

Video_Store

1.0

Generated by Doxygen 1.8.16

1 VideoStore - A video store that rents games and DVDs.	1
1.1 release.notes	1
2 Namespace Index	3
2.1 Namespace List	3
3 Hierarchical Index	5
3.1 Class Hierarchy	5
4 Class Index	7
4.1 Class List	7
5 File Index	9
5.1 File List	9
6 Namespace Documentation	11
6.1 AbstractItem Namespace Reference	11
6.1.1 Detailed Description	11
6.1.2 Function Documentation	11
6.1.2.1 main()	12
6.2 Customer Namespace Reference	12
6.2.1 Detailed Description	12
6.2.2 Function Documentation	12
6.2.2.1 main()	12
6.3 GenreSearch Namespace Reference	13
6.3.1 Detailed Description	13
6.4 Item Namespace Reference	13
6.4.1 Detailed Description	13
6.4.2 Variable Documentation	14
6.4.2.1 ABC	14
6.5 OverDuelItemSearch Namespace Reference	14
6.5.1 Detailed Description	14
6.6 SearchCondition Namespace Reference	14
6.6.1 Detailed Description	15
6.6.2 Variable Documentation	15
6.6.2.1 ABC	15
6.7 SimpleDate Namespace Reference	15
6.7.1 Detailed Description	15
6.7.2 Function Documentation	16
6.7.2.1 main()	16
6.8 StatusException Namespace Reference	16

6.8.1 Detailed Description	16
6.9 VideoStore Namespace Reference	16
6.9.1 Detailed Description	17
6.9.2 Function Documentation	17
6.9.2.1 main()	17
7 Class Documentation	19
7.1 AbstractItem.AbstractItem Class Reference	19
7.1.1 Detailed Description	21
7.1.2 Constructor & Destructor Documentation	21
7.1.2.1 __init__()	21
7.1.3 Member Function Documentation	22
7.1.3.1 __eq__()	22
7.1.3.2 __repr__()	22
7.1.3.3 __str__()	23
7.1.3.4 getBarcode()	23
7.1.3.5 getDueDate()	23
7.1.3.6 getGenre()	24
7.1.3.7 getTitle()	24
7.1.3.8 isRented()	24
7.1.3.9 setDueDate()	24
7.1.3.10 setRented()	25
7.1.3.11 setReturned()	25
7.1.4 Member Data Documentation	26
7.1.4.1 __barcode	26
7.1.4.2 __dueDate	26
7.1.4.3 __genre	26
7.1.4.4 __rented	27
7.1.4.5 __title	27
7.2 Customer.Customer Class Reference	27
7.2.1 Detailed Description	29
7.2.2 Constructor & Destructor Documentation	29
7.2.2.1 __init__()	29
7.2.3 Member Function Documentation	29
7.2.3.1 __repr__()	29
7.2.3.2 __str__()	30
7.2.3.3 bringBackItem()	30
7.2.3.4 canRent()	31
7.2.3.5 getBalance()	31

7.2.3.6 getName()	32
7.2.3.7 makePayment()	32
7.2.3.8 rentItem()	32
7.2.3.9 setBalance()	33
7.2.4 Member Data Documentation	33
7.2.4.1 __balance	33
7.2.4.2 __itemsOut	34
7.2.4.3 __name	34
7.3 GenreSearch.GenreSearch Class Reference	35
7.3.1 Detailed Description	36
7.3.2 Constructor & Destructor Documentation	36
7.3.2.1 __init__()	36
7.3.3 Member Function Documentation	36
7.3.3.1 matches()	36
7.3.4 Member Data Documentation	37
7.3.4.1 __genre	37
7.4 Item.Item Class Reference	37
7.4.1 Detailed Description	39
7.4.2 Member Function Documentation	39
7.4.2.1 calculateLateFee()	39
7.4.2.2 getBarcode()	39
7.4.2.3 getDueDate()	40
7.4.2.4 getGenre()	40
7.4.2.5 getRentalCost()	40
7.4.2.6 getRentalDays()	41
7.4.2.7 getTitle()	41
7.4.2.8 isRented()	41
7.4.2.9 setDueDate()	41
7.4.2.10 setRented()	42
7.4.2.11 setReturned()	42
7.4.3 Member Data Documentation	43
7.4.3.1 __metaclass__	43
7.5 OverDuelItemSearch.OverDuelItemSearch Class Reference	43
7.5.1 Detailed Description	44
7.5.2 Constructor & Destructor Documentation	44
7.5.2.1 __init__()	44
7.5.3 Member Function Documentation	45
7.5.3.1 matches()	45
7.5.4 Member Data Documentation	45

7.5.4.1 __date	45
7.6 SearchCondition.SearchCondition Class Reference	46
7.6.1 Detailed Description	47
7.6.2 Member Function Documentation	47
7.6.2.1 matches()	47
7.6.3 Member Data Documentation	47
7.6.3.1 __metaclass__	47
7.7 SimpleDate.SimpleDate Class Reference	48
7.7.1 Detailed Description	49
7.7.2 Constructor & Destructor Documentation	49
7.7.2.1 __init__()	49
7.7.3 Member Function Documentation	50
7.7.3.1 __add__()	50
7.7.3.2 __eq__()	50
7.7.3.3 __lt__()	51
7.7.3.4 __repr__()	51
7.7.3.5 __str__()	51
7.7.3.6 __sub__()	51
7.7.3.7 daysUntil()	52
7.7.3.8 isBefore()	52
7.7.3.9 SimpleDateFromDays()	53
7.7.3.10 today()	53
7.7.3.11 todayNow()	53
7.7.4 Member Data Documentation	53
7.7.4.1 __date	54
7.7.4.2 MILLIS_IN_24_HOURS	54
7.8 StatusException.StatusException Class Reference	54
7.8.1 Detailed Description	55
7.8.2 Constructor & Destructor Documentation	55
7.8.2.1 __init__()	55
7.8.3 Member Data Documentation	56
7.8.3.1 msg	56
7.9 VideoStore.VideoStore Class Reference	56
7.9.1 Detailed Description	57
7.9.2 Constructor & Destructor Documentation	58
7.9.2.1 __init__()	58
7.9.3 Member Function Documentation	58
7.9.3.1 __repr__()	58
7.9.3.2 __str__()	58

7.9.3.3 addCustomer()	58
7.9.3.4 addItem()	59
7.9.3.5 findUser()	59
7.9.3.6 search()	60
7.9.4 Member Data Documentation	60
7.9.4.1 __customers	60
7.9.4.2 __items	61
8 File Documentation	63
8.1 AbstractItem.py File Reference	63
8.2 Customer.py File Reference	63
8.3 GenreSearch.py File Reference	64
8.4 Item.py File Reference	64
8.5 OverDuelItemSearch.py File Reference	64
8.6 SearchCondition.py File Reference	65
8.7 SimpleDate.py File Reference	65
8.8 StatusException.py File Reference	65
8.9 VideoStore.py File Reference	66
Index	67

Chapter 1

VideoStore - A video store that rents games and DVDs.

The purpose of this assignment is to give you some experience working with inheritance. In this homework, you will implement code for a simple video store.

A video store consists of a list of items (various kinds of games and DVDs) and a list of customers (people who can rent items):

- Different kinds of items may have different policies for how long they may be rented, the cost of rental, and how late fees are calculated.
- Different kinds of customers may have different behavior, when it comes to late fees or other aspects of renting and returning items.

Please note that all cost amounts in the code are expressed in cents, e.g., where the spec says \$1.50, your code would use the integer value 150.

