

The post card list project

Generated by Doxygen 1.8.15

1 Python Assignment	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 Namespace Documentation	9
5.1 exam_requests Namespace Reference	9
5.1.1 Detailed Description	9
5.2 exam_solution.PostcardList Namespace Reference	9
5.2.1 Detailed Description	10
6 Class Documentation	11
6.1 exam_requests.PostcardList Class Reference	11
6.2 exam_solution.PostcardList.PostcardList Class Reference	11
6.2.1 Detailed Description	11
6.2.2 Constructor & Destructor Documentation	11
6.2.2.1 __init__()	12
6.2.3 Member Function Documentation	12
6.2.3.1 add_to_list()	12
6.2.3.2 getNumberOfPostcards()	12
6.2.3.3 getPostcardsByDateRange()	12
6.2.3.4 getPostcardsByReceiver()	13
6.2.3.5 getPostcardsBySender()	13
6.2.3.6 parsePostcards()	13
6.2.3.7 readFile()	13
6.2.3.8 updateFile()	14
6.2.3.9 updateLists()	14
6.2.3.10 writeFile()	14
6.3 exam_requests.Test Class Reference	15
Index	17

Chapter 1

Python Assignment

Herbert Nguruwe, Tommaso Ronconi

Project Structure: Doxygen - contains the [documentation](#) of the PostcardList. [exam_↔ solution](exam_solution) : contains the actual [PostcardList](#) module. [exam_requests.py](#): imports the PostcardList and run tests against.

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

exam_requests	9
exam_solution.PostcardList	9

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

exam_requests.PostcardList	11
exam_solution.PostcardList.PostcardList	11
TestCase	
exam_requests.Test	15

Chapter 4

Class Index

4.1 Class List

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

exam_requests.PostcardList	11
exam_solution.PostcardList.PostcardList	11
exam_requests.Test	15

Chapter 5

Namespace Documentation

5.1 exam_requests Namespace Reference

Classes

- class [PostcardList](#)
- class [Test](#)

5.1.1 Detailed Description

Exam requests:

1. Write a class `PostcardList` that satisfies the tests (defined using `unittest`).

SPECIFICATIONS:

Class `PostcardList` reads/writes Postcard messages from/to a properly formatted file: each record (i.e., file's line) is a Postcard message. The Postcard format is "date:\$(DATE); from:\$(SENDER); to:\$(RECEIVER);" (e.g., "date:2009-12-24; from:Daisy; to:John;").

Class `PostcardList` must have at least these attributes:

- `_file`: file name, eventually with the full path.
- `_postcards`: list of postcards read from `_file`.
- `_date`: is a dict where the key is the string date, and the value is a list of indices. Each index refers to a postcard.
- `_from`: is a dict where the key is the string sender, and the value is a list of indices. Each index refers to a postcard.
- `_to`: is a dict where the key is the string receiver, and the value is a list of indices. Each index refers to a postcard.

Class `PostcardList` must manage the I/O file through the following member functions. Note that you are free to use any other functions.

- `writeFile(self,...)`: write `self.{_date,_from,_to}` to `self._file`
- `readFile(self,...)`: from `self._file` read `self.{_date,_from,_to}`
- `parsePostcards(self,...)`: parse `self._postcards`, set `self.{_date,_from,_to}`
- `updateFile(self,...)`: as write but appending to `self._file`
- `updateLists(self,...)`: as read but appending to `self._postcards`
- `getNumberOfPostcards(self)`: returns length of `self._postcards`

Class `PostcardList` must manage the sorting of dates/senders/receivers. Note that the names and arguments of the following functions are defined in the tests.

- `getPostcardsByDateRange(self,date_range)`: returns the postcards within a `date_range`
- `getPostcardsBySender(self, sender)`: returns the postcards from a sender
- `getPostcardsByReceiver(self, receiver)`: returns the postcards to a receiver

5.2 exam_solution.PostcardList Namespace Reference

Classes

- class [PostcardList](#)

5.2.1 Detailed Description

@author Herbert Nguruwe and Tommaso Ronconi

Intresting notes:

Initially the class was implemented using the defaultdict class from collections package. This caused problems with the method test (test_check_sent_received_when) which is implemented using try-catch statement, so defaultdict doesn't return an error when trying to access an item that is not there it creates a new item. In this case it would create an empty list rather than throwing an exception.

All other functions passed the test. Defaultdict takes a type to initialize as a default type.

Chapter 6

Class Documentation

6.1 exam_requests.PostcardList Class Reference

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

- exam_requests.py

6.2 exam_solution.PostcardList.PostcardList Class Reference

Public Member Functions

- def `__init__` (self)
- def `writeFile` (self, _file)
- def `readFile` (self, _file)
- def `parsePostcards` (self, _postcard)
- def `updateFile` (self)
- def `updateLists` (self, _file)
- def `add_to_list` (self, _dict, _key, _index)
- def `getNumberOfPostcards` (self)
- def `getPostcardsByDateRange` (self, date_range)
- def `getPostcardsBySender` (self, sender)
- def `getPostcardsByReceiver` (self, receiver)

6.2.1 Detailed Description

```
class manages a list of post cards, that are added from a file.It
reads a file where each line represents a postcard with to, from,
and date when the post card was sent.
```

6.2.2 Constructor & Destructor Documentation

6.2.2.1 `__init__()`

```
def exam_solution.PostcardList.PostcardList.__init__ (
    self )

"Initialize a tree with all arguments to default
value. We decided to create an object without a file to
read from but use the methods provided by class. This is
design passed by the class.
```

