C++ Softwrae Transactional memory

Zoltan Fuzesi

Contents

Chapter 1

OSTM C++ Software Transactional Memory

1.1 Object Based Software Transactional Memory.

OSTM (p. 30) is a polymorphic solution to store and manage shared memory spaces within c++ programming context.
You can store and managed any kind of object in transactional environment as a shared and protected memory space.

1.1.1 Brief. Download the zip file from the provided link in the web-site, that contains the libostm.so, TM.h, TX.h, OSTM.h files.

Unzip the archive file to the desired destination possibly where in you program is stored.

1.1.2 Step 1: Download the archive file.

1.1.3 Step 2: Unzip in the target destination.

1.1.4 Step 3: Copy the shared library (libostm.so) to the operating system folder where the other shared library are stored.

It will be different destination folder on different platforms. (Linux, Windows, Mac OS) `More Information`

1.1.5 Step 4: Achieve the required class hierarchy between the OSTM library and your own class structure.

Details and instruction of class hierarchy requirements can be found on the web-site. www.serversite.info/ostm

1.1.6 Step 5: Create an executable file as you linking together the TM.h, TX.h, OSTM.h files with your own files.

1.1.7 Step 6: Now your application use transactional environment, that guarantees the consistency between object transactions.

1.1.8 Step 7: Run the application.

Abbreviation for bank names used in the test cases:
BOA (p. 13) - Bank of America
ULSTER (p. 42) - Ulster Bank
UNBL (p. 45) - United National Bank Limited
SWBPLC (p. 34) - Scottish Windows Bank PLC
AIB (p. 9) - Allied Irish Bank
BOI (p. 16) - Bank of Ireland

Chapter 2

README

threadArraySize = 1 - 300 Nested transaction Test with any number of threads
Function *nesting* is create a transaction and transfer between two objects
Then create a Transaction inside the transaction, (nested Transaction)
and inside the nested transaction invokes the *two_account_transfer* transaction, that will be the third level inner transaction.

TEST details: Used values :
Tanfer amount used in the transactions :
∗transferAmount = 1
Number of threads used in the test application :
∗threadArraySize = 1 -10 - 300 specified in every test

Six **OSTM** (p. 30) type smart pointer aib_ptr, boi_ptr, boa_ptr, swplc_ptr, ulster_ptr, unbl_ptr , these objects are inherites from **BANK** (p. 12) parent class.
Every thread used two object in the transaction within the *nesting* function, and creates a nested transaction using the same objects.
Inside the nested transaction makes a function call to the *two_account_transfer* function, which is creates a transaction and a nested transaction in the nested transaction.
So, every thread used the objects in four level deep transactions.
To make easyer to folow the calculation the transactions are using the objects in the same transfering oder, from-to.

Test 1: using only 1 Thread with two objetcs aib_ptr, boi_ptr and tranfering 1 unit in every transaction :
thArray[i] = std::thread(*nesting*, aib_ptr, boi_ptr, std::ref(tm), transferAmount);
Start values : Double value : 500 to each.
Nesting level 4.
After transactions :

- To account : aib_ptr - Double value : 504

- From account : boi_ptr - Double value : 496

- version number 1. for all object used in transaction

  Test 2: using 10 Thread with two objetcs aib_ptr, boi_ptr and tranfering 1 unit in every transaction ←-
  :
  Start values : Double value : 500 to each.
  Nesting level 4.
  After transactions :

- To account : aib_ptr - Double value : 540
- From account : boi_ptr - Double value : 460
- version number 10. for all object used in transaction

  Test 3: using 300 Thread with six objetcs aib_ptr, boi_ptr, boa_ptr, swplc_ptr, ulster_ptr, unbl_ptr and tranfering 1 unit in every transaction :
  Every object will be used by 100 threads.
  Start values : Double value : 500 to each.
  Nesting level 4.
  After transactions :

- To account : aib_ptr - Double value : 900
- From account : boi_ptr - Double value : 100
- To account : boa_ptr - Double value : 900
- From account : swplc_ptr - Double value : 100
- To account : ulster_ptr - Double value : 900
- From account : unbl_ptr - Double value : 100
- version number 100. for all object used in transaction

