in the table with all user input data

Parameters:

`id` - - is the id of the additionalCharge

Returns:

the search result for requested additional charge

Throws:

`DatabaseConnectionException` - - is thrown if Database connection fails

`SQLException` - - is thrown if their is a problem connection to sql

search

```
public java.util.ArrayList search(java.lang.String searchText)
  throws DatabaseConnectionException
```

Searches database for requested additional charge

Parameters:

`searchText` - - the additional_charges that are to being searched for

Returns:

the requested additional charge(s) from database

Throws:

`DatabaseConnectionException` - - is thrown if Database connection fails

persistence
Class BookingBroker

```
java.lang.Object
  |
  +-persistence.BookingBroker
```

public class **BookingBroker**
extends java.lang.Object

Method Summary

`void`	`close()` Closes the database connection
`Booking`	`getBookingInformation(int id)`
`java.util.List`	`getBookingsForClient(Client c, java.util.Date start, java.util.Date end)` Returns all bookings for a client, during a specified time frame
`java.util.List`	`getBookingsForFacility(Facility f, java.util.Date start, java.util.Date end)` Returns all bookings for a facility during a specified time frame
`java.util.List`	`getBookingsForInvoice(int invoice_no)` Returns all bookings that are associated with a particular invoice
`static BookingBroker`	`getBroker()` Gets the broker instance
`boolean`	`persist(java.lang.Object o)` If the id for catering exists in the database the data is updated with the new user input data If the id does not exist an new catering object is created and stored in the database
`boolean`	`remove(java.lang.Object o)` Removes the supplied object from the database
`java.util.List`	`search(java.lang.String searchText, java.util.Date startTime, java.util.Date endTime, Facility facility)` Searches Database for required information and returns it in a list format

Methods inherited from class `java.lang.Object`

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

Methods

getBroker

```
public static BookingBroker getBroker()
```

Gets the broker instance

(continued from last page)

Returns:
The current broker instance

Throws:
`DatabaseConnectionException` - - is thrown if the database connection fails

close

```
public void close()
```

Closes the database connection

persist

```
public boolean persist(java.lang.Object o)
  throws DatabaseConnectionException
```

If the id for catering exists in the database the data is updated with the new user input data If the id does not exist an new catering object is created and stored in the database

Parameters:
`o` - - catering object being searched for

Returns:
true/false - if the object is valid

Throws:
`DatabaseConnectionException` - - is thrown if the connection to the database fails

remove

```
public boolean remove(java.lang.Object o)
  throws DatabaseConnectionException
```

Removes the supplied object from the database

Parameters:
`o` - - is the booking object being removed

Returns:
true/false - if the object is valid

Throws:
`DatabaseConnectionException` - - is thrown if database connection fails

getBookingsForFacility

```
public java.util.List getBookingsForFacility(Facility f,
          java.util.Date start,
          java.util.Date end)
  throws DatabaseConnectionException
```

Returns all bookings for a facility during a specified time frame

Parameters:
`f` - - the facility to get bookings for
`start` - - the date to start looking for bookings (Non-inclusive)
`end` - - the date to stop looking for bookings (Non-inclusive)

(continued from last page)

Returns:

a list containing all bookings between the specified start and end dates for the chosen facility

Throws:

`DatabaseConnectionException` - - is thrown if the database connection fails

getBookingsForInvoice

```
public java.util.List getBookingsForInvoice(int invoice_no)
  throws DatabaseConnectionException
```

Returns all bookings that are associated with a particular invoice

Parameters:

`invoice_no` - - the invoice number of the invoice to find bookings for

Returns:

A list containing all booking related to the requested invoice number

Throws:

`DatabaseConnectionException` - - is thrown if DB connection fails

getBookingsForClient

```
public java.util.List getBookingsForClient(Client c,
         java.util.Date start,
         java.util.Date end)
  throws DatabaseConnectionException
```

Returns all bookings for a client, during a specified time frame

Parameters:

`c` - - the client to find bookings for
`start` - - the date to start looking for bookings (Non-inclusive)
`end` - - the date to stop looking for bookings (Non-inclusive)

Returns:

a list containing all bookings between the specified start and end dates for the chosen client

Throws:

`DatabaseConnectionException` - - is thrown if DB connection fails

search

```
public java.util.List search(java.lang.String searchText,
         java.util.Date startTime,
         java.util.Date endTime,
         Facility facility)
  throws DatabaseConnectionException
```

Searches Database for required information and returns it in a list format

Parameters:

`searchText` - - passes message along to database to search than returns a list

Returns:

a list of all requested information from database

Throws:

`DatabaseConnectionException` - - is thrown if DB connection fails

getBookingInformation

```
public Booking getBookingInformation(int id)
  throws DatabaseConnectionException
```

persistence
Class BookingTypeBroker

```
java.lang.Object
  |
  +-persistence.BookingTypeBroker
```

public class **BookingTypeBroker**
extends java.lang.Object

Method Summary

`void`	`close()` Closes database connection
`BookingType`	`getBookingTypeInformation(int id)` While booking type is not null, the query connects to the database And the try catch adds a new booking type to the table with all required fields.
`java.util.List`	`getBookingTypeList()` Searches Database for required information and returns it in a list format
`static BookingTypeBroker`	`getBroker()` Gets the broker instance
`boolean`	`persist(java.lang.Object o)` If the id for booking_type exists, booking_type is updated with new values.
`boolean`	`remove(java.lang.Object o)` Removes the supplied object from the database
`java.util.List`	`search(java.lang.String searchText)` Searches Database for required information and returns it in a list format

Methods inherited from class `java.lang.Object`

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

Methods

getBroker

```
public static BookingTypeBroker getBroker()
  throws DatabaseConnectionException
```

Gets the broker instance

Returns:
The current broker instance

Throws:
`DatabaseConnectionException` - if connection to database fails

close

```
public void close()
```

Closes database connection

getBookingTypeList

```
public java.util.List getBookingTypeList()
  throws DatabaseConnectionException
```

Searches Database for required information and returns it in a list format

Returns:
a list of all requested information from database

Throws:
`DatabaseConnectionException` - is thrown if database connection fails

persist

```
public boolean persist(java.lang.Object o)
  throws DatabaseConnectionException
```

If the id for booking_type exists, booking_type is updated with new values. If the id does not exist, information is inserted into booking type table

Parameters:
`o` - - booking type object being searched for

Returns:
true/false if object is valid

Throws:
`DatabaseConnectionException` - is thrown if database connection fails

remove

```
public boolean remove(java.lang.Object o)
  throws DatabaseConnectionException
```

Removes the supplied object from the database

Parameters:
`o` - - booking type object to be removed

Returns:
- true/false if the object is valid

Throws:
`DatabaseConnectionException` - is thrown if the database connection fails

getBookingTypeInformation

```
public BookingType getBookingTypeInformation(int id)
  throws DatabaseConnectionException
```

(continued from last page)

While booking type is not null, the query connects to the database And the try catch adds a new booking type to the table with all required fields.

Parameters:

`id` - is the id for the bookingType

Returns:

a Booking Type object with all the information retrieved from the database

Throws:

`DatabaseConnectionException` - is thrown if database connection fails

search

```
public java.util.List search(java.lang.String searchText)
  throws DatabaseConnectionException
```

Searches Database for required information and returns it in a list format

Parameters:

`searchText` - passes message along to database to search than returns a list

Returns:

a list of all requested information from database

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

persistence
Class ClientBroker

```
java.lang.Object
  |
  +-persistence.ClientBroker
```

public class **ClientBroker**
extends java.lang.Object

Method Summary

`void`	`close()` closes the broker
`static ClientBroker`	`getBroker()` Gets the broker instance
`Client`	`getClientInformation(int id)` Performs a set of queries on the database to retrieve all information that related to a Client with the id specified
`java.util.List`	`getClientList()` Gets all client information and the organizations that they are affiliated with from the database
`boolean`	`persist(java.lang.Object o)` If the id for client exists client is updated with new values.
`boolean`	`remove(java.lang.Object o)` Removes the supplied object from the database
`java.util.List`	`search(java.lang.String searchText)` Searches Database for required information and returns it in a list format
`int`	`validateLogin(java.lang.String email, java.lang.String password)`

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods

getBroker

```
public static ClientBroker getBroker()
  throws DatabaseConnectionException
```

Gets the broker instance

Returns:

The current broker instance

(continued from last page)

Throws:
`DatabaseConnectionException` - - is thrown if database connection fails

close

```
public void close()
  throws java.sql.SQLException
```

closes the broker

Throws:
`SQLException` - - if the query is invalid

getClientList

```
public java.util.List getClientList()
  throws DatabaseConnectionException
```

Gets all client information and the organizations that they are affiliated with from the database

Returns:
client information in a list format

Throws:
`DatabaseConnectionException` - - is thrown if the database connection fails

persist

```
public boolean persist(java.lang.Object o)
  throws DatabaseConnectionException
```

If the id for client exists client is updated with new values. If the id does not exist, information is inserted into client table

Returns:
true/false if the object is valid.

Throws:
`DatabaseConnectionException` - - is thrown if the database connection fails

remove

```
public boolean remove(java.lang.Object o)
  throws DatabaseConnectionException
```

Removes the supplied object from the database

Parameters:
`o` - - object to be removed

Returns:
true/false if the object is valid.