1.1 release.notes

This program runs either in python 2.7 or python 3.6

To run the program:

- python [VideoStore.py](#)

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

AbstractItem	11
Customer	12
GenreSearch	13
Item	13
OverDuelItemSearch	14
SearchCondition	14
SimpleDate	15
StatusException	16
VideoStore	16

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Exception	
StatusException.StatusException	54
object	
Customer.Customer	27
SimpleDate.SimpleDate	48
VideoStore.VideoStore	56
ABC	
Item.Item	37
AbstractItem.AbstractItem	19
SearchCondition.SearchCondition	46
GenreSearch.GenreSearch	35
OverDuelItemSearch.OverDuelItemSearch	43

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

AbstractItem.AbstractItem	
Partial implementation of the Item interface	19
Customer.Customer	
A Customer is a client of a VideoStore who can rent items	27
GenreSearch.GenreSearch	
Implementation of SearchCondition that matches items based on the genre (not case sensitive) . . .	35
Item.Item	
This is an abstract class	37
OverDuelItemSearch.OverDuelItemSearch	
A class for matching items with due date before a given date	43
SearchCondition.SearchCondition	
Abstraction of a search predicate for items	46
SimpleDate.SimpleDate	
Date consisting of a year, month, and day	48
StatusException.StatusException	
Exception type thrown for invalid operations such as attempting to rent an item that is already rented	54
VideoStore.VideoStore	
VideoStore consists of a list of items and a list of customers who can rent items	56

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

AbstractItem.py	63
Customer.py	63
GenreSearch.py	64
Item.py	64
OverDuelItemSearch.py	64
SearchCondition.py	65
SimpleDate.py	65
StatusException.py	65
VideoStore.py	66

Chapter 6

Namespace Documentation

6.1 AbstractItem Namespace Reference

Classes

- class [AbstractItem](#)
Partial implementation of the [Item](#) interface.

Functions

- def [main](#) ()

6.1.1 Detailed Description

Partial implementation of the [Item](#) interface.

Author

Paulo Roma

Since

11/07/2017

6.1.2 Function Documentation

6.1.2.1 main()

```
def AbstractItem.main ( )
```

Definition at line 143 of file AbstractItem.py.

6.2 Customer Namespace Reference

Classes

- class [Customer](#)
A [Customer](#) is a client of a [VideoStore](#) who can rent items.

Functions

- def [main](#) ()

6.2.1 Detailed Description

A class describing a customer of the Video Store.

Author

Paulo Roma

Since

11/07/2017

6.2.2 Function Documentation

6.2.2.1 main()

```
def Customer.main ( )
```

Definition at line 147 of file Customer.py.

6.3 GenreSearch Namespace Reference

Classes

- class [GenreSearch](#)

Implementation of [SearchCondition](#) that matches items based on the genre (not case sensitive).

6.3.1 Detailed Description

Class for searching items matching a given genre.

Author

Paulo Roma

Since

11/07/2017

6.4 Item Namespace Reference

Classes

- class [Item](#)

This is an abstract class.

Variables

- [ABC](#) = object

6.4.1 Detailed Description

A class modeling an item of the Video Store.

Author

Paulo Roma

Since

11/07/2017

6.4.2 Variable Documentation

6.4.2.1 ABC

`Item.ABC = object`

Definition at line 15 of file Item.py.

6.5 OverDuelItemSearch Namespace Reference

Classes

- class [OverDuelItemSearch](#)
A class for matching items with due date before a given date.

6.5.1 Detailed Description

A class for looking for overdue items in the Video Store.

Author

Paulo Roma

Since

26/11/2018

6.6 SearchCondition Namespace Reference

Classes

- class [SearchCondition](#)
Abstraction of a search predicate for items.

Variables

- [ABC](#) = object

6.6.1 Detailed Description

A search predicate for items.

Author

Paulo Roma

Since

11/07/2017

6.6.2 Variable Documentation

6.6.2.1 ABC

```
SearchCondition.ABC = object
```

Definition at line 15 of file SearchCondition.py.

6.7 SimpleDate Namespace Reference

Classes

- class [SimpleDate](#)
Date consisting of a year, month, and day.

Functions

- def [main](#) ()
Main program for testing.

6.7.1 Detailed Description

Packages dates in the format yyyy-mm-dd.

Author

Paulo Roma

Since

11/07/2017

6.7.2 Function Documentation

6.7.2.1 main()

```
def SimpleDate.main ( )
```

Main program for testing.

Definition at line 112 of file SimpleDate.py.

6.8 StatusException Namespace Reference

Classes

- class [StatusException](#)

Exception type thrown for invalid operations such as attempting to rent an item that is already rented.

6.8.1 Detailed Description

Exceptions to be thrown when an error has occurred.

Author

Paulo Roma

Since

11/07/2017

6.9 VideoStore Namespace Reference

Classes

- class [VideoStore](#)

VideoStore consists of a list of items and a list of customers who can rent items.

Functions

- def [main](#) ()

Main program for testing.

6.9.1 Detailed Description

A video store that rents DVDs, and Games.

Author

Paulo Roma

Since

11/07/2017

6.9.2 Function Documentation

6.9.2.1 main()

```
def VideoStore.main ( )
```

Main program for testing.

Definition at line 134 of file VideoStore.py.

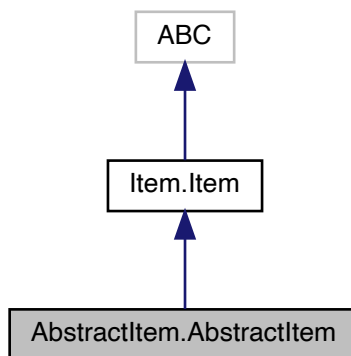
Chapter 7

Class Documentation

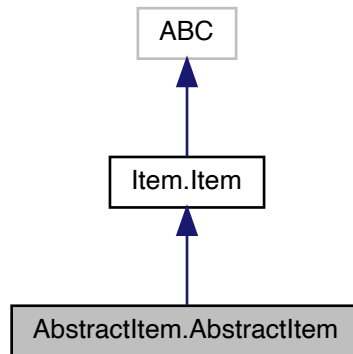
7.1 AbstractItem.AbstractItem Class Reference

Partial implementation of the [Item](#) interface.

Inheritance diagram for AbstractItem.AbstractItem:



Collaboration diagram for AbstractItem.AbstractItem:



Public Member Functions

- def `__init__` (self, title, genre, barcode)
Constructs a new [Item](#) with the given title, genre, and barcode.
- def `isRented` (self)
Returns whether this item is rented.
- def `getGenre` (self)
Gets this genre.
- def `getDueDate` (self)
Returns this item due date.
- def `setDueDate` (self, date=None)
Sets this item due date.
- def `getTitle` (self)
Returns this item title.
- def `getBarcode` (self)
Returns this item barcode.
- def `setReturned` (self)
Returns this item, if it is currently rented.
- def `setRented` (self, today)
Rents this item if it is not already rented and sets the due date.
- def `__str__` (self)
Returns a string representation of this item.
- def `__repr__` (self)
Returns a representation of the state of this object as a multiline string.
- def `__eq__` (self, obj)
Determines whether this item is the same as another one based on its barcode.

Private Attributes

- [__title](#)
Title of this item.
- [__genre](#)
Genre of this item.
- [__barcode](#)
Unique Barcode for this item.
- [__rented](#)
Rental status of this item.
- [__dueDate](#)
Due date for this item.

7.1.1 Detailed Description

Partial implementation of the [Item](#) interface.

Definition at line 19 of file AbstractItem.py.

7.1.2 Constructor & Destructor Documentation

7.1.2.1 `__init__()`

```
def AbstractItem.AbstractItem.__init__ (
    self,
    title,
    genre,
    barcode )
```

Constructs a new [Item](#) with the given title, genre, and barcode.

This constructor may only be invoked by subclasses.

Parameters

<i>title</i>	the title of the item
<i>genre</i>	the genre of the item
<i>barcode</i>	a unique integer identifier for the item

Definition at line 28 of file AbstractItem.py.

7.1.3 Member Function Documentation

7.1.3.1 `__eq__()`

```
def AbstractItem.AbstractItem.__eq__ (
    self,
    obj )
```

Determines whether this item is the same as another one based on its barcode.

Parameters

<i>obj</i>	the object to compare to this item
------------	------------------------------------

Returns

true if the given object is an [AbstractItem](#) with the same barcode as this one

Definition at line 138 of file AbstractItem.py.

References AbstractItem.AbstractItem.getBarcode().

7.1.3.2 `__repr__()`

```
def AbstractItem.AbstractItem.__repr__ (
    self )
```

Returns a representation of the state of this object as a multiline string.

The format is:

```
type
title
(genre)
status
```

The status is either

```
Rented: yyyy-mm-dd
```

or

```
Available
```

where "yyyy-mm-dd" is the current due date.

Returns

a string representation of this object

Definition at line 119 of file AbstractItem.py.