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

Chapter 4

Class Index

4.1 Class List

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

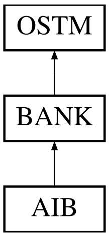
Chapter 5

Class Documentation

5.1 AIB Class Reference

```
#include <AIB.h>
```

Inheritance diagram for AIB:



Public Member Functions

- **AIB** ()
- **AIB** (int accountNumber, double balance, std::string firstName, std::string lastName, std::string address)
- **AIB** (std::shared_ptr< **BANK** > obj, int _version, int _unique_id)
- **AIB** (const **AIB** &orig)
- **AIB operator=** (const **AIB** &orig)
- virtual ∼**AIB** ()
- virtual void **copy** (std::shared_ptr< **OSTM** > to, std::shared_ptr< **OSTM** > from)

  copy function, make deep copy of the object/pointer
- virtual std::shared_ptr< **OSTM** > **getBaseCopy** (std::shared_ptr< **OSTM** > object)

  getBaseCopy function, make deep copy of the object/pointer and Return a new std::shared_ptr<BANK> type object
- virtual void **toString** ()

  _cast, is use to cast bak the std::shared_ptr<OSTM> to the required type
- virtual void **SetAddress** (std::string address)
- virtual std::string **GetAddress** () const
- virtual void **SetBalance** (double balance)
- virtual double **GetBalance** () const
- virtual void **SetAccountNumber** (int accountNumber)
- virtual int **GetAccountNumber** () const
- virtual void **SetLastName** (std::string lastName)
- virtual std::string **GetLastName** () const
- virtual void **SetFirstName** (std::string firstName)
- virtual std::string **GetFirstName** () const
- virtual void **SetFullname** (std::string fullname)
- virtual std::string **GetFullname** () const

5.1.1 Detailed Description

Inherit from **BANK** (p. 12)

Definition at line 18 of file AIB.h.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 AIB::AIB ( ) `[inline]`

Constructor

Definition at line 23 of file AIB.h.

5.1.2.2 AIB::AIB (int *accountNumber,* double *balance,* std::string *firstName,* std::string *lastName,* std::string *address*) `[inline]`

Custom constructor

Definition at line 36 of file AIB.h.

5.1.2.3 AIB::AIB (std::shared_ptr< BANK > *obj,* int *_version,* int *_unique_id*) `[inline]`

Custom constructor, used by the library for deep copying

Definition at line 48 of file AIB.h.

5.1.2.4 AIB::AIB (const AIB & *orig*)

Copy constructor

Definition at line 14 of file AIB.cpp.

5.1.2.5 AIB::∼AIB ( ) `[virtual]`

de-constructor

Definition at line 17 of file AIB.cpp.

5.1.3 Member Function Documentation

5.1.3.1 void AIB::copy (std::shared_ptr< OSTM > *to,* std::shared_ptr< OSTM > *from*) `[virtual]`

copy function, make deep copy of the object/pointer

Parameters

objTO	is a std::shared_ptr<BANK> type object casted back from std::shared_ptr<OSTM>
objFROM	is a std::shared_ptr<BANK> type object casted back from std::shared_ptr<OSTM>

Reimplemented from **OSTM** (p. 30).

Definition at line 37 of file AIB.cpp.

5.1.3.2 std::shared_ptr< OSTM > AIB::getBaseCopy (std::shared_ptr< OSTM > *object*) [virtual]

getBaseCopy function, make deep copy of the object/pointer and Return a new std::shared_ptr<BANK> type object

Parameters

objTO	is a **BANK** (p. 12) type pointer for casting
obj	is a std::shared_ptr<BANK> return type

Reimplemented from **OSTM** (p. 30).

Definition at line 24 of file AIB.cpp.

5.1.3.3 AIB AIB::operator= (const AIB & *orig*) [inline]

Operator

Definition at line 66 of file AIB.h.

5.1.3.4 void AIB::toString ( ) [virtual]

_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type

toString function, displays the object values in formatted way

Reimplemented from **OSTM** (p. 30).