Throws:
`DatabaseConnectionException` - - is thrown if the database connection fails

(continued from last page)

getClientInformation

```
public Client getClientInformation(int id)
  throws DatabaseConnectionException
```

Performs a set of queries on the database to retrieve all information that related to a Client with the id specified

Parameters:

`id` - is the id of the Client that information will be retrieved for

Returns:

A Client object that contains all information retrieved from the database

Throws:

`DatabaseConnectionException` - is thrown if the database connection fails

search

```
public java.util.List search(java.lang.String searchText)
  throws DatabaseConnectionException,
         java.sql.SQLException
```

Searches Database for required information and returns it in a list format

Parameters:

`searchText` - passes message along to database to find requested data than returns a list

Returns:

an ArrayList of all requested information from database

Throws:

`DatabaseConnectionException` - is thrown if the database connection fails

`SQLException` - is thrown if the query is invalid

validateLogin

```
public int validateLogin(java.lang.String email,
         java.lang.String password)
  throws DatabaseConnectionException
```

Parameters:

`email` - - the client email used to login

`password` - - the clients password used to login

Returns:

- returns the clients id

Throws:

`DatabaseConnectionException`

persistence
Class EmployeeBroker

```
java.lang.Object
  |
  +-persistence.EmployeeBroker
```

public class **EmployeeBroker**
extends java.lang.Object

Method Summary

void	`close()` Closes Database connection
static `EmployeeBroker`	`getBroker()` Gets the broker instance
`Employee`	`getEmployeeInformation(int id)` Performs a set of queries on the database to retrieve all information that related to a Employee with the id specified
java.util.List	`getEmployeeList()` Gets all employee information in the database
boolean	`persist(java.lang.Object o)` If the id for employee exists employee is updated with new values.
boolean	`remove(java.lang.Object o)` Removes the supplied object from the database
java.util.List	`search(java.lang.String searchText)` Searches Database for required information and returns it in a list format
int	`validateLogin(java.lang.String username, java.lang.String password)`

Methods inherited from class `java.lang.Object`

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

Methods

close

```
public void close()
```

Closes Database connection

getEmployeeList

```
public java.util.List getEmployeeList()
  throws DatabaseConnectionException
```

(continued from last page)

Gets all employee information in the database

Returns:
a list of employees

Throws:
DatabaseConnectionException - is thrown if database connection fails

persist

```
public boolean persist(java.lang.Object o)
  throws DatabaseConnectionException
```

If the id for employee exists employee is updated with new values. If the id does not exist, information is inserted into employee table

Returns:
true/false if the object is valid.

Throws:
DatabaseConnectionException - - is thrown if the database connection fails

remove

```
public boolean remove(java.lang.Object o)
  throws DatabaseConnectionException
```

Removes the supplied object from the database

Parameters:
o - - the object to be removed

Returns:
true/false if the object is valid.

Throws:
DatabaseConnectionException - - is thrown if the database connection fails)

getBroker

```
public static EmployeeBroker getBroker()
  throws DatabaseConnectionException
```

Gets the broker instance

Returns:
The current broker instance

Throws:
DatabaseConnectionException - is thrown if the database connection fails

getEmployeeInformation

```
public Employee getEmployeeInformation(int id)
  throws DatabaseConnectionException
```

Performs a set of queries on the database to retrieve all information that related to a Employee with the id specified

Parameters:

(continued from last page)

`id` - is the id for the employee that the information will be retrieved for

Returns:

an Employee object with all the information retrieved from the database

Throws:

`DatabaseConnectionException` - is thrown if the database connection fails

search

```
public java.util.List search(java.lang.String searchText)
  throws DatabaseConnectionException
```

Searches Database for required information and returns it in a list format

Parameters:

`searchText` - passes message along to database to search than returns a list

Returns:

list of all requested information from database

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

validateLogin

```
public int validateLogin(java.lang.String username,
         java.lang.String password)
  throws DatabaseConnectionException
```

persistence
Class FacilityBroker

```
java.lang.Object
  |
  +-persistence.FacilityBroker
```

public class **FacilityBroker**
extends java.lang.Object

Method Summary

`void`	`close()` Closes Database connection using the singleton method
`static FacilityBroker`	`getBroker()` Gets the broker instance
`Facility`	`getFacilityInformation(int id)` This method retrieves the facility details using the id of the facility.
`java.util.List`	`getFacilityList()` The method retrieves all the facility objects stored in the database.
`boolean`	`persist(java.lang.Object o)` If the id for facility exists facility is updated with new values If the id does not exist information is inserted into facility table The method first casts the object being passed it as a facility, afterwards it checks if the id of the facility is 0 or an actual number.
`boolean`	`remove(java.lang.Object o)` Deletes information from rate table where facility_id = facility.getId If id does not exist a exception will be thrown.
`java.util.List`	`search(java.lang.String searchText)` Searches for facilities containing some form of the search string passed to the method.

Methods inherited from class `java.lang.Object`

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

Methods

persist

```
public boolean persist(java.lang.Object o)
  throws DatabaseConnectionException
```

If the id for facility exists facility is updated with new values If the id does not exist information is inserted into facility table The method first casts the object being passed it as a facility, afterwards it checks if the id of the facility is 0 or an actual number. If the id is a number then the method opens a connection with the mysql database and performs an update on the facility page. Afterwards the rates are updated for the facility. This is done by opening a connection with the database and deleting all the data from the facility_rates where the facility_id is the same. Afterwards the method inserts all the new facility/rate association into the table to associate the charges related to facility. If the id is 0 than the system performs an insert instead of an update into the facility table.

(continued from last page)

Returns:
true/false if the object is valid.

Throws:
`DatabaseConnectionException` - - is thrown if DB connection fails

remove

```
public boolean remove(java.lang.Object o)
  throws DatabaseConnectionException
```

Deletes information from rate table where facility_id = facility.getId If id does not exist a exception will be thrown. This method opens a database connection to the mysql database and deletes all facilities with the same id's as the object passed in.

Returns:
true/false if the object is valid.

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails

close

```
public void close()
```

Closes Database connection using the singleton method

getBroker

```
public static FacilityBroker getBroker()
  throws DatabaseConnectionException
```

Gets the broker instance

Returns:
The current broker instance

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails

getFacilityInformation

```
public Facility getFacilityInformation(int id)
  throws DatabaseConnectionException
```

This method retrieves the facility details using the id of the facility. The method open a connection with the mysql database and queries the database for the facility information based on the facility id. The method queries the rates and additional charges related to the facility and afterwards finds the remaining information from the facility table. From the results the method constructs a facility object and returns it.

Parameters:
`id` - is the id for the facility

Returns:
the resultSet for requested facility

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails

getFacilityList

```
public java.util.List getFacilityList()
  throws DatabaseConnectionException
```

The method retrieves all the facility objects stored in the database. The method creates an ArrayList of facilities. The method open a connection with the mysql database and queries the database for the facility information. For every facility found in the database the following takes place: The method queries the rates and additional charges related to the facility and afterwards finds the remaining information from the facility table. From the results the method constructs a facility object and adds it to the facilities ArrayList. The method returns the facilities ArrayList.

Returns:

updated result set, returns facilities

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

search

```
public java.util.List search(java.lang.String searchText)
  throws DatabaseConnectionException
```

Searches for facilities containing some form of the search string passed to the method. A database connection is created with the mysql database. Afterwards it is queried for all facilities with a name that is similar to the name of the facilities in the database. The ResultSet received is then looped through afterwards and the results are created into generic facility object, meaning they do not contain any rates or additional charges. They are than added to an ArrayList of facilities which is returned later on.

Returns:

information requested in a list format

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

persistence
Class InvoiceBroker

```
java.lang.Object
  |
  +-persistence.InvoiceBroker
```

public class **InvoiceBroker**
extends java.lang.Object

Method Summary

void	`close()` Closes Database connection
double	`getACCost(Invoice invoice)`
java.util.List	`getAllInvoicesForClient(Client client)`
java.util.List	`getAllUnPaidInvoices()`
static InvoiceBroker	`getBroker()` Gets the broker instance.
java.util.Date	`getDueDate(int invoice_no)`
Invoice	`getInvoiceInformation(int id)` Gets all required information and adds it to resultSet
java.util.List	`getInvoicesForClient(Client c)` Gets all required information adds to resultSet
double	`getPaymentOwed(Invoice invoice)`
java.util.List	`getPayments(Invoice invoice)`
double	`getSubtotal(Invoice invoice)`
boolean	`isPaid(Invoice invoice)`
void	`payInvoice(Invoice i, Payment p)`
boolean	`persist(java.lang.Object o)` If id exists information in invoice table is updated If id != exist information is inserted into invoice table This method utilizes a database connection
boolean	`remove(java.lang.Object o)` Deletes information from rate table where invoice_id = invoice.getId If id does not exist a exception will be thrown.

Methods inherited from class `java.lang.Object`

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

Methods

getBroker

```
public static InvoiceBroker getBroker()
  throws DatabaseConnectionException
```

Gets the broker instance. Method used for the implementation of a singleton.

Returns:
The current broker instance

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails

close

```
public void close()
```

Closes Database connection

persist

```
public boolean persist(java.lang.Object o)
  throws DatabaseConnectionException
```

If id exists information in invoice table is updated If id != exist information is inserted into invoice table This method utilizes a database connection

Returns:
true/false if the object is valid.

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails

payInvoice

```
public void payInvoice(Invoice i,
         Payment p)
```

remove

```
public boolean remove(java.lang.Object o)
  throws DatabaseConnectionException
```

Deletes information from rate table where invoice_id = invoice.getId If id does not exist a exception will be thrown.

Returns:
true/false if the object is valid.