References `AbstractItem.AbstractItem.getDueDate()`, `AbstractItem.AbstractItem.getGenre()`, `AbstractItem.AbstractItem.getTitle()`, and `AbstractItem.AbstractItem.isRented()`.

7.1.3.3 `__str__()`

```
def AbstractItem.AbstractItem.__str__ (
    self )
```

Returns a string representation of this item.

Definition at line 95 of file AbstractItem.py.

References `AbstractItem.AbstractItem.getTitle()`.

7.1.3.4 `getBarcode()`

```
def AbstractItem.AbstractItem.getBarcode (
    self )
```

Returns this item barcode.

Reimplemented from [Item.Item](#).

Definition at line 71 of file AbstractItem.py.

References `AbstractItem.AbstractItem.__barcode`.

Referenced by `AbstractItem.AbstractItem.__eq__()`, and `AbstractItem.AbstractItem.setReturned()`.

7.1.3.5 `getDueDate()`

```
def AbstractItem.AbstractItem.getDueDate (
    self )
```

Returns this item due date.

Reimplemented from [Item.Item](#).

Definition at line 52 of file AbstractItem.py.

References `AbstractItem.AbstractItem.__dueDate`, and `AbstractItem.AbstractItem.__rented`.

Referenced by `AbstractItem.AbstractItem.__repr__()`.

7.1.3.6 getGenre()

```
def AbstractItem.AbstractItem.getGenre (
    self )
```

Gets this genre.

Returns

genre.

Reimplemented from [Item.Item](#).

Definition at line 48 of file AbstractItem.py.

References AbstractItem.AbstractItem.__genre.

Referenced by AbstractItem.AbstractItem.__repr__().

7.1.3.7 getTitle()

```
def AbstractItem.AbstractItem.getTitle (
    self )
```

Returns this item title.

Reimplemented from [Item.Item](#).

Definition at line 67 of file AbstractItem.py.

References AbstractItem.AbstractItem.__title.

Referenced by AbstractItem.AbstractItem.__repr__(), and AbstractItem.AbstractItem.__str__().

7.1.3.8 isRented()

```
def AbstractItem.AbstractItem.isRented (
    self )
```

Returns whether this item is rented.

Reimplemented from [Item.Item](#).

Definition at line 41 of file AbstractItem.py.

References AbstractItem.AbstractItem.__rented.

Referenced by AbstractItem.AbstractItem.__repr__(), and AbstractItem.AbstractItem.setReturned().

7.1.3.9 setDueDate()

```
def AbstractItem.AbstractItem.setDueDate (
    self,
    date = None )
```

Sets this item due date.

If a date is given, the item is set as rented.

Parameters

<i>date</i>	due date.
-------------	-----------

Reimplemented from [Item.Item](#).

Definition at line 62 of file AbstractItem.py.

References AbstractItem.AbstractItem.__dueDate, and AbstractItem.AbstractItem.__rented.

Referenced by AbstractItem.AbstractItem.setReturned().

7.1.3.10 setRented()

```
def AbstractItem.AbstractItem.setRented (
    self,
    today )
```

Rents this item if it is not already rented and sets the due date.

Parameters

<i>today</i>	rental date.
--------------	--------------

Exceptions

StatusException	
---------------------------------	--

Reimplemented from [Item.Item](#).

Definition at line 90 of file AbstractItem.py.

7.1.3.11 setReturned()

```
def AbstractItem.AbstractItem.setReturned (
    self )
```

Returns this item, if it is currently rented.

Sets its dueDate to None.

Exceptions

StatusException	
---------------------------------	--

Definition at line 79 of file AbstractItem.py.

References `AbstractItem.AbstractItem.getBarcode()`, `AbstractItem.AbstractItem.isRented()`, and `AbstractItem.setDueDate()`.

7.1.4 Member Data Documentation

7.1.4.1 `__barcode`

`AbstractItem.AbstractItem.__barcode` [private]

Unique Barcode for this item.

Definition at line 34 of file AbstractItem.py.

Referenced by `AbstractItem.AbstractItem.getBarcode()`.

7.1.4.2 `__dueDate`

`AbstractItem.AbstractItem.__dueDate` [private]

Due date for this item.

Definition at line 38 of file AbstractItem.py.

Referenced by `AbstractItem.AbstractItem.getDueDate()`, and `AbstractItem.AbstractItem.setDueDate()`.

7.1.4.3 `__genre`

`AbstractItem.AbstractItem.__genre` [private]

Genre of this item.

Definition at line 32 of file AbstractItem.py.

Referenced by `AbstractItem.AbstractItem.getGenre()`, and `GenreSearch.GenreSearch.matches()`.

7.1.4.4 `__rented`

```
AbstractItem.AbstractItem.__rented [private]
```

Rental status of this item.

Definition at line 36 of file AbstractItem.py.

Referenced by `AbstractItem.AbstractItem.getDueDate()`, `AbstractItem.AbstractItem.isRented()`, and `AbstractItem.
AbstractItem.setDueDate()`.

7.1.4.5 `__title`

```
AbstractItem.AbstractItem.__title [private]
```

Title of this item.

Definition at line 30 of file AbstractItem.py.

Referenced by `AbstractItem.AbstractItem.getTitle()`.

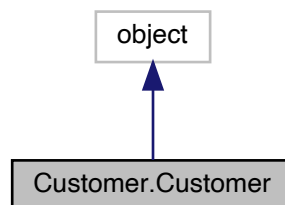
The documentation for this class was generated from the following file:

- [AbstractItem.py](#)

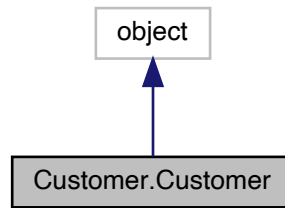
7.2 Customer.Customer Class Reference

A [Customer](#) is a client of a [VideoStore](#) who can rent items.

Inheritance diagram for `Customer.Customer`:



Collaboration diagram for Customer.Customer:



Public Member Functions

- `def __init__ (self, name)`
Constructs a [Customer](#) with the given name.
- `def rentItem (self, item, today)`
Rents an item and adds it to this customer's list of items.
- `def setBalance (self, fee, date, today)`
Updates the balance of this customer.
- `def bringBackItem (self, barcode, today)`
Returns an item that this customer currently has rented and updates the balance if a late fee or credit is due.
- `def getBalance (self)`
Returns the balance for this customer.
- `def getName (self)`
Returns the name of this customer.
- `def makePayment (self, amount)`
Makes a payment on this customer's balance.
- `def __str__ (self)`
Returns a string representation of this customer.
- `def __repr__ (self)`
Returns a string representation of this customer.
- `def canRent (self, today)`
Helper method determines whether this customer already has overdue items.

Private Attributes

- `__itemsOut`
Items currently rented by this customer.
- `__name`
Name of this customer.
- `__balance`
Balance currently owed by this customer.

7.2.1 Detailed Description

A [Customer](#) is a client of a [VideoStore](#) who can rent items.

A client is identified by a unique name. At any given time a [Customer](#) has a list of items currently rented, and a balance representing rental charges, late fees, or credits (where a negative balance indicates a credit). Balances are in cents. Ordinary customers are not allowed to rent new items if they have any items overdue.

Definition at line 26 of file Customer.py.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 `__init__()`

```
def Customer.Customer.__init__ (
    self,
    name )
```

Constructs a [Customer](#) with the given name.

Initially there are no items rented and the balance is zero.

Parameters

<i>name</i>	the new customer's name
-------------	-------------------------

Definition at line 34 of file Customer.py.

7.2.3 Member Function Documentation

7.2.3.1 `__repr__()`

```
def Customer.Customer.__repr__ (
    self )
```

Returns a string representation of this customer.

The format consists of multiple lines. The first line is the patron's name. Subsequent lines are formed from the `repr()` values of the items currently rented, separated by a newline.

Returns

representation of this object as a multi-line string

Definition at line 125 of file Customer.py.

References Customer.Customer.__itemsOut, and Customer.Customer.__name.

7.2.3.2 __str__()

```
def Customer.Customer.__str__ (
    self )
```

Returns a string representation of this customer.

The format is the customer type, name and balance.

Definition at line 114 of file Customer.py.

References Customer.Customer.__itemsOut, Customer.Customer.__name, and Customer.Customer.getBalance().