6.2.3 Member Function Documentation

6.2.3.1 `add_to_list()`

```
def exam_solution.PostcardList.PostcardList.add_to_list (
    self,
    _dict,
    _key,
    _index )

@param _dict: dictionary to be added
@param _key: key to be used
@param _index: The index of the record
Function used to add an item to the dictionary
```

6.2.3.2 `getNumberOfPostcards()`

```
def exam_solution.PostcardList.PostcardList.getNumberOfPostcards (
    self )

returns the number of postcards
@return int: Number of postcards in the self._postcards
```

6.2.3.3 `getPostcardsByDateRange()`

```
def exam_solution.PostcardList.PostcardList.getPostcardsByDateRange (
    self,
    date_range )

returns the postcards within a date_range
@param date_range : search for all the postcards
that were within the range that was passed.
@return: list of postcards sent within the range.
```


6.2.3.4 getPostcardsByReceiver()

```
def exam_solution.PostcardList.PostcardList.getPostcardsByReceiver (
    self,
    receiver )
```

get postcards send to a receiver
@param receiver: search for all the postcards
that were sent by Sender
@return: list of postcards send by the sender.

6.2.3.5 getPostcardsBySender()

```
def exam_solution.PostcardList.PostcardList.getPostcardsBySender (
    self,
    sender )
```

returns the postcards from a sender.
@param sender: search for all the postcards
that were sent by sender
@return: list of postcards send by the sender.

6.2.3.6 parsePostcards()

```
def exam_solution.PostcardList.PostcardList.parsePostcards (
    self,
    _postcard )
```

@param _postcard: list of postcards in the case object's.
parse self._postcards, set self.{_date,_from,_to}
and return to the calling function

6.2.3.7 readFile()

```
def exam_solution.PostcardList.PostcardList.readFile (
    self,
    _file )
```

Read from a file to self._postcardlist.
@param _file: inputfile to be written to using data that from
object's _postcard's list member. Call the update list function

6.2.3.8 updateFile()

```
def exam_solution.PostcardList.PostcardList.updateFile (
    self )
```

Append to a self.{_date,_from,_to} to self._file.
@param _file: outputfile to be updated using data from
object's _postcard's list member.

6.2.3.9 updateLists()

```
def exam_solution.PostcardList.PostcardList.updateLists (
    self,
    _file )
```

Update the object's _postcardlist with
postcards from the passed file.
@param _file: the file to be read from to
update the _postcardlist.

6.2.3.10 writeFile()

```
def exam_solution.PostcardList.PostcardList.writeFile (
    self,
    _file )
```

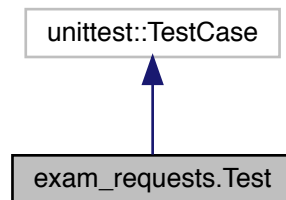
write to a new file self.{_date,_from,_to} to self._file.
@param _file: inputfile to be written to using data from
object's _postcard's list member.

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

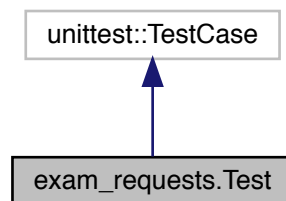
- exam_solution/PostcardList.py

6.3 exam_requests.Test Class Reference

Inheritance diagram for exam_requests.Test:



Collaboration diagram for exam_requests.Test:



Public Member Functions

- def **setUpClass** (self)
- def **test_missing_attributes** (self)
- def **test_check_getPostcardByDateRange** (self)
- def **test_check_getPostcardBySender** (self)
- def **test_check_getPostcardByReceiver** (self)
- def **test_check_sent_received_when** (self)

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

- exam_requests.py

Index

- `__init__`
 - `exam_solution.PostcardList.PostcardList`, [11](#)
- `add_to_list`
 - `exam_solution.PostcardList.PostcardList`, [12](#)
- `exam_requests`, [9](#)
- `exam_requests.PostcardList`, [11](#)
- `exam_requests.Test`, [15](#)
- `exam_solution.PostcardList`, [9](#)
- `exam_solution.PostcardList.PostcardList`, [11](#)
 - `__init__`, [11](#)
 - `add_to_list`, [12](#)
 - `getNumberOfPostcards`, [12](#)
 - `getPostcardsByDateRange`, [12](#)
 - `getPostcardsByReceiver`, [12](#)
 - `getPostcardsBySender`, [13](#)
 - `parsePostcards`, [13](#)
 - `readFile`, [13](#)
 - `updateFile`, [13](#)
 - `updateLists`, [14](#)
 - `writeFile`, [14](#)
- `getNumberOfPostcards`
 - `exam_solution.PostcardList.PostcardList`, [12](#)
- `getPostcardsByDateRange`
 - `exam_solution.PostcardList.PostcardList`, [12](#)
- `getPostcardsByReceiver`
 - `exam_solution.PostcardList.PostcardList`, [12](#)
- `getPostcardsBySender`
 - `exam_solution.PostcardList.PostcardList`, [13](#)
- `parsePostcards`
 - `exam_solution.PostcardList.PostcardList`, [13](#)
- `readFile`
 - `exam_solution.PostcardList.PostcardList`, [13](#)
- `updateFile`
 - `exam_solution.PostcardList.PostcardList`, [13](#)
- `updateLists`
 - `exam_solution.PostcardList.PostcardList`, [14](#)
- `writeFile`
 - `exam_solution.PostcardList.PostcardList`, [14](#)