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails

getInvoicesForClient

```
public java.util.List getInvoicesForClient(Client c)
  throws DatabaseConnectionException,
         java.sql.SQLException
```

Gets all required information adds to resultSet

Returns:
updated result set, returns invoices for clients where client_id = c.getId

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails
`SQLException` - is thrown if the sql is not working properly

getInvoiceInformation

```
public Invoice getInvoiceInformation(int id)
  throws DatabaseConnectionException,
         java.sql.SQLException
```

Gets all required information and adds it to resultSet

Parameters:
`id` - is the id for the invoice

Returns:
the resultSet for requested invoice

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails
`SQLException` - is thrown if the sql is not working properly

getACCost

```
public double getACCost(Invoice invoice)
  throws DatabaseConnectionException
```

Parameters:
`invoice` - contains all needed information

Returns:
double for the actual cost owed

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails

getSubtotal

```
public double getSubtotal(Invoice invoice)
  throws DatabaseConnectionException
```

(continued from last page)

getPaymentOwed

```
public double getPaymentOwed(Invoice invoice)
  throws DatabaseConnectionException
```

Parameters:

`invoice` - contains all needed information

Returns:

a double for the payment owed that is inputted

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

getPayments

```
public java.util.List getPayments(Invoice invoice)
  throws DatabaseConnectionException
```

Parameters:

`invoice` - contains all needed information

Returns:

all requested payments in a list format

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

getAllUnPaidInvoices

```
public java.util.List getAllUnPaidInvoices()
  throws DatabaseConnectionException,
         java.sql.SQLException
```

Returns:

all unPaidInvoices in a list format

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails
`SQLException` - is thrown if the sql is not working properly

isPaid

```
public boolean isPaid(Invoice invoice)
  throws DatabaseConnectionException
```

Parameters:

`invoice` - contains all needed invoice information

Returns:

boolean for isPaid

(continued from last page)

Throws:

DatabaseConnectionException - is thrown if DB connection fails

getDueDate

```
public java.util.Date getDueDate(int invoice_no)
  throws DatabaseConnectionException
```

getAllInvoicesForClient

```
public java.util.List getAllInvoicesForClient(Client client)
  throws DatabaseConnectionException
```

persistence
Class OrganizationBroker

```
java.lang.Object
  |
  +-persistence.OrganizationBroker
```

public class **OrganizationBroker**
extends java.lang.Object

Method Summary

`void`	`close()` Closes Database connection
`static OrganizationBroker`	`getBroker()` Gets the broker instance
`Organization`	`getOrganizationInformation(int id)` Gets all required information and adds it to resultSet
`java.util.List`	`getOrgList()` Gets needed information from clientTable in database and creates a new organization to add to organization table
`java.util.List`	`getOrgsForClientID(int clientID)` Retrieves a list of all Organizations that the client belongs to
`boolean`	`persist(java.lang.Object o)` If id exists information in organization table is updated If id != exist information is inserted into organization table
`boolean`	`remove(java.lang.Object o)` Deletes information from rate table where organization_id = organization.getId If id does not exist a exception will be thrown.
`java.util.List`	`search(java.lang.String searchText)` Searches Database for required information and returns it in a list format

Methods inherited from class `java.lang.Object`

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

Methods

close

`public void close()`

Closes Database connection

(continued from last page)

getOrgsForClientID

```
public java.util.List getOrgsForClientID(int clientID)
  throws DatabaseConnectionException,
         java.sql.SQLException
```

Retrieves a list of all Organizations that the client belongs to

Parameters:

`clientID` - the id of the Client to find organizations for

Returns:

a List containing all Organizations that the client with the specified id belongs to

Throws:

`DatabaseConnectionException`
`SQLException`

getOrgList

```
public java.util.List getOrgList()
  throws DatabaseConnectionException,
         java.sql.SQLException
```

Gets needed information from clientTable in database and creates a new organization to add to organization table

Returns:

list of requested organizations

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails
`SQLException`

persist

```
public boolean persist(java.lang.Object o)
  throws DatabaseConnectionException
```

If id exists information in organization table is updated If id != exist information is inserted into organization table

Returns:

true/false if the object is valid.

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

remove

```
public boolean remove(java.lang.Object o)
  throws DatabaseConnectionException
```

Deletes information from rate table where organization_id = organization.getId If id does not exist a exception will be thrown.

Returns:

true/false if the object is valid.

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

getBroker

```
public static OrganizationBroker getBroker()
  throws DatabaseConnectionException
```

Gets the broker instance

Returns:
The current broker instance

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails

getOrganizationInformation

```
public Organization getOrganizationInformation(int id)
  throws DatabaseConnectionException,
         java.sql.SQLException
```

Gets all required information and adds it to resultSet

Parameters:
`id` - is the id for the organization

Returns:
An resultSet for requested organization

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails
`SQLException`

search

```
public java.util.List search(java.lang.String searchText)
  throws DatabaseConnectionException,
         java.sql.SQLException
```

Searches Database for required information and returns it in a list format

Parameters:
`searchText` - passes message along to database to search than returns a list

Returns:
list of all requested information from database

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails
`SQLException`

persistence
Class RateBroker

```
java.lang.Object
  |
  +-persistence.RateBroker
```

public class **RateBroker**
extends java.lang.Object

Method Summary

`void`	`close()` Closes Database connection
`static RateBroker`	`getBroker()` Gets the broker instance
`Rate`	`getRateInformation(int id)` Gets all required information and adds it to resultSet
`java.util.List`	`getRateList()` Gets all information from Database for that table
`boolean`	`persist(java.lang.Object o)` If id exists information in rate table is updated If id != exist information is inserted into rate table
`boolean`	`remove(java.lang.Object o)` Deletes information from rate table where rate_id = rate.getId If id does not exist a exception will be thrown.
`java.util.List`	`search(java.lang.String searchText)` Searches Database for required information and returns it in a list format

Methods inherited from class `java.lang.Object`

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

Methods

getBroker

```
public static RateBroker getBroker()
  throws DatabaseConnectionException
```

Gets the broker instance

Returns:
The current broker instance

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails

close

```
public void close()
```

Closes Database connection

getRateList

```
public java.util.List getRateList()
  throws DatabaseConnectionException
```

Gets all information from Database for that table

Returns:
list of requested rates

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails

persist

```
public boolean persist(java.lang.Object o)
  throws DatabaseConnectionException
```

If id exists information in rate table is updated If id != exist information is inserted into rate table

Returns:
true/false if the object is valid.

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails

remove

```
public boolean remove(java.lang.Object o)
  throws DatabaseConnectionException
```

Deletes information from rate table where rate_id = rate.getId If id does not exist a exception will be thrown.

Returns:
true/false if the object is valid.

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails

getRateInformation

```
public Rate getRateInformation(int id)
  throws DatabaseConnectionException
```

Gets all required information and adds it to resultSet

Parameters:
`id` - is the id for the rate

Returns:
An resultSet for requested rate

(continued from last page)

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

search

```
public java.util.List search(java.lang.String searchText)
  throws DatabaseConnectionException
```

Searches Database for required information and returns it in a list format

Parameters:

`searchText` - passes message along to database to search than returns a list

Returns:

list of all requested information from database

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

persistence
Class ToDoItemBroker

```
java.lang.Object
  |
  +-persistence.ToDoItemBroker
```

public class **ToDoItemBroker**
extends java.lang.Object

Method Summary

void	`close()` Closes database connection
static `ToDoItemBroker`	`getBroker()` Gets the broker instance
`ToDoItem`	`getToDoItemInformation(int id)` Gets all required information and adds it to resultSet
java.util.List	`getToDoItemList()` Gets all information from Database for that table
boolean	`persist(java.lang.Object o)` If id exists information in toDoItem table is updated If id != exist information is inserted into toDoItem table
boolean	`remove(java.lang.Object o)` Deletes information from toDoItem table where toDoItem_id = toDoItem.getId If id does not exist a exception will be thrown.
java.util.List	`search(java.lang.String searchText)` Searches Database for required information and returns it in a list format

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods

close

```
public void close()
```

Closes database connection

getToDoItemList

```
public java.util.List getToDoItemList()
  throws DatabaseConnectionException
```

Gets all information from Database for that table

(continued from last page)

Returns:
list of requested toDoItem

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails

persist

```
public boolean persist(java.lang.Object o)
  throws DatabaseConnectionException
```

If id exists information in toDoItem table is updated If id != exist information is inserted into toDoItem table

Returns:
true/false if the object is valid.

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails

remove

```
public boolean remove(java.lang.Object o)
  throws DatabaseConnectionException
```

Deletes information from toDoItem table where toDoItem_id = toDoItem.getId If id does not exist a exception will be thrown.

Returns:
true/false if the object is valid.

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails

getBroker

```
public static ToDoItemBroker getBroker()
  throws DatabaseConnectionException
```

Gets the broker instance

Returns:
The current broker instance

Throws:
`DatabaseConnectionException` - is thrown if DB connection fails

getToDoItemInformation

```
public ToDoItem getToDoItemInformation(int id)
  throws DatabaseConnectionException
```

Gets all required information and adds it to resultSet

Parameters:
`id` - is the id for the toDoItem

Returns:
An resultSet for requested toDoItem

(continued from last page)

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

search

```
public java.util.List search(java.lang.String searchText)
  throws DatabaseConnectionException
```

Searches Database for required information and returns it in a list format

Parameters:

`searchText` - passes message along to database to search than returns a list

Returns:

list of all requested information from database

Throws:

`DatabaseConnectionException` - is thrown if DB connection fails

Package
problemDomain

problemDomain
Class AdditionalCharge

```
java.lang.Object
  |
  +-problemDomain.AdditionalCharge
```

All Implemented Interfaces:
java.io.Serializable

public class **AdditionalCharge**
extends java.lang.Object
implements java.io.Serializable

Constructor Summary

public	`AdditionalCharge()` Initializes AdditionalCharge
public	`AdditionalCharge(int id, java.lang.String name, double cost)`
public	`AdditionalCharge(java.lang.String name, double cost)`

Method Summary

double	`getCost()`
int	`getId()`
java.lang.String	`getName()`
void	`setCost(double cost)` The cost to set
void	`setId(int id)` The id to set
void	`setName(java.lang.String name)` The name to set

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

AdditionalCharge

```
public AdditionalCharge()
```

(continued from last page)

Initializes AdditionalCharge

AdditionalCharge

```
public AdditionalCharge(int id,
                        java.lang.String name,
                        double cost)
```

Parameters:

`id` - - id of additional charges
`name` - - name of additional charge
`cost` - - amount being charged

AdditionalCharge

```
public AdditionalCharge(java.lang.String name,
                        double cost)
```

Parameters:

`name` - - name of additional charge
`cost` - - amount being charged

Methods

getId

```
public int getId()
```

Returns:

the id - id of additional charge

getName

```
public java.lang.String getName()
```

Returns:

the name - name of additional charge

getCost

```
public double getCost()
```

Returns:

the cost - amount of charge

setName

```
public void setName(java.lang.String name)
```

(continued from last page)

The name to set

Parameters:

`name` - - name of additional charge

setCost

```
public void setCost(double cost)
```

The cost to set

Parameters:

`cost` - - cost of additional charge

setId

```
public void setId(int id)
```

The id to set

Parameters:

`id` - - id of additional charge

problemDomain
Class Booking

```
java.lang.Object
  |
  +-problemDomain.Booking
```

All Implemented Interfaces:
java.io.Serializable

---

public class **Booking**
extends java.lang.Object
implements java.io.Serializable

Constructor Summary

public	Booking(int id, java.lang.String eventTitle, BookingType eventType, java.util.Date startTime, java.util.Date endTime, int setupTime, int tearDownTime, Client client, Catering catering, Employee creator, int numberOfPeople, Rate rate, java.util.ArrayList additionalCharges, Facility facility, int invoice_no)
public	Booking(int id, java.lang.String eventTitle, java.lang.String clientName, java.util.Date startTime, java.util.Date endTime)
public	Booking(java.lang.String eventTitle, BookingType eventType, java.util.Date startTime, java.util.Date endTime, int setupTime, int tearDownTime, Client client, Employee creator, int numberOfPeople, Rate rate, java.util.ArrayList additionalCharges, Facility facility, int invoice_no)
public	Booking() Initializes booking

Method Summary

java.util.ArrayList	getAdditionalCharges()
Catering	getCatering()
Client	getClient()
Employee	getCreator()
java.util.Date	getEndTime()
java.lang.String	getEventTitle()
BookingType	getEventType()
Facility	getFacility()

int	getId()
int	getInvoice_no()
int	getLength()
int	getNumberOfPeople()
Rate	getRate()
int	getSetupTime() The setupTime to set
java.util.Date	getStartTime()
int	getTearDownTime()
void	setAdditionalCharges(java.util.ArrayList additionalCharges) The additionalCharges to set
void	setCatering(Catering catering) The catering to set
void	setClient(Client client) The client to set
void	setCreator(Employee creator) The creator to set
void	setEndTime(java.util.Date endTime) The endTime to set
void	setEventTitle(java.lang.String eventTitle) The eventTitle to set
void	setEventType(BookingType eventType) The eventType to set
void	setFacility(Facility facility) The facility to set
void	setId(int id) The id to set
void	setInvoice_no(int invoiceNo) The invoice_no to set
void	setNumberOfPeople(int numberOfPeople) The numberOfPeople to set
void	setRate(Rate rate) The rate to set
void	setSetupTime(int setupTime) The setupTime to set

void	`setStartTime(java.util.Date startTime)` The startTime to set
void	`setTearDownTime(int tearDownTime)` The tearDownTime to set

Methods inherited from class `java.lang.Object`

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

Constructors

Booking

```
public Booking(int id,
               java.lang.String eventTitle,
               BookingType eventType,
               java.util.Date startTime,
               java.util.Date endTime,
               int setupTime,
               int tearDownTime,
               Client client,
               Catering catering,
               Employee creator,
               int numberOfPeople,
               Rate rate,
               java.util.ArrayList additionalCharges,
               Facility facility,
               int invoice_no)
```

Parameters:

`id` - - id of booking
`eventTitle` - - name of event
`eventType` - - what type of event it is
`startTime` - - time event starts at
`endTime` - - time event ends at
`setupTime` - - time needed to setup the facility for event
`tearDownTime` - - time needed to clean up after event
`client` - - name of client who is booking event
`catering` - - catering orders
`creator` - - name of employee who creates event
`numberOfPeople` - - number of people attending
`rate` - - rate being charged for booking of facility for the event
`additionalCharges` - - charges for additional extras

Booking

```
public Booking(int id,
               java.lang.String eventTitle,
               java.lang.String clientName,
               java.util.Date startTime,
               java.util.Date endTime)
```

Parameters:

`id` - the id of the booking

(continued from last page)

`eventTitle` - the title of the booking
`clientName` - the name of the client
`startTime` - the time the booking starts at
`endTime` - the time the booking ends at

Booking

```
public Booking(java.lang.String eventTitle,
               BookingType eventType,
               java.util.Date startTime,
               java.util.Date endTime,
               int setupTime,
               int tearDownTime,
               Client client,
               Employee creator,
               int numberOfPeople,
               Rate rate,
               java.util.ArrayList additionalCharges,
               Facility facility,
               int invoice_no)
```

Parameters:

`eventTitle` - - name of the event
`eventType` - - type of event being booked
`startTime` - - time the event starts
`endTime` - - time the event ends
`setupTime` - - time needed to setup facility before event
`tearDownTime` - - time needed to clean up facility after event
`client` - - client booking event
`creator` - - employee making booking
`rate` - - amount being charged for facility
`additionalCharges` - - amount due for any additional extras added

Booking

```
public Booking()
```

Initializes booking

Methods

getInvoice_no

```
public int getInvoice_no()
```

Returns:

the invoice_no

setInvoice_no

```
public void setInvoice_no(int invoiceNo)
```

The invoice_no to set

Parameters:

`invoiceNo` - - invoiceNo for booking

getFacility

`public Facility getFacility()`

Returns:
the facility - which is being booked

setFacility

`public void setFacility(Facility facility)`

The facility to set

Parameters:
`facility` - - facility of booking

getId

`public int getId()`

Returns:
the id - of the booking

setId

`public void setId(int id)`

The id to set

Parameters:
`id` - - id of booking

getEventTitle

`public java.lang.String getEventTitle()`

Returns:
the eventTitle - name of the booking

getEventType

`public BookingType getEventType()`

Returns:
the eventType - type of booking

getStartTime

`public java.util.Date getStartTime()`

(continued from last page)

Returns:
the startTime - time event starts

getEndTime

```
public java.util.Date getEndTime()
```

Returns:
the endTime - time event ends

getSetupTime

```
public int getSetupTime()
```

The setupTime to set

Returns:
the setupTime - time needed to setup facility before event

getTearDownTime

```
public int getTearDownTime()
```

Returns:
the tearDownTime - time needed to clean up facility after event

getClient

```
public Client getClient()
```

Returns:
the client - client making booking

getCatering

```
public Catering getCatering()
```

Returns:
the catering - catering orders

getCreator

```
public Employee getCreator()
```

Returns:
the creator - employee making the booking

getNumberOfPeople

`public int getNumberOfPeople()`

Returns:
the numberOfPeople - number of people attending event

getRate

`public Rate getRate()`

Returns:
the rate - amount being charged for renting facility

getAdditionalCharges

`public java.util.ArrayList getAdditionalCharges()`

Returns:
the additionalCharges - extra charges incurred

setEventTitle

`public void setEventTitle(java.lang.String eventTitle)`

The eventTitle to set

Parameters:
`eventTitle` - - eventTitle of booking

setEventType

`public void setEventType(BookingType eventType)`

The eventType to set

Parameters:
`eventType` - - type of event

setStartTime

`public void setStartTime(java.util.Date startTime)`

The startTime to set

Parameters:
`startTime` - - time event starts

setEndTime

`public void setEndTime(java.util.Date endTime)`

(continued from last page)

The endTime to set

Parameters:
`endTime` - - time event ends

setSetupTime

```
public void setSetupTime(int setupTime)
```

The setupTime to set

Parameters:
`setupTime` - - time needed to setup facility

setTearDownTime

```
public void setTearDownTime(int tearDownTime)
```

The tearDownTime to set

Parameters:
`tearDownTime` - - time needed to clean up facility

setClient

```
public void setClient(Client client)
```

The client to set

Parameters:
`client` - - client making booking

setCatering

```
public void setCatering(Catering catering)
```

The catering to set

Parameters:
`catering` - - catering orders

setCreator

```
public void setCreator(Employee creator)
```

The creator to set

Parameters:
`creator` - - employee making booking

setNumberOfPeople

```
public void setNumberOfPeople(int numberOfPeople)
```

The numberOfPeople to set

Parameters:
`numberOfPeople` - - number of people at event

setRate

```
public void setRate(Rate rate)
```

The rate to set

Parameters:

`rate` - - amount being charged for facility

setAdditionalCharges

```
public void setAdditionalCharges(java.util.ArrayList additionalCharges)
```

The additionalCharges to set

Parameters:

`additionalCharges` - - extra charges incurred

getLength

```
public int getLength()
```

Returns:

length - length of time booked

problemDomain
Class BookingType

```
java.lang.Object
  |
  +-problemDomain.BookingType
```

All Implemented Interfaces:
java.io.Serializable

public class **BookingType**
extends java.lang.Object
implements java.io.Serializable

Constructor Summary

public	BookingType()
public	BookingType(int id, java.lang.String name, int setupTime, int tearDownTime)
public	BookingType(java.lang.String name, int setupTime, int tearDownTime)

Method Summary

int	getId()
java.lang.String	getName()
int	getSetupTime()
int	getTearDownTime()
void	setId(int id) The id to set
void	setName(java.lang.String name) The name to set
void	setSetupTime(int setupTime) The setupTime to set
void	setTearDownTime(int tearDownTime) The tearDownTime to set

Methods inherited from class `java.lang.Object`

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

(continued from last page)

Constructors

BookingType

```
public BookingType()
```

BookingType

```
public BookingType(int id,
                   java.lang.String name,
                   int setupTime,
                   int tearDownTime)
```

Parameters:

`id` - - id of booking
`name` - - name of booking
`setupTime` - - amount of time for setup
`tearDownTime` - - amount of time for tear down

BookingType

```
public BookingType(java.lang.String name,
                   int setupTime,
                   int tearDownTime)
```

Parameters:

`name` - - name of booking
`setupTime` - - amount of time for setup
`tearDownTime` - - amount of time for tear down

Methods

getId

```
public int getId()
```

Returns:

the id - id of booking

setId

```
public void setId(int id)
```

The id to set

Parameters:

`id` - - id of booking

(continued from last page)

getName

`public java.lang.String **getName**()`

Returns:
the name - name of booking

getSetupTime

`public int **getSetupTime**()`

Returns:
the setupTime - amount of time for setup

getTearDownTime

`public int **getTearDownTime**()`

Returns:
the tearDownTime - amount of time for tear down

setName

`public void **setName**(java.lang.String name)`

The name to set

Parameters:
`name` - - name of booking

setSetupTime

`public void **setSetupTime**(int setupTime)`

The setupTime to set

Parameters:
`setupTime` - -amount of time for setup

setTearDownTime

`public void **setTearDownTime**(int tearDownTime)`

The tearDownTime to set

Parameters:
`tearDownTime` - - amount of time for tear down

problemDomain
Class Catering

```
java.lang.Object
  |
  +-problemDomain.Catering
```