7.2.3.3 bringBackItem()

```
def Customer.Customer.bringBackItem (
    self,
    barcode,
    today )
```

Returns an item that this customer currently has rented and updates the balance if a late fee or credit is due.

If the item can be successfully returned, this method updates the item's status and removes it from this customer's list of items. If the customer does not have the item rented, a [StatusException](#) is thrown.

Parameters

<i>barcode</i>	identifier for the item to be returned
<i>today</i>	the date on which the item is being returned

Exceptions

StatusException	if this customer does not have the given item rented
---------------------------------	--

Definition at line 83 of file Customer.py.

7.2.3.4 canRent()

```
def Customer.Customer.canRent (
    self,
    today )
```

Helper method determines whether this customer already has overdue items.

Parameters

<i>today</i>	the current date
--------------	------------------

Returns

true if the customer has no overdue items, false otherwise

Definition at line 141 of file Customer.py.

References Customer.Customer.__itemsOut.

Referenced by Customer.Customer.rentItem().

7.2.3.5 getBalance()

```
def Customer.Customer.getBalance (
    self )
```

Returns the balance for this customer.

Returns

this customer's balance

Definition at line 92 of file Customer.py.

References Customer.Customer.__balance.

Referenced by Customer.Customer.__str__().

7.2.3.6 getName()

```
def Customer.Customer.getName (
    self )
```

Returns the name of this customer.

Returns

this customer's name

Definition at line 100 of file Customer.py.

References Customer.Customer.__name.

7.2.3.7 makePayment()

```
def Customer.Customer.makePayment (
    self,
    amount )
```

Makes a payment on this customer's balance.

Parameters

<i>amount</i>	the amount to be paid, in cents
---------------	---------------------------------

Definition at line 108 of file Customer.py.

References Customer.Customer.__balance.

7.2.3.8 rentItem()

```
def Customer.Customer.rentItem (
    self,
    item,
    today )
```

Rents an item and adds it to this customer's list of items.

If the item can be rented, this method updates the item's status (including the due date) and then adds it to this customer's list of items. If the item cannot be rented to this customer, a [StatusException](#) is thrown.

Parameters

<i>item</i>	the item to be rented
<i>today</i>	the date on which the item is being rented

Exceptions

<i>StatusException</i>	if the item cannot be rented to this customer for any reason
--	--

Definition at line 54 of file Customer.py.

References Customer.Customer.__balance, Customer.Customer.__itemsOut, and Customer.Customer.canRent().

7.2.3.9 setBalance()

```
def Customer.Customer.setBalance (
    self,
    fee,
    date,
    today )
```

Updates the balance of this customer.

Parameters

<i>fee</i>	late fee.
<i>date</i>	due date.
<i>today</i>	return date.

Definition at line 67 of file Customer.py.

References Customer.Customer.__balance.

7.2.4 Member Data Documentation

7.2.4.1 __balance

```
Customer.Customer.__balance [private]
```

Balance currently owed by this customer.

Definition at line 40 of file Customer.py.

Referenced by Customer.Customer.getBalance(), Customer.Customer.makePayment(), Customer.Customer.rentItem(), and Customer.Customer.setBalance().

7.2.4.2 `__itemsOut`

```
Customer.Customer.__itemsOut    [private]
```

Items currently rented by this customer.

Definition at line 36 of file Customer.py.

Referenced by Customer.Customer.__repr__(), Customer.Customer.__str__(), Customer.Customer.canRent(), and Customer.Customer.rentItem().

7.2.4.3 `__name`

```
Customer.Customer.__name    [private]
```

Name of this customer.

Definition at line 38 of file Customer.py.

Referenced by Customer.Customer.__repr__(), Customer.Customer.__str__(), and Customer.Customer.getName().

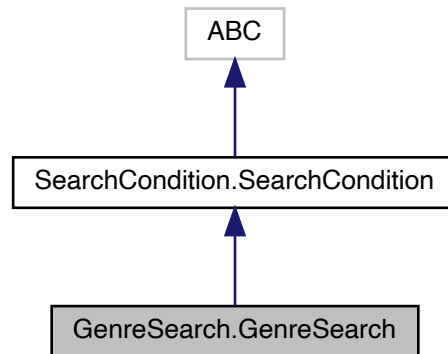
The documentation for this class was generated from the following file:

- [Customer.py](#)

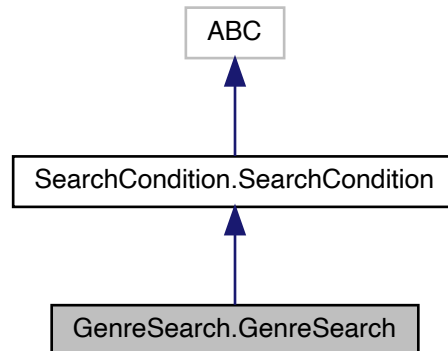
7.3 GenreSearch.GenreSearch Class Reference

Implementation of [SearchCondition](#) that matches items based on the genre (not case sensitive).

Inheritance diagram for GenreSearch.GenreSearch:



Collaboration diagram for GenreSearch.GenreSearch:



Public Member Functions

- def [__init__](#) (self, genre)
Constructs a [GenreSearch](#) for the given value.
- def [matches](#) (self, item)
Matches this genre with the genre of a given item.

Private Attributes

- [__genre](#)

The genre we are searching for.

7.3.1 Detailed Description

Implementation of [SearchCondition](#) that matches items based on the genre (not case sensitive).

Definition at line 17 of file GenreSearch.py.

7.3.2 Constructor & Destructor Documentation

7.3.2.1 `__init__()`

```
def GenreSearch.GenreSearch.__init__ (
    self,
    genre )
```

Constructs a [GenreSearch](#) for the given value.

Parameters

<i>genre</i>	the genre to search for
--------------	-------------------------

Definition at line 24 of file GenreSearch.py.

7.3.3 Member Function Documentation

7.3.3.1 `matches()`

```
def GenreSearch.GenreSearch.matches (
    self,
    item )
```

Matches this genre with the genre of a given item.

Parameters

<i>item</i>	containing the genre to compare to.
-------------	-------------------------------------

Reimplemented from [SearchCondition.SearchCondition](#).

Definition at line 33 of file GenreSearch.py.

References `GenreSearch.GenreSearch.__genre`, and `AbstractItem.AbstractItem.__genre`.

7.3.4 Member Data Documentation

7.3.4.1 `__genre`

```
GenreSearch.GenreSearch.__genre [private]
```

The genre we are searching for.

Definition at line 26 of file GenreSearch.py.

Referenced by `GenreSearch.GenreSearch.matches()`.

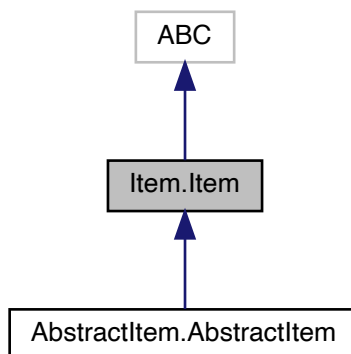
The documentation for this class was generated from the following file:

- [GenreSearch.py](#)

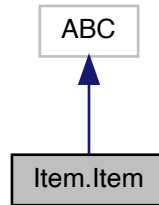
7.4 Item.Item Class Reference

This is an abstract class.

Inheritance diagram for `Item.Item`:



Collaboration diagram for Item.Item:



Public Member Functions

- def [setRented](#) (self, today)
Rents this item if it is not already rented and sets the due date.
- def [setReturned](#) (self, today)
Returns this item, if it is currently rented.
- def [setDueDate](#) (self, date=None)
Sets this item due date.
- def [getRentalCost](#) (self)
Returns the cost to rent this item.
- def [getRentalDays](#) (self)
Returns the number of days allowed to keep this item.
- def [calculateLateFee](#) (self, today)
Calculates the late fee (or bonus) that would be charged (or applied) for returning the item on the given date.
- def [getGenre](#) (self)
Returns a String representing the genre of this item.
- def [isRented](#) (self)
Determines whether this item is currently rented.
- def [getDueDate](#) (self)
Returns the due date for this item if it is currently rented, or null if the item is not rented.
- def [getTitle](#) (self)
Returns the title of this item.
- def [getBarcode](#) (self)
Returns the integer barcode for this item.