Definition at line 56 of file AIB.cpp.

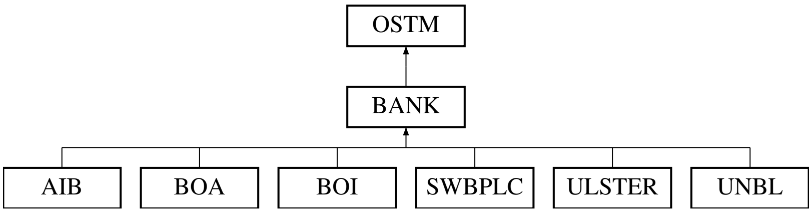
The documentation for this class was generated from the following files:

- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/↩ Test01/AIB.h
- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/↩ Test01/AIB.cpp

5.2 BANK Class Reference

```
#include <BANK.h>
```

Inheritance diagram for BANK:



Public Member Functions

- **BANK** ()
- **BANK** (int _version, int _unique_id)
- **BANK** (const **BANK** &orig)
- virtual ∼**BANK** ()
- virtual void **SetAddress** (std::string address)
- virtual std::string **GetAddress** () const
- virtual void **SetBalance** (double balance)
- virtual double **GetBalance** () const
- virtual void **SetAccountNumber** (int accountNumber)
- virtual int **GetAccountNumber** () const
- virtual void **SetLastName** (std::string lastName)
- virtual std::string **GetLastName** () const
- virtual void **SetFirstName** (std::string firstName)
- virtual std::string **GetFirstName** () const
- virtual void **SetFullname** (std::string fullname)
- virtual std::string **GetFullname** () const

5.2.1 Detailed Description

BANK (p. 12) inherit from the **OSTM** (p. 30) library. It is declares the common functions in the child classes as a virtual function.

Definition at line 16 of file BANK.h.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 BANK::BANK ( ) [inline]

Constructor

Definition at line 23 of file BANK.h.

5.2.2.2 BANK::BANK (int *_version,* int *_unique_id*) [inline]

Custom Constructor

Definition at line 29 of file BANK.h.

5.2.2.3 BANK::BANK (const BANK & *orig*)

Copy constructor

Definition at line 11 of file BANK.cpp.

5.2.2.4 BANK::∼BANK ( ) [virtual]

de-constructor

Definition at line 14 of file BANK.cpp.

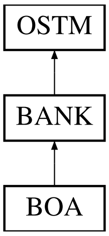
The documentation for this class was generated from the following files:

- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/←↩ Test01/BANK.h
- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/←↩ Test01/BANK.cpp

5.3 BOA Class Reference

```
#include <BOA.h>
```

Inheritance diagram for BOA:



Public Member Functions

- **BOA** ()
- **BOA** (int accountNumber, double balance, std::string firstName, std::string lastName, std::string address)
- **BOA** (std::shared_ptr< **BANK** > obj, int _version, int _unique_id)
- **BOA** (const **BOA** &orig)
- **BOA operator=** (const **BOA** &orig)
- virtual ∼**BOA** ()
- virtual void **copy** (std::shared_ptr< **OSTM** > to, std::shared_ptr< **OSTM** > from)

  copy function, make deep copy of the object/pointer
- virtual std::shared_ptr< **OSTM** > **getBaseCopy** (std::shared_ptr< **OSTM** > object)

  getBaseCopy function, make deep copy of the object/pointer and Return a new std::shared_ptr<BANK> type object
- virtual void **toString** ()

  _cast, is use to cast bak the std::shared_ptr<OSTM> to the required type
- virtual void **SetAddress** (std::string address)
- virtual std::string **GetAddress** () const
- virtual void **SetBalance** (double balance)
- virtual double **GetBalance** () const
- virtual void **SetAccountNumber** (int accountNumber)
- virtual int **GetAccountNumber** () const
- virtual void **SetLastName** (std::string lastName)
- virtual std::string **GetLastName** () const
- virtual void **SetFirstName** (std::string firstName)
- virtual std::string **GetFirstName** () const
- virtual void **SetFullname** (std::string fullname)
- virtual std::string **GetFullname** () const

5.3.1 Detailed Description

Inherit from **BANK** (p. 12)

Definition at line 18 of file BOA.h.