All Implemented Interfaces:
java.io.Serializable

public class **Catering**
extends java.lang.Object
implements java.io.Serializable

Constructor Summary

public	Catering()
public	Catering(int id, java.lang.String chargeName, double charge, java.lang.String description)
public	Catering(java.lang.String chargeName, double charge, java.lang.String description)

Method Summary

double	getCharge()
java.lang.String	getChargeName()
java.lang.String	getDescription()
int	getId()
void	setCharge(double charge) The charge to set
void	setChargeName(java.lang.String chargeName) The chargeName to set
void	setDescription(java.lang.String description)
void	setId(int id) The id to set

Methods inherited from class `java.lang.Object`

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

(continued from last page)

Constructors

Catering

```
public Catering()
```

Catering

```
public Catering(int id,
                java.lang.String chargeName,
                double charge,
                java.lang.String description)
```

Parameters:

`id` - - catering id
`chargeName` - - a quick reference for searching purposes
`charge` - - amount being charged
`description` - - description of catering order

Catering

```
public Catering(java.lang.String chargeName,
                double charge,
                java.lang.String description)
```

Parameters:

`chargeName` - - a quick reference for searching purposes
`charge` - - amount being charged
`description` - - description of catering order

Methods

getChargeName

```
public java.lang.String getChargeName()
```

Returns:

the chargeName - quick reference for searching purposes

getCharge

```
public double getCharge()
```

Returns:

the charge - amount being charged

(continued from last page)

setChargeName

```
public void setChargeName(java.lang.String chargeName)
```

The chargeName to set

Parameters:
`chargeName` - - quick reference for searching purposes

setCharge

```
public void setCharge(double charge)
```

The charge to set

Parameters:
`charge` - - amount being charged

getDescription

```
public java.lang.String getDescription()
```

Returns:
description of catering

setDescription

```
public void setDescription(java.lang.String description)
```

Parameters:
`description` - of catering

getId

```
public int getId()
```

Returns:
the id - id of charge

setId

```
public void setId(int id)
```

The id to set

Parameters:
`id` - - id of charge

problemDomain
Class Client

```
java.lang.Object
  |
  +-problemDomain.Client
```

All Implemented Interfaces:

java.io.Serializable

public class **Client**
extends java.lang.Object
implements java.io.Serializable

Constructor Summary

public	`Client()` Initializes client
public	`Client(int id)`
public	`Client(int id, java.lang.String givenName, java.lang.String surname, java.lang.String email, java.lang.String workPhone, java.lang.String homePhone, java.lang.String cellPhone)`
public	`Client(java.lang.String givenName, java.lang.String surname, java.lang.String email, java.lang.String address, java.lang.String city, java.lang.String province, java.lang.String country, java.lang.String postalCode, int discount, java.lang.String password, java.lang.String homePhone, java.lang.String workPhone, java.lang.String cellPhone, java.util.ArrayList organizations)`
public	`Client(int id, java.lang.String givenName, java.lang.String surname, java.lang.String email, java.lang.String address, java.lang.String city, java.lang.String province, java.lang.String country, java.lang.String postalCode, int discount, java.lang.String password, java.lang.String homePhone, java.lang.String workPhone, java.lang.String cellPhone, java.util.ArrayList organizations)`

Method Summary

java.lang.String	`getAddress()`
java.lang.String	`getCellPhone()`
java.lang.String	`getCity()`
java.lang.String	`getCountry()`
int	`getDiscount()`
java.lang.String	`getEmail()`

java.lang.String	getGivenName()
java.lang.String	getHomePhone()
int	getId()
java.util.ArrayList	getOrganization() Organizations to get
java.lang.String	getPassword()
java.lang.String	getPostalCode()
java.lang.String	getProvince()
java.lang.String	getSurname()
java.lang.String	getWorkPhone()
void	setAddress(java.lang.String address) The address to set
void	setCellPhone(java.lang.String cellPhone) The cellPhone to set
void	setCity(java.lang.String city) The city to set
void	setCountry(java.lang.String country) The country to set
void	setDiscount(int discount) The discount to set
void	setEmail(java.lang.String email) The email to set
void	setGivenName(java.lang.String givenName) The givenName to set
void	setHomePhone(java.lang.String homePhone) The homePhone to set
void	setId(int id) The id to set
void	setOrganizations(java.util.ArrayList organizations) The organizations to set
void	setPassword(java.lang.String password) The password to set
void	setPostalCode(java.lang.String postalCode) The postalCode to set

`void`	`setProvince(java.lang.String province)` The province to set
`void`	`setSurname(java.lang.String surname)` The surname to set
`void`	`setWorkPhone(java.lang.String workPhone)` The workPhone to set

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

Client

```
public Client()
```

Initializes client

Client

```
public Client(int id)
```

Parameters:

`id`

Client

```
public Client(int id,
              java.lang.String givenName,
              java.lang.String surname,
              java.lang.String email,
              java.lang.String workPhone,
              java.lang.String homePhone,
              java.lang.String cellPhone)
```

Parameters:

`id` - - client id

`givenName` - - client given name

`surname` - - client surname

`email` - - client email

`workPhone` - - client work phone number

`homePhone` - - client home phone number

`cellPhone` - - client cell phone number

(continued from last page)

Client

```
public Client(java.lang.String givenName,
              java.lang.String surname,
              java.lang.String email,
              java.lang.String address,
              java.lang.String city,
              java.lang.String province,
              java.lang.String country,
              java.lang.String postalCode,
              int discount,
              java.lang.String password,
              java.lang.String homePhone,
              java.lang.String workPhone,
              java.lang.String cellPhone,
              java.util.ArrayList organizations)
```

Parameters:

`givenName` - - client given name
`surname` - - client surname
`email` - - client email
`address` - - client address
`city` - - client city
`province` - - client province
`country` - - client country
`postalCode` - - client postalCode
`discount` - - discount been given to client
`password` - - client password for online access
`homePhone` - - client home phone
`workPhone` - - client work phone
`cellPhone` - - client cell phone
`organizations` - - client organizations

Client

```
public Client(int id,
              java.lang.String givenName,
              java.lang.String surname,
              java.lang.String email,
              java.lang.String address,
              java.lang.String city,
              java.lang.String province,
              java.lang.String country,
              java.lang.String postalCode,
              int discount,
              java.lang.String password,
              java.lang.String homePhone,
              java.lang.String workPhone,
              java.lang.String cellPhone,
              java.util.ArrayList organizations)
```

Parameters:

`id` - - client id
`givenName` - - client given name
`surname` - - client surname
`email` - - client email
`address` - - client address
`city` - - client city
`province` - - client province

(continued from last page)

`country` - - client country
`postalCode` - - client postalCode
`discount` - - discount been given to client
`password` - - client password for online access
`homePhone` - - client home phone
`workPhone` - - client work phone
`cellPhone` - - client cell phone
`organizations` - - client organizations

Methods

getId

```
public int getId()
```

Returns:
the id - client id

getGivenName

```
public java.lang.String getGivenName()
```

Returns:
the givenName - client given name

getSurname

```
public java.lang.String getSurname()
```

Returns:
the surname - client surname

getEmail

```
public java.lang.String getEmail()
```

Returns:
the email - client email

getAddress

```
public java.lang.String getAddress()
```

Returns:
the address - client address

(continued from last page)

getCity

```
public java.lang.String getCity()
```

Returns:
the city - client city

getProvince

```
public java.lang.String getProvince()
```

Returns:
the province - client province

getCountry

```
public java.lang.String getCountry()
```

Returns:
the country - client country

getPostalCode

```
public java.lang.String getPostalCode()
```

Returns:
the postalCode - client postal code

getDiscount

```
public int getDiscount()
```

Returns:
the discount - discount given to client

getPassword

```
public java.lang.String getPassword()
```

Returns:
the password - clients online password

getHomePhone

```
public java.lang.String getHomePhone()
```

(continued from last page)

Returns:
the homePhone - clients home phone

getWorkPhone

`public java.lang.String getWorkPhone()`

Returns:
the workPhone - clients work phone

getCellPhone

`public java.lang.String getCellPhone()`

Returns:
the cellPhone - client cell phones

getOrganization

`public java.util.ArrayList getOrganization()`

Organizations to get

Returns:
the Clients Organizations

setId

`public void setId(int id)`

The id to set

Parameters:
`id` - - id of client

setGivenName

`public void setGivenName(java.lang.String givenName)`

The givenName to set

Parameters:
`givenName` - - given name of client

setSurname

`public void setSurname(java.lang.String surname)`

The surname to set

Parameters:
`surname` - - surname of client

setEmail

```
public void setEmail(java.lang.String email)
```

The email to set

Parameters:
`email` - - email of client

setAddress

```
public void setAddress(java.lang.String address)
```

The address to set

Parameters:
`address` - - address of client

setCity

```
public void setCity(java.lang.String city)
```

The city to set

Parameters:
`city` - - city of client

setProvince

```
public void setProvince(java.lang.String province)
```

The province to set

Parameters:
`province` - - province of client

setCountry

```
public void setCountry(java.lang.String country)
```

The country to set

Parameters:
`country` - - country of client

setPostalCode

```
public void setPostalCode(java.lang.String postalCode)
```

The postalCode to set

Parameters:
`postalCode` - - postalcode of client

setDiscount

```
public void setDiscount(int discount)
```

(continued from last page)