Static Private Attributes

- [__metaclass__](#) = ABCMeta

7.4.1 Detailed Description

This is an abstract class.

All methods here must be implemented elsewhere.

An item represents a movie or game that can be rented from a video store. Each item has a title, a genre, and a unique integer identifier called a barcode. An item can be rented or available, and if rented it has a due date.

Definition at line 25 of file Item.py.

7.4.2 Member Function Documentation

7.4.2.1 calculateLateFee()

```
def Item.Item.calculateLateFee (
    self,
    today )
```

Calculates the late fee (or bonus) that would be charged (or applied) for returning the item on the given date.

Parameters

<i>today</i>	the date on which the item is being returned
--------------	--

Returns

the late fee or bonus for returning the item on the given date, or zero if the item is not currently rented

Definition at line 87 of file Item.py.

7.4.2.2 getBarcode()

```
def Item.Item.getBarcode (
    self )
```

Returns the integer barcode for this item.

Returns

barcode of this item

Reimplemented in [AbstractItem.AbstractItem](#).

Definition at line 133 of file Item.py.

7.4.2.3 `getDueDate()`

```
def Item.Item.getDueDate (
    self )
```

Returns the due date for this item if it is currently rented, or null if the item is not rented.

Returns

due date for this item

Reimplemented in [AbstractItem.AbstractItem](#).

Definition at line 115 of file Item.py.

7.4.2.4 `getGenre()`

```
def Item.Item.getGenre (
    self )
```

Returns a String representing the genre of this item.

Returns

genre of this item

Reimplemented in [AbstractItem.AbstractItem](#).

Definition at line 96 of file Item.py.

7.4.2.5 `getRentalCost()`

```
def Item.Item.getRentalCost (
    self )
```

Returns the cost to rent this item.

Returns

cost to rent the item

Definition at line 66 of file Item.py.

7.4.2.6 getRentalDays()

```
def Item.Item.getRentalDays (
    self )
```

Returns the number of days allowed to keep this item.

Returns

number of allowed rental days.

Definition at line 74 of file Item.py.

7.4.2.7 getTitle()

```
def Item.Item.getTitle (
    self )
```

Returns the title of this item.

Returns

title of this item

Reimplemented in [AbstractItem.AbstractItem](#).

Definition at line 124 of file Item.py.

7.4.2.8 isRented()

```
def Item.Item.isRented (
    self )
```

Determines whether this item is currently rented.

Returns

true if this item is rented, false otherwise

Reimplemented in [AbstractItem.AbstractItem](#).

Definition at line 105 of file Item.py.

7.4.2.9 setDueDate()

```
def Item.Item.setDueDate (
    self,
    date = None )
```

Sets this item due date.

If a date is given, the item is set as rented.

Parameters

<i>date</i>	due date.
-------------	-----------

Reimplemented in [AbstractItem.AbstractItem](#).

Definition at line 57 of file Item.py.

7.4.2.10 setRented()

```
def Item.Item.setRented (
    self,
    today )
```

Rents this item if it is not already rented and sets the due date.

Parameters

<i>today</i>	the date on which this item is being rented
--------------	---

Exceptions

StatusException	if the item cannot be rented
---------------------------------	------------------------------

Reimplemented in [AbstractItem.AbstractItem](#).

Definition at line 37 of file Item.py.

7.4.2.11 setReturned()

```
def Item.Item.setReturned (
    self,
    today )
```

Returns this item, if it is currently rented.

Parameters

<i>today</i>	the date on which the item is being returned
--------------	--

Exceptions

StatusException	if the item is not currently rented
---------------------------------	-------------------------------------

Definition at line 48 of file Item.py.

7.4.3 Member Data Documentation

7.4.3.1 `__metaclass__`

```
Item.Item.__metaclass__ = ABCMeta [static], [private]
```

Definition at line 26 of file Item.py.

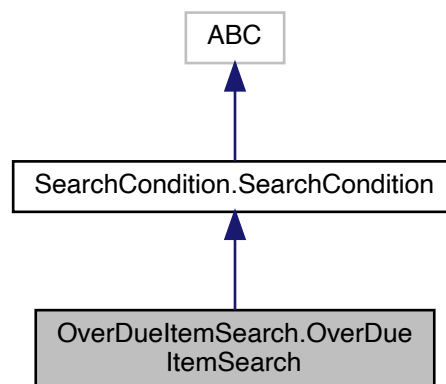
The documentation for this class was generated from the following file:

- [Item.py](#)

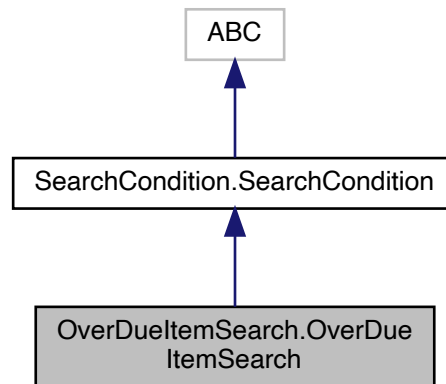
7.5 OverDuelItemSearch.OverDuelItemSearch Class Reference

A class for matching items with due date before a given date.

Inheritance diagram for OverDuelItemSearch.OverDuelItemSearch:



Collaboration diagram for OverDuelItemSearch.OverDuelItemSearch:



Public Member Functions

- def `__init__` (self, date)
Constructor.
- def `matches` (self, item)
Return True if the item's due date is before a given date.

Private Attributes

- `__date`
date for the search.

7.5.1 Detailed Description

A class for matching items with due date before a given date.

Definition at line 16 of file OverDuelItemSearch.py.

7.5.2 Constructor & Destructor Documentation

7.5.2.1 `__init__`()

```
def OverDuelItemSearch.OverDuelItemSearch.__init__ (  
    self,  
    date )
```

Constructor.

Parameters

<i>date</i>	given date.
-------------	-------------

Definition at line 21 of file OverDueItemSearch.py.

7.5.3 Member Function Documentation

7.5.3.1 matches()

```
def OverDueItemSearch.OverDueItemSearch.matches (
    self,
    item )
```

Return True if the item's due date is before a given date.

Returns

match with due dates smaller then date.

Reimplemented from [SearchCondition.SearchCondition](#).

Definition at line 30 of file OverDueItemSearch.py.

References [OverDueItemSearch.OverDueItemSearch.__date](#).

7.5.4 Member Data Documentation

7.5.4.1 __date

```
OverDueItemSearch.OverDueItemSearch.__date [private]
```

date for the search.

Definition at line 23 of file OverDueItemSearch.py.

Referenced by [SimpleDate.SimpleDate.__eq__\(\)](#), [SimpleDate.SimpleDate.__repr__\(\)](#), [SimpleDate.SimpleDate.__str__\(\)](#), [SimpleDate.SimpleDate.daysUntil\(\)](#), and [OverDueItemSearch.OverDueItemSearch.matches\(\)](#).

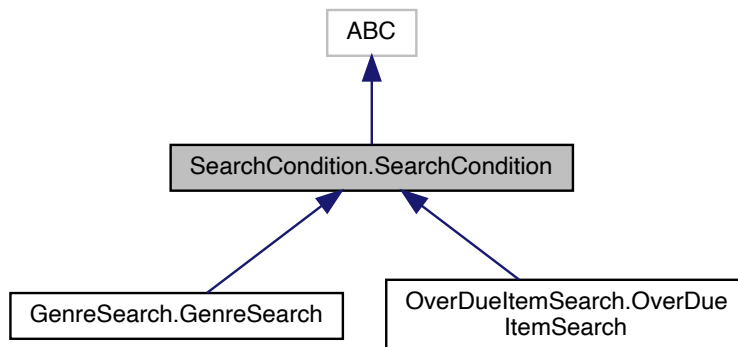
The documentation for this class was generated from the following file:

- [OverDueItemSearch.py](#)

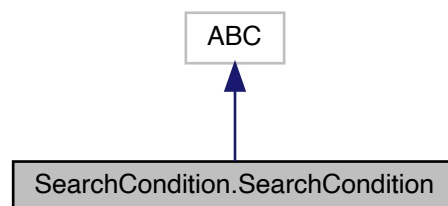
7.6 SearchCondition.SearchCondition Class Reference

Abstraction of a search predicate for items.