5.3.2 Constructor & Destructor Documentation

5.3.2.1 BOA::BOA ( ) `[inline]`

Constructor

Definition at line 24 of file BOA.h.

5.3.2.2 BOA::BOA (int *accountNumber,* double *balance,* std::string *firstName,* std::string *lastName,* std::string *address*) `[inline]`

Custom constructor

Definition at line 35 of file BOA.h.

5.3.2.3 BOA::BOA (std::shared_ptr< **BANK** > *obj,* int *_version,* int *_unique_id*) [inline]

Custom constructor, used by the library for deep copying

Definition at line 46 of file BOA.h.

5.3.2.4 BOA::BOA (const **BOA** & *orig*)

Copy constructor

Definition at line 12 of file BOA.cpp.

5.3.2.5 BOA::∼BOA ( ) [virtual]

de-constructor

Definition at line 15 of file BOA.cpp.

5.3.3 Member Function Documentation

5.3.3.1 void BOA::copy (std::shared_ptr< **OSTM** > *to,* std::shared_ptr< **OSTM** > *from*) [virtual]

copy function, make deep copy of the object/pointer

Parameters

objTO	is a std::shared_ptr<BANK> type object casted back from std::shared_ptr<OSTM>
objFROM	is a std::shared_ptr<BANK> type object casted back from std::shared_ptr<OSTM>

Reimplemented from **OSTM** (p. 30).

Definition at line 34 of file BOA.cpp.

5.3.3.2 std::shared_ptr< **OSTM** > BOA::getBaseCopy (std::shared_ptr< **OSTM** > *object*) [virtual]

getBaseCopy function, make deep copy of the object/pointer and Return a new std::shared_ptr<BANK> type object

Parameters

objTO	is a **BANK** (p. 12) type pointer for casting
obj	is a std::shared_ptr<BANK> return type

Reimplemented from **OSTM** (p. 30).

Definition at line 22 of file BOA.cpp.

5.3.3.3 BOA BOA::operator= (const BOA & *orig*) [inline]

Operator

Definition at line 64 of file BOA.h.

5.3.3.4 void BOA::toString ( ) [virtual]

_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type

toString function, displays the object values in formatted way

Reimplemented from **OSTM** (p. 30).

Definition at line 54 of file BOA.cpp.

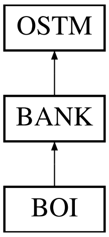
The documentation for this class was generated from the following files:

- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/↩ Test01/BOA.h
- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/↩ Test01/BOA.cpp

5.4 BOI Class Reference

```
#include <BOI.h>
```

Inheritance diagram for BOI:



Public Member Functions

- **BOI** ()
- **BOI** (int accountNumber, double balance, std::string firstName, std::string lastName, std::string address)
- **BOI** (std::shared_ptr< **BOI** > obj, int _version, int _unique_id)
- **BOI** (const **BOI** &orig)
- **BOI operator=** (const **BOI** &orig)
- virtual ∼**BOI** ()
- virtual std::shared_ptr< **OSTM** > **getBaseCopy** (std::shared_ptr< **OSTM** > object)

  getBaseCopy function, make deep copy of the object/pointer and Return a new BANK∗ type object
- virtual void **copy** (std::shared_ptr< **OSTM** > to, std::shared_ptr< **OSTM** > from)

  copy function, make deep copy of the object/pointer

- virtual void **toString** ()

  _cast, is use to cast bak the std::shared_ptr< OSTM > to the required type
- virtual void **SetAddress** (std::string address)
- virtual std::string **GetAddress** () const
- virtual void **SetBalance** (double balance)
- virtual double **GetBalance** () const
- virtual void **SetAccountNumber** (int accountNumber)
- virtual int **GetAccountNumber** () const
- virtual void **SetLastName** (std::string lastName)
- virtual std::string **GetLastName** () const
- virtual void **SetFirstName** (std::string firstName)
- virtual std::string **GetFirstName** () const
- virtual void **SetFullname** (std::string fullname)
- virtual std::string **GetFullname** () const

5.4.1 Detailed Description

Inherit from **BANK** (p. 12)

Definition at line 19 of file BOI.h.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 BOI::BOI ( ) [inline]

Constructor

Definition at line 24 of file BOI.h.