The discount to set

Parameters:
`discount` - - discount given to client

setPassword

`public void **setPassword**(java.lang.String password)`

The password to set

Parameters:
`password` - - online password for client

setHomePhone

`public void **setHomePhone**(java.lang.String homePhone)`

The homePhone to set

Parameters:
`homePhone` - - home phone of client

setWorkPhone

`public void **setWorkPhone**(java.lang.String workPhone)`

The workPhone to set

Parameters:
`workPhone` - - work phone of client

setCellPhone

`public void **setCellPhone**(java.lang.String cellPhone)`

The cellPhone to set

Parameters:
`cellPhone` - - cell phone of client

setOrganizations

`public void **setOrganizations**(java.util.ArrayList organizations)`

The organizations to set

Parameters:
`organizations` - - organization of client

problemDomain
Class Employee

```
java.lang.Object
  |
  +-problemDomain.Employee
```

All Implemented Interfaces:

java.io.Serializable

public class **Employee**
extends java.lang.Object
implements java.io.Serializable

Constructor Summary

public	`Employee()`
public	`Employee(int id, java.lang.String username, java.lang.String password, java.lang.String givenName, java.lang.String surname, int employeeLevel)`
public	`Employee(int id, java.lang.String username, java.lang.String givenName, java.lang.String surname, int employeeLevel)`
public	`Employee(java.lang.String username, java.lang.String password, java.lang.String givenName, java.lang.String surname, int employeeLevel)`

Method Summary

int	`getEmployeeLevel()` The employeeLevel to get
java.lang.String	`getGivenName()`
int	`getId()`
java.lang.String	`getPassword()`
java.lang.String	`getSurname()`
java.lang.String	`getUsername()`
void	`setEmployeeLevel(int employeeLevel)` The employeeLevel to set
void	`setGivenName(java.lang.String givenName)` The givenName to set

`void`	`setId(int id)` The id to set
`void`	`setPassword(java.lang.String password)` The password to set
`void`	`setSurname(java.lang.String surname)` The surname to set
`void`	`setUsername(java.lang.String username)` The username to set

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

Employee

```
public Employee()
```

Employee

```
public Employee(int id,
                java.lang.String username,
                java.lang.String password,
                java.lang.String givenName,
                java.lang.String surname,
                int employeeLevel)
```

Parameters:

`id` - - employee id
`username` - - employee login username
`password` - - employee login password
`givenName` - - employee given name
`surname` - - employee surname
`employeeLevel` - - employee level, the amount of access they have to system

Employee

```
public Employee(int id,
                java.lang.String username,
                java.lang.String givenName,
                java.lang.String surname,
                int employeeLevel)
```

Parameters:

`id` - - employee id
`username` - - employee login username
`givenName` - - employee login password
`surname` - - employee surname

(continued from last page)

`employeeLevel` - - employee level, the amount of access they have to system

Employee

```
public Employee(java.lang.String username,
                java.lang.String password,
                java.lang.String givenName,
                java.lang.String surname,
                int employeeLevel)
```

Parameters:

`username` - - employee login username
`password` - - employee login password
`givenName` - - employee given name
`surname` - - employee surname
`employeeLevel` - - employee level, the amount of access they have to system

Methods

getId

```
public int getId()
```

Returns:

the id - employee id

getUsername

```
public java.lang.String getUsername()
```

Returns:

the username - employee login username

getPassword

```
public java.lang.String getPassword()
```

Returns:

the password - employee login password

getGivenName

```
public java.lang.String getGivenName()
```

Returns:

the givenName - employee given name

(continued from last page)

getSurname

```
public java.lang.String getSurname()
```

Returns:
the surname - employee surname

setId

```
public void setId(int id)
```

The id to set

Parameters:
`id` - - employee id

setUsername

```
public void setUsername(java.lang.String username)
```

The username to set

Parameters:
`username` - - employee login username

setPassword

```
public void setPassword(java.lang.String password)
```

The password to set

Parameters:
`password` - - employee login password

setGivenName

```
public void setGivenName(java.lang.String givenName)
```

The givenName to set

Parameters:
`givenName` - - employee given name

setSurname

```
public void setSurname(java.lang.String surname)
```

The surname to set

Parameters:
`surname` - - employee surname

getEmployeeLevel

```
public int getEmployeeLevel()
```

(continued from last page)

The employeeLevel to get

Returns:
the employee Level

setEmployeeLevel

```
public void setEmployeeLevel(int employeeLevel)
```

The employeeLevel to set

Parameters:
`employeeLevel` - - employee Level

problemDomain
Class Facility

```
java.lang.Object
  |
  +-problemDomain.Facility
```

All Implemented Interfaces:
java.io.Serializable

---

public class **Facility**
extends java.lang.Object
implements java.io.Serializable

Constructor Summary

`public`	`Facility()` Initializes facility
`public`	`Facility(int id)` Initializes facility
`public`	`Facility(int id, java.lang.String name)`
`public`	`Facility(int id, int openTime, int closeTime, int setupTime, int tearDownTime, java.lang.String name, java.util.ArrayList rates, java.util.ArrayList additionalCharges, int maxCapacity, int minBookingInterval, int minBookingTime)`
`public`	`Facility(int openTime, int closeTime, int setupTime, int tearDownTime, java.lang.String name, java.util.ArrayList rates, java.util.ArrayList additionalCharges, int maxCapacity, int minBookingInterval, int minBookingTime)`

Method Summary

`java.util.ArrayList`	`getAdditionalCharges()`
`int`	`getCloseTime()`
`int`	`getId()`
`int`	`getMaxCapacity()`
`int`	`getMinBookingInterval()`
`int`	`getMinBookingTime()`
`java.lang.String`	`getName()`

int	getOpenTime()
java.util.ArrayList	getRates()
int	getSetupTime()
int	getTearDownTime()
void	setAdditionalCharges(java.util.ArrayList additionalCharges) The additionalCharges to set
void	setCloseTime(int closeTime) The closeTime to set
void	setId(int id) The id to set
void	setMaxCapacity(int maxCapacity) The maxCapacity to set
void	setMinBookingInterval(int minBookingInterval) The minBookingInterval to set
void	setMinBookingTime(int minBookingTime) The minBookingTime to set
void	setName(java.lang.String name) The name to set
void	setOpenTime(int openTime) The openTime to set
void	setRates(java.util.ArrayList rates) The rates to set
void	setSetupTime(int setupTime) The setupTime to set
void	setTearDownTime(int tearDownTime) The tearDownTime to set

Methods inherited from class `java.lang.Object`

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

Constructors

Facility

`public **Facility**()`

Initializes facility

(continued from last page)

Facility

```
public Facility(int id)
```

Initializes facility

Facility

```
public Facility(int id,
                java.lang.String name)
```

Parameters:

`id` - - id of facility

`name` - - name of facility

Facility

```
public Facility(int id,
                int openTime,
                int closeTime,
                int setupTime,
                int tearDownTime,
                java.lang.String name,
                java.util.ArrayList rates,
                java.util.ArrayList additionalCharges,
                int maxCapacity,
                int minBookingInterval,
                int minBookingTime)
```

Parameters:

`id` - - facility id

`openTime` - - time facility is open

`closeTime` - - time facility is closed

`setupTime` - - amount of time needed to setup

`tearDownTime` - - amount of time needed to tear down

`name` - - name of facility

`rates` - - rates applicable to this facility

`additionalCharges` - - any additional charges applicable to that facility

`maxCapacity` - - maximum number of people allowed in facility

`minBookingInterval` - - minimum amount of time between bookings

`minBookingTime` - - minimum amount of time that can be booked for facility

Facility

```
public Facility(int openTime,
                int closeTime,
                int setupTime,
                int tearDownTime,
                java.lang.String name,
                java.util.ArrayList rates,
                java.util.ArrayList additionalCharges,
                int maxCapacity,
                int minBookingInterval,
                int minBookingTime)
```

Parameters:

(continued from last page)

`openTime` - - time facility is open
`closeTime` - - time facility is closed
`setupTime` - - amount of time needed to tear down
`tearDownTime` - - amount of the time needed to tear down
`name` - - name of facility
`rates` - - rates applicable to this facility
`additionalCharges` - - any additional charges applicable to that facility
`maxCapacity` - - maximum number of people allowed in facility
`minBookingInterval` - - minimum amount of time between bookings
`minBookingTime` - - minimum amount of time that can be booked for facility

Methods

getId

```
public int getId()
```

Returns:
the id of the facility

setId

```
public void setId(int id)
```

The id to set

Parameters:
`id` - - id of facility

getOpenTime

```
public int getOpenTime()
```

Returns:
the openTime - open time of the facility

getCloseTime

```
public int getCloseTime()
```

Returns:
the closeTime - close time of the facility

getSetupTime

```
public int getSetupTime()
```

Returns:
the setupTime - amount of setup time needed for facility

getTearDownTime

```
public int getTearDownTime()
```

Returns:
the tearDownTime - amount of time needed to tear down in facility

getName

```
public java.lang.String getName()
```

Returns:
the name of facility

getRates

```
public java.util.ArrayList getRates()
```

Returns:
the rates - rate amounts applicable to the facility

getAdditionalCharges

```
public java.util.ArrayList getAdditionalCharges()
```

Returns:
the additionalCharges - additional charges applicable to facility

getMaxCapacity

```
public int getMaxCapacity()
```

Returns:
the maxCapacity - maximum number of people allowed in facility

getMinBookingInterval

```
public int getMinBookingInterval()
```

Returns:
the minBookingInterval - minimum amount of time between bookings

getMinBookingTime

```
public int getMinBookingTime()
```

(continued from last page)

Returns:
the minBookingTime - minimum amount of time facility can be booked for

setOpenTime

```
public void setOpenTime(int openTime)
```

The openTime to set

Parameters:
`openTime` - - open time of facility

setCloseTime

```
public void setCloseTime(int closeTime)
```

The closeTime to set

Parameters:
`closeTime` - - closing time of facility

setSetupTime

```
public void setSetupTime(int setupTime)
```

The setupTime to set

Parameters:
`setupTime` - - amount of setup time needed for facility

setTearDownTime

```
public void setTearDownTime(int tearDownTime)
```

The tearDownTime to set

Parameters:
`tearDownTime` - - amount of tear down time needed for facility

setName

```
public void setName(java.lang.String name)
```

The name to set

Parameters:
`name` - - name of facility

setRates

```
public void setRates(java.util.ArrayList rates)
```

The rates to set

Parameters:
`rates` - - rates applicable to facility

setAdditionalCharges

`public void **setAdditionalCharges**(java.util.ArrayList additionalCharges)`

The additionalCharges to set

Parameters:
`additionalCharges` - - additional charges applicable to facility

setMaxCapacity

`public void **setMaxCapacity**(int maxCapacity)`

The maxCapacity to set

Parameters:
`maxCapacity` - - maximum number of people allowed in facility

setMinBookingInterval

`public void **setMinBookingInterval**(int minBookingInterval)`

The minBookingInterval to set

Parameters:
`minBookingInterval` - - minimum amount of time between bookings in facility

setMinBookingTime

`public void **setMinBookingTime**(int minBookingTime)`

The minBookingTime to set

Parameters:
`minBookingTime` - - minimum amount of time for a booking in facility

problemDomain
Class Invoice

```
java.lang.Object
  |
  +-problemDomain.Invoice
```