Inheritance diagram for SearchCondition.SearchCondition:



Collaboration diagram for SearchCondition.SearchCondition:



Public Member Functions

- def [matches](#) (self, item)
Determine whether the given item matches this search condition's criteria for inclusion.

Static Private Attributes

- [__metaclass__](#) = ABCMeta

7.6.1 Detailed Description

Abstraction of a search predicate for items.

Subtypes can customize the nature of the search:

- e.g., exact title, title keywords, genre, etc.

Definition at line 22 of file SearchCondition.py.

7.6.2 Member Function Documentation

7.6.2.1 matches()

```
def SearchCondition.SearchCondition.matches (
    self,
    item )
```

Determine whether the given item matches this search condition's criteria for inclusion.

Parameters

<i>item</i>	the item to be checked
-------------	------------------------

Returns

true if the item matches this condition's criteria, false otherwise

Reimplemented in [GenreSearch.GenreSearch](#), and [OverDuelItemSearch.OverDuelItemSearch](#).

Definition at line 33 of file SearchCondition.py.

7.6.3 Member Data Documentation

7.6.3.1 __metaclass__

```
SearchCondition.SearchCondition.__metaclass__ = ABCMeta [static], [private]
```

Definition at line 23 of file SearchCondition.py.

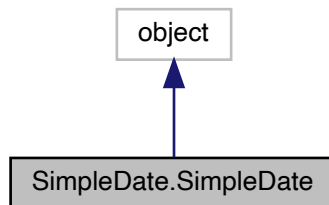
The documentation for this class was generated from the following file:

- [SearchCondition.py](#)

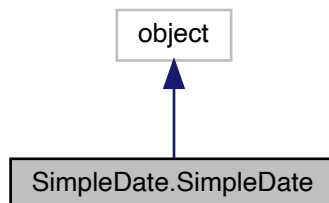
7.7 SimpleDate.SimpleDate Class Reference

Date consisting of a year, month, and day.

Inheritance diagram for SimpleDate.SimpleDate:



Collaboration diagram for SimpleDate.SimpleDate:



Public Member Functions

- def `__init__` (self, year, month, day, hour=0, min=0, sec=0)
Constructs a [SimpleDate](#) with the given year, month, and day.
- def `SimpleDateFromDays` (self, additionalDays)
Returns a [SimpleDate](#) that is a given number of days after this date.
- def `isBefore` (self, other)
Determines whether this date is strictly earlier than the given date.
- def `daysUntil` (self, other)
Returns the number of days from this date until the given date.
- def `__eq__` (self, other)

- Operator ==.*
- def `__lt__` (self, other)
- Operator <.*
- def `__add__` (self, ndays)
- Operator +.*
- def `__sub__` (self, other)
- Operator -.*
- def `__repr__` (self)
- Prints this simpleDate in the format hour:min:sec day-month-year.*
- def `__str__` (self)
- Prints this simpleDate in the format year-month-day.*

Static Public Member Functions

- def `today` ()
- Return today's date.*
- def `todayNow` (delta=3)
- Return today's date and time.*

Static Public Attributes

- int `MILLIS_IN_24_HOURS` = 1000 * 60 * 60 * 24
- Number of milliseconds in one day.*

Private Attributes

- `__date`
- Holds the date in this [SimpleDate](#) object.*

7.7.1 Detailed Description

Date consisting of a year, month, and day.

Definition at line 17 of file SimpleDate.py.

7.7.2 Constructor & Destructor Documentation

7.7.2.1 `__init__()`

```
def SimpleDate.SimpleDate.__init__ (
    self,
    year,
    month,
    day,
    hour = 0,
    min = 0,
    sec = 0 )
```

Constructs a [SimpleDate](#) with the given year, month, and day.

Parameters

<i>year</i>	four-digit year
<i>month</i>	1-based month number
<i>day</i>	1-based day of month
<i>hour</i>	24-based hour time
<i>min</i>	minutes of time
<i>sec</i>	seconds of time

See also

https://www.tutorialspoint.com/python/time_strptime.htm

<https://docs.python.org/3/library/datetime.html>

Definition at line 46 of file SimpleDate.py.

7.7.3 Member Function Documentation

7.7.3.1 `__add__()`

```
def SimpleDate.SimpleDate.__add__ (
    self,
    ndays )
```

Operator +.

Adds ndays to this date.

Definition at line 91 of file SimpleDate.py.

References SimpleDate.SimpleDate.SimpleDateFromDays().

7.7.3.2 `__eq__()`

```
def SimpleDate.SimpleDate.__eq__ (
    self,
    other )
```

Operator ==.

Definition at line 83 of file SimpleDate.py.

References OverDuelItemSearch.OverDuelItemSearch.__date, and SimpleDate.SimpleDate.__date.

7.7.3.3 `__lt__()`

```
def SimpleDate.SimpleDate.__lt__ (
    self,
    other )
```

Operator <.

Definition at line 87 of file SimpleDate.py.

References SimpleDate.SimpleDate.isBefore().

7.7.3.4 `__repr__()`

```
def SimpleDate.SimpleDate.__repr__ (
    self )
```

Prints this simpleDate in the format hour:min:sec day-month-year.

Definition at line 101 of file SimpleDate.py.

References OverDuelItemSearch.OverDuelItemSearch.__date, and SimpleDate.SimpleDate.__date.

7.7.3.5 `__str__()`

```
def SimpleDate.SimpleDate.__str__ (
    self )
```

Prints this simpleDate in the format year-month-day.

Definition at line 107 of file SimpleDate.py.

References OverDuelItemSearch.OverDuelItemSearch.__date, and SimpleDate.SimpleDate.__date.

7.7.3.6 `__sub__()`

```
def SimpleDate.SimpleDate.__sub__ (
    self,
    other )
```

Operator -.

Number of days between this and other.

Returns

(other - self) in days.

Definition at line 96 of file SimpleDate.py.

7.7.3.7 daysUntil()

```
def SimpleDate.SimpleDate.daysUntil (
    self,
    other )
```

Returns the number of days from this date until the given date.

Returns a negative number if this date after the given date.

Parameters

<i>other</i>	the future date
--------------	-----------------

Returns

number of days until the given date (negative if it is in the past)

Definition at line 78 of file SimpleDate.py.

References OverDuelItemSearch.OverDuelItemSearch.__date, and SimpleDate.SimpleDate.__date.

Referenced by SimpleDate.SimpleDate.isBefore().

7.7.3.8 isBefore()

```
def SimpleDate.SimpleDate.isBefore (
    self,
    other )
```

Determines whether this date is strictly earlier than the given date.

Parameters

<i>other</i>	
--------------	--

Returns

true if this date is strictly before the given date. false otherwise

Definition at line 69 of file SimpleDate.py.

References SimpleDate.SimpleDate.daysUntil().

Referenced by SimpleDate.SimpleDate.__lt__().

7.7.3.9 SimpleDateFromDays()

```
def SimpleDate.SimpleDate.SimpleDateFromDays (
    self,
    additionalDays )
```

Returns a [SimpleDate](#) that is a given number of days after this date.

Parameters

<i>additionalDays</i>	the number of days to be added to this date
-----------------------	---

Definition at line 56 of file SimpleDate.py.

Referenced by SimpleDate.SimpleDate.__add__().

7.7.3.10 today()

```
def SimpleDate.SimpleDate.today ( ) [static]
```

Return today's date.

Definition at line 24 of file SimpleDate.py.

7.7.3.11 todayNow()

```
def SimpleDate.SimpleDate.todayNow (
    delta = 3 ) [static]
```

Return today's date and time.

Definition at line 30 of file SimpleDate.py.

7.7.4 Member Data Documentation

7.7.4.1 `__date`

```
SimpleDate.SimpleDate.__date [private]
```

Holds the date in this [SimpleDate](#) object.

Definition at line 49 of file SimpleDate.py.

Referenced by `SimpleDate.SimpleDate.__eq__()`, `SimpleDate.SimpleDate.__repr__()`, `SimpleDate.SimpleDate.__str__()`, and `SimpleDate.SimpleDate.daysUntil()`.

7.7.4.2 `MILLIS_IN_24_HOURS`

```
int SimpleDate.SimpleDate.MILLIS_IN_24_HOURS = 1000 * 60 * 60 * 24 [static]
```

Number of milliseconds in one day.