5.4.2.2 BOI::BOI (int *accountNumber,* double *balance,* std::string *firstName,* std::string *lastName,* std::string *address*) [inline]

Custom constructor

Definition at line 37 of file BOI.h.

5.4.2.3 BOI::BOI (std::shared_ptr< BOI > *obj,* int *_version,* int *_unique_id*) [inline]

Custom constructor, used by the library for deep copying

Definition at line 49 of file BOI.h.

5.4.2.4 BOI::BOI (const BOI & *orig*)

Copy constructor

Definition at line 15 of file BOI.cpp.

5.4.2.5 BOI::∼BOI ( ) [virtual]

de-constructor

Definition at line 12 of file BOI.cpp.

5.4.3 Member Function Documentation

5.4.3.1 void BOI::copy (std::shared_ptr< OSTM > *to,* std::shared_ptr< OSTM > *from*) [virtual]

copy function, make deep copy of the object/pointer

Parameters

objTO	is a BANK∗ type object casted back from std::shared_ptr<OSTM>
objFROM	is a BANK∗ type object casted back from std::shared_ptr<OSTM>

Reimplemented from **OSTM** (p. 30).

Definition at line 35 of file BOI.cpp.

5.4.3.2 std::shared_ptr< OSTM > BOI::getBaseCopy (std::shared_ptr< OSTM > *object*) [virtual]

getBaseCopy function, make deep copy of the object/pointer and Return a new BANK∗ type object

Parameters

objTO	is a **BANK** (p. 12) type pointer for casting
obj	is a BANK∗ return type

Reimplemented from **OSTM** (p. 30).

Definition at line 22 of file BOI.cpp.

5.4.3.3 BOI BOI::operator= (const BOI & *orig*) [inline]

Operator

Definition at line 65 of file BOI.h.

5.4.3.4 void BOI::toString ( ) [virtual]

_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type

toString function, displays the object values in formatted way

Reimplemented from **OSTM** (p. 30).

Definition at line 54 of file BOI.cpp.

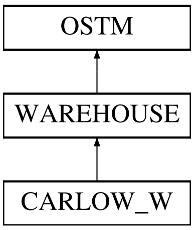
The documentation for this class was generated from the following files:

- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/↩ Test01/BOI.h
- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/↩ Test01/BOI.cpp

5.5 CARLOW_W Class Reference

```
#include <CARLOW_W.h>
```

Inheritance diagram for CARLOW_W:



Public Member Functions

- **CARLOW_W** ()
- **CARLOW_W** (std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)
- **CARLOW_W** (std::shared_ptr< **WAREHOUSE** > obj, int _version, int _unique_id)
- **CARLOW_W** (const **CARLOW_W** &orig)
- **CARLOW_W operator=** (const **CARLOW_W** &orig)
- virtual ∼**CARLOW_W** ()
- virtual void **copy** (std::shared_ptr< **OSTM** > to, std::shared_ptr< **OSTM** > from)

  copy function, make deep copy of the object/pointer
- virtual std::shared_ptr< **OSTM** > **getBaseCopy** (std::shared_ptr< **OSTM** > object)

  getBaseCopy function, make deep copy of the object/pointer and Return a new BANK∗ type object
- virtual void **toString** ()