All Implemented Interfaces:
java.io.Serializable

public class **Invoice**
extends java.lang.Object
implements java.io.Serializable

Constructor Summary

public	Invoice() Initializes invoice
public	Invoice(int id)
public	Invoice(int id, java.util.Date date, Client client, java.util.List bookings, java.lang.Boolean paid, double paymentDue, java.lang.String description, java.util.Date dueDate, java.util.List payments)
public	Invoice(int id, java.util.Date date, Client client, java.util.List bookings, java.lang.String description)
public	Invoice(java.util.Date date, Client client, java.util.ArrayList bookings, java.lang.Boolean paid, double paymentDue, java.lang.String description, java.util.Date dueDate, java.util.ArrayList payments)

Method Summary

java.util.ArrayList	getBookings()
Client	getClient()
java.util.Date	getDate()
java.lang.String	getDescription()
java.util.Date	getDueDate()
int	getId()
java.lang.Boolean	getPaid()
double	getPaymentDue()

java.util.ArrayList	getPayments()
double	getSubtotal()
void	setBookings(java.util.ArrayList bookings) The bookings to set
void	setClient(Client client) The client to set
void	setDate(java.util.Date date) The date to set
void	setDescription(java.lang.String description) The description to set
void	setDueDate(java.util.Date dueDate) The dueDate to set
void	setId(int id) The id to set
void	setPaid(java.lang.Boolean paid) The paid to set
void	setPaymentDue(double paymentDue) The paymentDue to set
void	setPayments(java.util.ArrayList payments) The payments to set
void	setSubtotal(double subtotal)

Methods inherited from class `java.lang.Object`

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

Constructors

Invoice

`public **Invoice**()`

Initializes invoice

Invoice

`public **Invoice**(int id)`

(continued from last page)

Invoice

```
public Invoice(int id,
               java.util.Date date,
               Client client,
               java.util.List bookings,
               java.lang.Boolean paid,
               double paymentDue,
               java.lang.String description,
               java.util.Date dueDate,
               java.util.List payments)
```

Parameters:

`id` - - id of invoice
`date` - - invoice date
`client` - - client being invoiced
`bookings` - - bookings included in invoice
`paid` - - invoice has been paid
`paymentDue` - - how much client owes
`description` - - description of service that client is paying for
`dueDate` - - date the payment needs to be paid by
`payments` - - Any payments made on this invoice

Invoice

```
public Invoice(int id,
               java.util.Date date,
               Client client,
               java.util.List bookings,
               java.lang.String description)
```

Parameters:

`id` - - id of invoice
`date` - - invoice date
`client` - - client being invoiced
`bookings` - - bookings included in invoice
`description` - - description of service that client is paying for

Invoice

```
public Invoice(java.util.Date date,
               Client client,
               java.util.ArrayList bookings,
               java.lang.Boolean paid,
               double paymentDue,
               java.lang.String description,
               java.util.Date dueDate,
               java.util.ArrayList payments)
```

Parameters:

`date` - - invoice date
`client` - - client being invoiced
`bookings` - - bookings included in invoice
`paid` - - invoice has been paid
`paymentDue` - - how much client owes
`description` - - description of service that client is paying for

(continued from last page)

`dueDate` - - date the payment needs to be paid by
`payments` - - any payments made on this invoice

Methods

getId

`public int getId()`

Returns:
the id of the invoice

setId

`public void setId(int id)`

The id to set

Parameters:
`id` - - id of invoice

getDate

`public java.util.Date getDate()`

Returns:
the date of the invoice

getClient

`public Client getClient()`

Returns:
the client who is being billed

getBookings

`public java.util.ArrayList getBookings()`

Returns:
the bookings that are being added to the invoice

getPaid

`public java.lang.Boolean getPaid()`

Returns:
the paid - invoice is paid

getPaymentDue

```
public double getPaymentDue()
```

Returns:
the paymentDue - amount that is due

getDescription

```
public java.lang.String getDescription()
```

Returns:
the description - description of service

getDueDate

```
public java.util.Date getDueDate()
```

Returns:
the dueDate - date payment is due by

getPayments

```
public java.util.ArrayList getPayments()
```

Returns:
the paymentDate - date payment is made

setDate

```
public void setDate(java.util.Date date)
```

The date to set

Parameters:
`date` - - date of invoice

setClient

```
public void setClient(Client client)
```

The client to set

Parameters:
`client` - - client being invoiced

setBookings

```
public void setBookings(java.util.ArrayList bookings)
```

(continued from last page)

The bookings to set

Parameters:
`bookings` - - bookings being added to invoice

setPaid

`public void **setPaid**(java.lang.Boolean paid)`

The paid to set

Parameters:
`paid` - - invoice is paid

setPaymentDue

`public void **setPaymentDue**(double paymentDue)`

The paymentDue to set

Parameters:
`paymentDue` - - amount due

setDescription

`public void **setDescription**(java.lang.String description)`

The description to set

Parameters:
`description` - - description of service

setDueDate

`public void **setDueDate**(java.util.Date dueDate)`

The dueDate to set

Parameters:
`dueDate` - - date amount to be paid by

setPayments

`public void **setPayments**(java.util.ArrayList payments)`

The payments to set

Parameters:
`payments` - - set any payments made to the invoice

setSubtotal

`public void **setSubtotal**(double subtotal)`

Parameters:
`subtotal` - the subtotal to set

getSubtotal

```
public double getSubtotal()
```

Returns:

the subtotal

problemDomain
Class Organization

```
java.lang.Object
  |
  +-problemDomain.Organization
```

All Implemented Interfaces:

java.io.Serializable

public class **Organization**
extends java.lang.Object
implements java.io.Serializable

Constructor Summary

`public`	`Organization()` Initializes organization
`public`	`Organization(java.lang.String name, double discount, Client contact, java.lang.String description)`
`public`	`Organization(int id, java.lang.String name, double discount, Client contact, java.lang.String description)`
`public`	`Organization(int id, java.lang.String name, java.lang.String description)`
`public`	`Organization(int id, java.lang.String name)`

Method Summary

`Client`	`getContact()`
`java.lang.String`	`getDescription()`
`double`	`getDiscount()`
`int`	`getId()`
`java.lang.String`	`getName()`
`void`	`setContact(Client contact)` The contact to set
`void`	`setDescription(java.lang.String description)` The description to set
`void`	`setDiscount(double discount)` The discount to set

`void`	`setId(int id)` The id to set
`void`	`setName(java.lang.String name)` The name to set

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

Organization

```
public Organization()
```

Initializes organization

Organization

```
public Organization(java.lang.String name,
                    double discount,
                    Client contact,
                    java.lang.String description)
```

Parameters:

`name` - - name of organization
`discount` - - discount given to organization
`contact` - - contact name in organization
`description` - - description of organization

Organization

```
public Organization(int id,
                    java.lang.String name,
                    double discount,
                    Client contact,
                    java.lang.String description)
```

Parameters:

`id` - - organization id
`name` - - name of the organization
`discount` - - discount given to organization
`contact` - - contact name in organization
`description` - - description of organization

Organization

```
public Organization(int id,
                    java.lang.String name,
                    java.lang.String description)
```

(continued from last page)

Parameters:
`id` - - id of organization
`name` - - name of organization
`description` - - description of organization

Organization

```
public Organization(int id,
                    java.lang.String name)
```

Parameters:
`id` - - id of organization
`name` - - name of organization

Methods

getName

```
public java.lang.String getName()
```

Returns:
the name - name of the organization

getDiscount

```
public double getDiscount()
```

Returns:
the discount - discount given to organization

getContact

```
public Client getContact()
```

Returns:
the contact - contact in organization

getDescription

```
public java.lang.String getDescription()
```

Returns:
the description - description of organization

setName

```
public void setName(java.lang.String name)
```

(continued from last page)

The name to set

Parameters:
`name` - - name of organization

setDiscount

```
public void setDiscount(double discount)
```

The discount to set

Parameters:
`discount` - - discount given to organization

setContact

```
public void setContact(Client contact)
```

The contact to set

Parameters:
`contact` - - contact in organization

setDescription

```
public void setDescription(java.lang.String description)
```

The description to set

Parameters:
`description` - - description of organization

getId

```
public int getId()
```

Returns:
the id of the organization

setId

```
public void setId(int id)
```

The id to set

Parameters:
`id` - - id of organization

problemDomain
Class Payment

```
java.lang.Object
  |
  +-problemDomain.Payment
```

All Implemented Interfaces:
java.io.Serializable

---

public class **Payment**
extends java.lang.Object
implements java.io.Serializable

Constructor Summary

`public`	`Payment()` Initializes payment
`public`	`Payment(int id, java.util.Date date, double amount, java.lang.String type)`
`public`	`Payment(int int1, java.sql.Date date2, double double1)`

Method Summary

`double`	`getAmount()`
`java.util.Date`	`getDate()`
`int`	`getId()`
`java.lang.String`	`getType()`
`void`	`setAmount(double amount)` The amount to set
`void`	`setDate(java.util.Date date)` The date to set
`void`	`setId(int id)` The id to set
`void`	`setType(java.lang.String type)`

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

(continued from last page)

Constructors

Payment

```
public Payment()
```

Initializes payment

Payment

```
public Payment(int id,
               java.util.Date date,
               double amount,
               java.lang.String type)
```