Definition at line 20 of file SimpleDate.py.

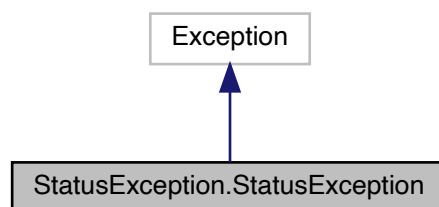
The documentation for this class was generated from the following file:

- [SimpleDate.py](#)

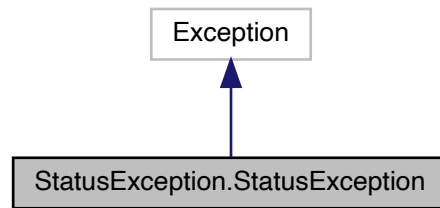
7.8 `StatusException.StatusException` Class Reference

Exception type thrown for invalid operations such as attempting to rent an item that is already rented.

Inheritance diagram for `StatusException.StatusException`:



Collaboration diagram for `StatusException.StatusException`:



Public Member Functions

- `def __init__(self, msg)`
Constructs a `StatusException` with the given message.

Public Attributes

- `msg`

7.8.1 Detailed Description

Exception type thrown for invalid operations such as attempting to rent an item that is already rented.

Definition at line 16 of file `StatusException.py`.

7.8.2 Constructor & Destructor Documentation

7.8.2.1 `__init__()`

```
def StatusException.StatusException.__init__(  
    self,  
    msg )
```

Constructs a `StatusException` with the given message.

Parameters

<i>msg</i>	message for this exception
------------	----------------------------

Definition at line 22 of file `StatusException.py`.

7.8.3 Member Data Documentation

7.8.3.1 `msg`

`StatusException.StatusException.msg`

hold the error message.

Definition at line 27 of file `StatusException.py`.

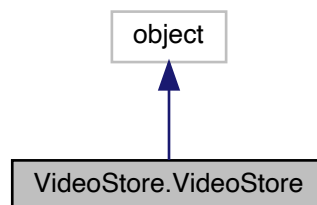
The documentation for this class was generated from the following file:

- [StatusException.py](#)

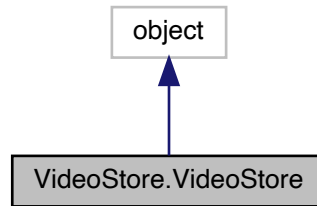
7.9 VideoStore.VideoStore Class Reference

[VideoStore](#) consists of a list of items and a list of customers who can rent items.

Inheritance diagram for `VideoStore.VideoStore`:



Collaboration diagram for VideoStore.VideoStore:



Public Member Functions

- `def __init__ (self)`
Constructs a [VideoStore](#) that initially has no items and no customers.
- `def addItem (self, item)`
Adds an item to this store's list of items, provided that there is not already an item with the same barcode.
- `def addCustomer (self, customer)`
Adds a customer to this store's list of customers.
- `def findUser (self, name)`
Returns the customer with the given name.
- `def search (self, condition)`
Search the store's collection for items satisfying the given [SearchCondition](#).
- `def __str__ (self)`
Returns the set of users and items in this store.
- `def __repr__ (self)`
Returns the set of users and items in this store.

Private Attributes

- `__items`
The items in this store.
- `__customers`
The list of customers of this store.

7.9.1 Detailed Description

[VideoStore](#) consists of a list of items and a list of customers who can rent items.

Definition at line 36 of file VideoStore.py.

7.9.2 Constructor & Destructor Documentation

7.9.2.1 `__init__()`

```
def VideoStore.VideoStore.__init__ (
    self )
```

Constructs a [VideoStore](#) that initially has no items and no customers.

Definition at line 41 of file VideoStore.py.

7.9.3 Member Function Documentation

7.9.3.1 `__repr__()`

```
def VideoStore.VideoStore.__repr__ (
    self )
```

Returns the set of users and items in this store.

Returns

a string with all users and items.

Definition at line 121 of file VideoStore.py.

References `VideoStore.VideoStore.__customers`, and `VideoStore.VideoStore.__items`.

7.9.3.2 `__str__()`

```
def VideoStore.VideoStore.__str__ (
    self )
```

Returns the set of users and items in this store.

Returns

a string with all users and items.

Definition at line 104 of file VideoStore.py.

References `VideoStore.VideoStore.__customers`, and `VideoStore.VideoStore.__items`.

7.9.3.3 `addCustomer()`

```
def VideoStore.VideoStore.addCustomer (
    self,
    customer )
```

Adds a customer to this store's list of customers.

Parameters

<i>customer</i>	the customer to be added
-----------------	--------------------------

Returns

customer

Definition at line 65 of file VideoStore.py.

References VideoStore.VideoStore.__customers.

7.9.3.4 addItem()

```
def VideoStore.VideoStore.addItem (
    self,
    item )
```

Adds an item to this store's list of items, provided that there is not already an item with the same barcode.

Parameters

<i>item</i>	the item to be added
-------------	----------------------

Returns

item

Definition at line 54 of file VideoStore.py.

References VideoStore.VideoStore.__items.

7.9.3.5 findUser()

```
def VideoStore.VideoStore.findUser (
    self,
    name )
```

Returns the customer with the given name.

Parameters

<i>name</i>	the name of the customer to search for
-------------	--

Returns

the customer

Definition at line 78 of file VideoStore.py.

References VideoStore.VideoStore.__customers.

7.9.3.6 search()

```
def VideoStore.VideoStore.search (
    self,
    condition )
```

Search the store's collection for items satisfying the given [SearchCondition](#).

Parameters

<i>condition</i>	the SearchCondition
------------------	-------------------------------------

Returns

list of items satisfying the condition

Definition at line 92 of file VideoStore.py.

References VideoStore.VideoStore.__items.

7.9.4 Member Data Documentation**7.9.4.1 __customers**

```
VideoStore.VideoStore.__customers [private]
```

The list of customers of this store.

Definition at line 45 of file VideoStore.py.

Referenced by VideoStore.VideoStore.__repr__(), VideoStore.VideoStore.__str__(), VideoStore.VideoStore.addCustomer(), and VideoStore.VideoStore.findUser().

7.9.4.2 `__items`

```
VideoStore.VideoStore.__items [private]
```

The items in this store.

Definition at line 43 of file VideoStore.py.

Referenced by `VideoStore.VideoStore.__repr__()`, `VideoStore.VideoStore.__str__()`, `VideoStore.VideoStore.addItem()`, and `VideoStore.VideoStore.search()`.

The documentation for this class was generated from the following file:

- [VideoStore.py](#)

Chapter 8

File Documentation

8.1 AbstractItem.py File Reference

Classes

- class [AbstractItem.AbstractItem](#)
Partial implementation of the [Item](#) interface.

Namespaces

- [AbstractItem](#)

Functions

- def [AbstractItem.main](#) ()

8.2 Customer.py File Reference

Classes

- class [Customer.Customer](#)
A [Customer](#) is a client of a [VideoStore](#) who can rent items.

Namespaces

- [Customer](#)

Functions

- def [Customer.main](#) ()

8.3 GenreSearch.py File Reference

Classes

- class [GenreSearch.GenreSearch](#)
Implementation of [SearchCondition](#) that matches items based on the genre (not case sensitive).

Namespaces

- [GenreSearch](#)

8.4 Item.py File Reference

Classes

- class [Item.Item](#)
This is an abstract class.

Namespaces

- [Item](#)

Variables

- [Item.ABC](#) = object

8.5 OverDuelItemSearch.py File Reference

Classes

- class [OverDuelItemSearch.OverDuelItemSearch](#)
A class for matching items with due date before a given date.

Namespaces

- [OverDuelItemSearch](#)

8.6 SearchCondition.py File Reference

Classes

- class [SearchCondition.SearchCondition](#)
Abstraction of a search predicate for items.

Namespaces

- [SearchCondition](#)

Variables

- [SearchCondition.ABC](#) = object

8.7 SimpleDate.py File Reference

Classes

- class [SimpleDate.SimpleDate](#)
Date consisting of a year, month, and day.

Namespaces

- [SimpleDate](#)

Functions

- def [SimpleDate.main](#) ()
Main program for testing.