Parameters:

`id` - - payment id

`date` - - date of payment

`amount` - - payment amount

Payment

```
public Payment(int int1,
               java.sql.Date date2,
               double double1)
```

Methods

getId

```
public int getId()
```

Returns:

the id of the payment

getDate

```
public java.util.Date getDate()
```

Returns:

the date of the payment

getAmount

```
public double getAmount()
```

Returns:

the amount of the payment

setId

```
public void setId(int id)
```

The id to set

Parameters:
`id` - - id of payment

setDate

```
public void setDate(java.util.Date date)
```

The date to set

Parameters:
`date` - - date of payment

setAmount

```
public void setAmount(double amount)
```

The amount to set

Parameters:
`amount` - - amount of payment

setType

```
public void setType(java.lang.String type)
```

Parameters:
`type` - the type to set

getType

```
public java.lang.String getType()
```

Returns:
the type

problemDomain
Class Rate

```
java.lang.Object
  |
  +-problemDomain.Rate
```

All Implemented Interfaces:

java.io.Serializable

public class **Rate**
extends java.lang.Object
implements java.io.Serializable

Constructor Summary

public	`Rate()` Initializes rate
public	`Rate(int id, java.lang.String name, java.lang.String description, double rate, double damageDeposit, double bookingDeposit)`
public	`Rate(int id, java.lang.String name, java.lang.String description, double rate, double damageDeposit, double bookingDeposit, boolean isHourly, java.util.Date validStartTime, java.util.Date validEndTime, boolean sunday, boolean monday, boolean tuesday, boolean wednesday, boolean thursday, boolean friday, boolean saturday)`
public	`Rate(int id, java.lang.String name, double rate)`
public	`Rate(int id, java.lang.String name, java.lang.String description, double rate)`
public	`Rate(java.lang.String name, java.lang.String description, double rate, double damageDeposit, double bookingDeposit)`

Method Summary

boolean	`equals(java.lang.Object o)`
double	`getBookingDeposit()`
double	`getDamageDeposit()`
java.lang.String	`getDescription()`
int	`getId()`
java.lang.String	`getName()`

double	getRate()
java.util.Date	getValidEndTime()
java.util.Date	getValidStartTime()
boolean	isFriday()
boolean	isHourly()
boolean	isMonday()
boolean	isSaturday()
boolean	isSunday()
boolean	isThursday()
boolean	isTuesday()
boolean	isWednesday()
void	setBookingDeposit(double bookingDeposit) The bookingDeposit to set
void	setDamageDeposit(double damageDeposit) The damageDeposit to set
void	setDescription(java.lang.String description) The description to set
void	setFriday(boolean friday)
void	setHourly(boolean isHourly)
void	setId(int id) The id to set
void	setMonday(boolean monday)
void	setName(java.lang.String name) The name to set
void	setRate(double rate) The rate to set
void	setSaturday(boolean saturday)
void	setSunday(boolean sunday)
void	setThursday(boolean thursday)

void	setTuesday(boolean tuesday)
void	setValidEndTime(java.util.Date validEndTime)
void	setValidStartTime(java.util.Date validStartTime)
void	setWednesday(boolean wednesday)

Methods inherited from class `java.lang.Object`
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

Rate

```
public Rate()
```

Initializes rate

Rate

```
public Rate(int id,
            java.lang.String name,
            java.lang.String description,
            double rate,
            double damageDeposit,
            double bookingDeposit)
```

Parameters:

`name` - - type of rate
`description` - - description of what the rate entails
`rate` - - rate amount
`damageDeposit` - - damage deposit amount needed
`bookingDeposit` - - booking deposit amount needed

Rate

```
public Rate(int id,
            java.lang.String name,
            java.lang.String description,
            double rate,
            double damageDeposit,
            double bookingDeposit,
            boolean isHourly,
            java.util.Date validStartTime,
            java.util.Date validEndTime,
            boolean sunday,
            boolean monday,
            boolean tuesday,
            boolean wednesday,
            boolean thursday,
            boolean friday,
            boolean saturday)
```

(continued from last page)

Parameters:

`id`
`name`
`description`
`rate`
`damageDeposit`
`bookingDeposit`
`isHourly`
`validStartTime`
`validEndTime`
`sunday`
`monday`
`tuesday`
`wednesday`
`thursday`
`friday`
`saturday`

Rate

```
public Rate(int id,
            java.lang.String name,
            double rate)
```

Parameters:

`id` - - rate id
`name` - - type of rate
`rate` - - rate amount

Rate

```
public Rate(int id,
            java.lang.String name,
            java.lang.String description,
            double rate)
```

Parameters:

`id` - - rate id
`name` - - type of rate
`rate` - - rate amount

Rate

```
public Rate(java.lang.String name,
            java.lang.String description,
            double rate,
            double damageDeposit,
            double bookingDeposit)
```

Parameters:

`name` - - name of rate
`description` - - description of rate

(continued from last page)

`rate` - - rate of rate
`damageDeposit` - - damageDeposit of rate
`bookingDeposit` - - bookingDeposit of rate

Methods

getName

```
public java.lang.String getName()
```

Returns:
the name of the rate

getDescription

```
public java.lang.String getDescription()
```

Returns:
the description of the rate

getRate

```
public double getRate()
```

Returns:
the rate - rate amount

getDamageDeposit

```
public double getDamageDeposit()
```

Returns:
damageDeposit - the amount of the damage deposit

getBookingDeposit

```
public double getBookingDeposit()
```

Returns:
bookingDeposit - the amount of the booking deposit

setName

```
public void setName(java.lang.String name)
```

The name to set

Parameters:

(continued from last page)

`name` - - the name of the rate

setDescription

`public void setDescription(java.lang.String description)`

The description to set

Parameters:
`description` - - description of the rate

setRate

`public void setRate(double rate)`

The rate to set

Parameters:
`rate` - - rate amount

setDamageDeposit

`public void setDamageDeposit(double damageDeposit)`

The damageDeposit to set

Parameters:
`damageDeposit` - - amount of damage deposit

setBookingDeposit

`public void setBookingDeposit(double bookingDeposit)`

The bookingDeposit to set

Parameters:
`bookingDeposit` - - amount of booking deposit

getId

`public int getId()`

Returns:
the id - of the invoice

setId

`public void setId(int id)`

The id to set

Parameters:
`id` - - of the invoice

(continued from last page)

isHourly

```
public boolean isHourly()
```

setHourly

```
public void setHourly(boolean isHourly)
```

getValidStartTime

```
public java.util.Date getValidStartTime()
```

setValidStartTime

```
public void setValidStartTime(java.util.Date validStartTime)
```

getValidEndTime

```
public java.util.Date getValidEndTime()
```

setValidEndTime

```
public void setValidEndTime(java.util.Date validEndTime)
```

isSunday

```
public boolean isSunday()
```

setSunday

```
public void setSunday(boolean sunday)
```

isMonday

```
public boolean isMonday()
```

setMonday

```
public void setMonday(boolean monday)
```

(continued from last page)

isTuesday

```
public boolean isTuesday()
```

setTuesday

```
public void setTuesday(boolean tuesday)
```

isWednesday

```
public boolean isWednesday()
```

setWednesday

```
public void setWednesday(boolean wednesday)
```

isThursday

```
public boolean isThursday()
```

setThursday

```
public void setThursday(boolean thursday)
```

isFriday

```
public boolean isFriday()
```

setFriday

```
public void setFriday(boolean friday)
```

isSaturday

```
public boolean isSaturday()
```

(continued from last page)

setSaturday

```
public void setSaturday(boolean saturday)
```

equals

```
public boolean equals(java.lang.Object o)
```

problemDomain
Class ToDoItem

```
java.lang.Object
  |
  +-problemDomain.ToDoItem
```

All Implemented Interfaces:
java.io.Serializable

public class **ToDoItem**
extends java.lang.Object
implements java.io.Serializable

Constructor Summary

public	`ToDoItem()` Initializes ToDoItem
public	`ToDoItem(java.util.Date date, java.lang.String name, java.lang.String description)`
public	`ToDoItem(int id, java.util.Date date, java.lang.String name, java.lang.String description)`

Method Summary

`java.util.Date`	`getDate()`
`java.lang.String`	`getDescription()`
`int`	`getId()`
`java.lang.String`	`getName()`
`void`	`setDate(java.util.Date date)` The date to set
`void`	`setDescription(java.lang.String description)` The description to set
`void`	`setId(int id)` The id to set
`void`	`setName(java.lang.String name)` The name to set

Methods inherited from class `java.lang.Object`
`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

ToDoItem

```
public ToDoItem()
```

Initializes ToDoItem

ToDoItem

```
public ToDoItem(java.util.Date date,
                java.lang.String name,
                java.lang.String description)
```

Parameters:

`date` - - date that the ToDoItem needs to be done
`name` - - name of the task in the ToDoItem
`description` - - description of what needs to be done in the ToDoItem

ToDoItem

```
public ToDoItem(int id,
                java.util.Date date,
                java.lang.String name,
                java.lang.String description)
```

Parameters:

`date` - - date that the ToDoItem needs to be done
`name` - - name of the task in the ToDoItem
`description` - - description of what needs to be done in the ToDoItem
`id` - - Unique identifier for the ToDoItem

Methods

getId

```
public int getId()
```

Returns:

the id of the ToDoItem

getDate

```
public java.util.Date getDate()
```

Returns:

the date of the ToDoItem

getName

```
public java.lang.String getName()
```

Returns:
the name of the ToDoItem

getDescription

```
public java.lang.String getDescription()
```

Returns:
the description of the task that needs to be done in the ToDoItem

setId

```
public void setId(int id)
```

The id to set

Parameters:
`id` - - id of ToDoItem

setDate

```
public void setDate(java.util.Date date)
```

The date to set

Parameters:
`date` - - date of ToDoItem

setName

```
public void setName(java.lang.String name)
```

The name to set

Parameters:
`name` - - name of ToDoItem

setDescription

```
public void setDescription(java.lang.String description)
```

The description to set

Parameters:
`description` - - description of ToDoItem