8.8 StatusException.py File Reference

Classes

- class [StatusException.StatusException](#)
Exception type thrown for invalid operations such as attempting to rent an item that is already rented.

Namespaces

- [StatusException](#)

8.9 VideoStore.py File Reference

Classes

- class [VideoStore.VideoStore](#)
[VideoStore](#) consists of a list of items and a list of customers who can rent items.

Namespaces

- [VideoStore](#)

Functions

- def [VideoStore.main](#) ()
Main program for testing.

Index

- `__add__`
 - `SimpleDate.SimpleDate`, [50](#)
 - `__balance`
 - `Customer.Customer`, [33](#)
 - `__barcode`
 - `AbstractItem.AbstractItem`, [26](#)
 - `__customers`
 - `VideoStore.VideoStore`, [60](#)
 - `__date`
 - `OverDuelItemSearch.OverDuelItemSearch`, [45](#)
 - `SimpleDate.SimpleDate`, [53](#)
 - `__dueDate`
 - `AbstractItem.AbstractItem`, [26](#)
 - `__eq__`
 - `AbstractItem.AbstractItem`, [22](#)
 - `SimpleDate.SimpleDate`, [50](#)
 - `__genre`
 - `AbstractItem.AbstractItem`, [26](#)
 - `GenreSearch.GenreSearch`, [37](#)
 - `__init__`
 - `AbstractItem.AbstractItem`, [21](#)
 - `Customer.Customer`, [29](#)
 - `GenreSearch.GenreSearch`, [36](#)
 - `OverDuelItemSearch.OverDuelItemSearch`, [44](#)
 - `SimpleDate.SimpleDate`, [49](#)
 - `StatusException.StatusException`, [55](#)
 - `VideoStore.VideoStore`, [58](#)
 - `__items`
 - `VideoStore.VideoStore`, [60](#)
 - `__itemsOut`
 - `Customer.Customer`, [34](#)
 - `__lt__`
 - `SimpleDate.SimpleDate`, [50](#)
 - `__metaclass__`
 - `Item.Item`, [43](#)
 - `SearchCondition.SearchCondition`, [47](#)
 - `__name`
 - `Customer.Customer`, [34](#)
 - `__rented`
 - `AbstractItem.AbstractItem`, [26](#)
 - `__repr__`
 - `AbstractItem.AbstractItem`, [22](#)
 - `Customer.Customer`, [29](#)
 - `SimpleDate.SimpleDate`, [51](#)
 - `VideoStore.VideoStore`, [58](#)
 - `__str__`
 - `AbstractItem.AbstractItem`, [23](#)
 - `Customer.Customer`, [30](#)
 - `SimpleDate.SimpleDate`, [51](#)
 - `VideoStore.VideoStore`, [58](#)
 - `__sub__`
 - `SimpleDate.SimpleDate`, [51](#)
 - `__title`
 - `AbstractItem.AbstractItem`, [27](#)
- ABC
 - `Item`, [14](#)
 - `SearchCondition`, [15](#)
- `AbstractItem`, [11](#)
 - `main`, [11](#)
- `AbstractItem.AbstractItem`, [19](#)
 - `__barcode`, [26](#)
 - `__dueDate`, [26](#)
 - `__eq__`, [22](#)
 - `__genre`, [26](#)
 - `__init__`, [21](#)
 - `__rented`, [26](#)
 - `__repr__`, [22](#)
 - `__str__`, [23](#)
 - `__title`, [27](#)
 - `getBarcode`, [23](#)
 - `getDueDate`, [23](#)
 - `getGenre`, [23](#)
 - `getTitle`, [24](#)
 - `isRented`, [24](#)
 - `setDueDate`, [24](#)
 - `setRented`, [25](#)
 - `setReturned`, [25](#)
- `AbstractItem.py`, [63](#)
- `addCustomer`
 - `VideoStore.VideoStore`, [58](#)
- `addItem`
 - `VideoStore.VideoStore`, [59](#)
- `bringBackItem`
 - `Customer.Customer`, [30](#)
- `calculateLateFee`
 - `Item.Item`, [39](#)
- `canRent`
 - `Customer.Customer`, [31](#)

- Customer, 12
 - main, 12
- Customer.Customer, 27
 - __balance, 33
 - __init__, 29
 - __itemsOut, 34
 - __name, 34
 - __repr__, 29
 - __str__, 30
 - bringBackItem, 30
 - canRent, 31
 - getBalance, 31
 - getName, 31
 - makePayment, 32
 - rentItem, 32
 - setBalance, 33
- Customer.py, 63
- daysUntil
 - SimpleDate.SimpleDate, 51
- findUser
 - VideoStore.VideoStore, 59
- GenreSearch, 13
- GenreSearch.GenreSearch, 35
 - __genre, 37
 - __init__, 36
 - matches, 36
- GenreSearch.py, 64
- getBalance
 - Customer.Customer, 31
- getBarcode
 - AbstractItem.AbstractItem, 23
 - Item.Item, 39
- getDueDate
 - AbstractItem.AbstractItem, 23
 - Item.Item, 39
- getGenre
 - AbstractItem.AbstractItem, 23
 - Item.Item, 40
- getName
 - Customer.Customer, 31
- getRentalCost
 - Item.Item, 40
- getRentalDays
 - Item.Item, 40
- getTitle
 - AbstractItem.AbstractItem, 24
 - Item.Item, 41
- isBefore
 - SimpleDate.SimpleDate, 52
- isRented
 - AbstractItem.AbstractItem, 24
 - Item.Item, 41
- Item, 13
 - ABC, 14
- Item.Item, 37
 - __metaclass__, 43
 - calculateLateFee, 39
 - getBarcode, 39
 - getDueDate, 39
 - getGenre, 40
 - getRentalCost, 40
 - getRentalDays, 40
 - getTitle, 41
 - isRented, 41
 - setDueDate, 41
 - setRented, 42
 - setReturned, 42
- Item.py, 64
- main
 - AbstractItem, 11
 - Customer, 12
 - SimpleDate, 16
 - VideoStore, 17
- makePayment
 - Customer.Customer, 32
- matches
 - GenreSearch.GenreSearch, 36
 - OverDuelItemSearch.OverDuelItemSearch, 45
 - SearchCondition.SearchCondition, 47
- MILLIS_IN_24_HOURS
 - SimpleDate.SimpleDate, 54
- msg
 - StatusException.StatusException, 56
- OverDuelItemSearch, 14
- OverDuelItemSearch.OverDuelItemSearch, 43
 - __date, 45
 - __init__, 44
 - matches, 45
- OverDuelItemSearch.py, 64
- rentItem
 - Customer.Customer, 32
- search
 - VideoStore.VideoStore, 60
- SearchCondition, 14
 - ABC, 15
- SearchCondition.py, 65
- SearchCondition.SearchCondition, 46
 - __metaclass__, 47
 - matches, 47
- setBalance
 - Customer.Customer, 33
- setDueDate

- AbstractItem.AbstractItem, 24
- Item.Item, 41
- setRented
 - AbstractItem.AbstractItem, 25
 - Item.Item, 42
- setReturned
 - AbstractItem.AbstractItem, 25
 - Item.Item, 42
- SimpleDate, 15
 - main, 16
- SimpleDate.py, 65
- SimpleDate.SimpleDate, 48
 - __add__, 50
 - __date__, 53
 - __eq__, 50
 - __init__, 49
 - __lt__, 50
 - __repr__, 51
 - __str__, 51
 - __sub__, 51
 - daysUntil, 51
 - isBefore, 52
 - MILLIS_IN_24_HOURS, 54
 - SimpleDateFromDays, 52
 - today, 53
 - todayNow, 53
- SimpleDateFromDays
 - SimpleDate.SimpleDate, 52
- StatusException, 16
- StatusException.py, 65
- StatusException.StatusException, 54
 - __init__, 55
 - msg, 56
- today
 - SimpleDate.SimpleDate, 53
- todayNow
 - SimpleDate.SimpleDate, 53
- VideoStore, 16
 - main, 17
- VideoStore.py, 66
- VideoStore.VideoStore, 56
 - __customers__, 60
 - __init__, 58
 - __items__, 60
 - __repr__, 58
 - __str__, 58
 - addCustomer, 58
 - addItem, 59
 - findUser, 59
 - search, 60