  _cast, is use to cast bak the std::shared_ptr<OSTM> to the required type
- virtual void **SetNumber_of_alcatel** (int _number_of_alcatel)
- virtual int **GetNumber_of_alcatel** ()
- virtual void **SetNumber_of_nokia** (int _number_of_nokia)
- virtual int **GetNumber_of_nokia** ()
- virtual void **SetNumber_of_huawei** (int _number_of_huawei)
- virtual int **GetNumber_of_huawei** ()
- virtual void **SetNumber_of_sony** (int _number_of_sony)
- virtual int **GetNumber_of_sony** ()
- virtual void **SetNumber_of_samsung** (int _number_of_samsung)
- virtual int **GetNumber_of_samsung** ()
- virtual void **SetNumber_of_iphones** (int _number_of_iphones)
- virtual int **GetNumber_of_iphones** ()
- virtual void **SetShop_name** (std::string _shop_name)
- virtual std::string **GetShop_name** ()
- virtual void **SetShop_address** (std::string _shop_address)
- virtual std::string **GetShop_address** ()

5.5.1 Detailed Description

Inherit from **WAREHOUSE** (p. 47)

Definition at line 19 of file CARLOW_W.h.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 CARLOW_W::CARLOW_W ( ) [inline]

Constructor

Definition at line 24 of file CARLOW_W.h.

5.5.2.2 CARLOW_W::CARLOW_W (std::string *address,* std::string *shop_name,* int *iphone,* int *samsung,* int *sony,* int *huawei,* int *nokia,* int *alcatel*) [inline]

Custom constructor

Definition at line 38 of file CARLOW_W.h.

5.5.2.3 CARLOW_W::CARLOW_W (std::shared_ptr< WAREHOUSE > *obj,* int *_version,* int *_unique_id*) [inline]

Custom constructor, used by the library for deep copying

Definition at line 55 of file CARLOW_W.h.

5.5.2.4 CARLOW_W::CARLOW_W (const CARLOW_W & *orig*)

Copy constructor

Definition at line 17 of file CARLOW_W.cpp.

5.5.2.5 CARLOW_W::∼CARLOW_W ( ) [virtual]

de-constructor

Definition at line 14 of file CARLOW_W.cpp.

5.5.3 Member Function Documentation

5.5.3.1 void CARLOW_W::copy (std::shared_ptr< OSTM > *to,* std::shared_ptr< OSTM > *from*) [virtual]

copy function, make deep copy of the object/pointer

Parameters

objTO	is a BANK∗ type object casted back from std::shared_ptr<OSTM>
objFROM	is a BANK∗ type object casted back from std::shared_ptr<OSTM>

Reimplemented from **OSTM** (p. 30).

Definition at line 37 of file CARLOW_W.cpp.

5.5.3.2 std::shared_ptr< OSTM > CARLOW_W::getBaseCopy (std::shared_ptr< OSTM > *object*) [virtual]

getBaseCopy function, make deep copy of the object/pointer and Return a new BANK∗ type object

Parameters

objTO	is a **BANK** (p. 12) type pointer for casting
obj	is a BANK∗ return type

Reimplemented from **OSTM** (p. 30).

Definition at line 24 of file CARLOW_W.cpp.

5.5.3.3 CARLOW_W CARLOW_W::operator= (const CARLOW_W & *orig*) [inline]

Operator

Definition at line 75 of file CARLOW_W.h.

5.5.3.4 void CARLOW_W::toString ( ) [virtual]

_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type

toString function, displays the object values in formatted way

Reimplemented from **OSTM** (p. 30).

Definition at line 64 of file CARLOW_W.cpp.

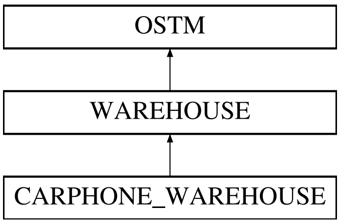
The documentation for this class was generated from the following files:

- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/↩ Test01/CARLOW_W.h
- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/↩ Test01/CARLOW_W.cpp

5.6 CARPHONE_WAREHOUSE Class Reference

```
#include <CARPHONE_WAREHOUSE.h>
```

Inheritance diagram for CARPHONE_WAREHOUSE:



Public Member Functions

- **CARPHONE_WAREHOUSE** ()
- **CARPHONE_WAREHOUSE** (std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)
- **CARPHONE_WAREHOUSE** (std::shared_ptr< **WAREHOUSE** > obj, int _version, int _unique_id)
- **CARPHONE_WAREHOUSE** (const **CARPHONE_WAREHOUSE** &orig)
- **CARPHONE_WAREHOUSE operator=** (const **CARPHONE_WAREHOUSE** &orig)
- virtual ∼**CARPHONE_WAREHOUSE** ()
- virtual void **copy** (std::shared_ptr< **OSTM** > to, std::shared_ptr< **OSTM** > from)

  copy function, make deep copy of the object/pointer
- virtual std::shared_ptr< **OSTM** > **getBaseCopy** (std::shared_ptr< **OSTM** > object)

  getBaseCopy function, make deep copy of the object/pointer and Return a new BANK∗ type object
- virtual void **toString** ()

  _cast, is use to cast bak the std::shared_ptr<OSTM> to the required type
- virtual void **SetNumber_of_alcatel** (int _number_of_alcatel)
- virtual int **GetNumber_of_alcatel** ()
- virtual void **SetNumber_of_nokia** (int _number_of_nokia)
- virtual int **GetNumber_of_nokia** ()
- virtual void **SetNumber_of_huawei** (int _number_of_huawei)
- virtual int **GetNumber_of_huawei** ()
- virtual void **SetNumber_of_sony** (int _number_of_sony)
- virtual int **GetNumber_of_sony** ()
- virtual void **SetNumber_of_samsung** (int _number_of_samsung)
- virtual int **GetNumber_of_samsung** ()
- virtual void **SetNumber_of_iphones** (int _number_of_iphones)
- virtual int **GetNumber_of_iphones** ()
- virtual void **SetShop_name** (std::string _shop_name)
- virtual std::string **GetShop_name** ()
- virtual void **SetShop_address** (std::string _shop_address)
- virtual std::string **GetShop_address** ()

5.6.1 Detailed Description

Inherit from **WAREHOUSE** (p. 47)

Definition at line 19 of file CARPHONE_WAREHOUSE.h.

5.6.2 Constructor & Destructor Documentation

5.6.2.1 CARPHONE_WAREHOUSE::CARPHONE_WAREHOUSE ( ) [inline]

Constructor

Definition at line 24 of file CARPHONE_WAREHOUSE.h.

5.6.2.2 CARPHONE_WAREHOUSE::CARPHONE_WAREHOUSE (std::string *address,* std::string *shop_name,* int *iphone,* int *samsung,* int *sony,* int *huawei,* int *nokia,* int *alcatel*) [inline]

Custom constructor

Definition at line 38 of file CARPHONE_WAREHOUSE.h.

5.6.2.3 CARPHONE_WAREHOUSE::CARPHONE_WAREHOUSE (std::shared_ptr< WAREHOUSE > *obj,* int *_version,* int *_unique_id*) [inline]

Custom constructor, used by the library for deep copying

Definition at line 55 of file CARPHONE_WAREHOUSE.h.

5.6.2.4 CARPHONE_WAREHOUSE::CARPHONE_WAREHOUSE (const CARPHONE_WAREHOUSE & *orig*)

Copy constructor

Definition at line 11 of file CARPHONE_WAREHOUSE.cpp.

5.6.2.5 CARPHONE_WAREHOUSE::∼CARPHONE_WAREHOUSE ( ) [virtual]

de-constructor

Definition at line 14 of file CARPHONE_WAREHOUSE.cpp.

5.6.3 Member Function Documentation

5.6.3.1 void CARPHONE_WAREHOUSE::copy (std::shared_ptr< OSTM > *to,* std::shared_ptr< OSTM > *from*) [virtual]

copy function, make deep copy of the object/pointer

Parameters

objTO	is a BANK∗ type object casted back from std::shared_ptr<OSTM>
objFROM	is a BANK∗ type object casted back from std::shared_ptr<OSTM>

Reimplemented from **OSTM** (p. 30).

Definition at line 34 of file CARPHONE_WAREHOUSE.cpp.

5.6.3.2 std::shared_ptr< OSTM > CARPHONE_WAREHOUSE::getBaseCopy (std::shared_ptr< OSTM > *object*) `[virtual]`

getBaseCopy function, make deep copy of the object/pointer and Return a new BANK∗ type object

Parameters

objTO	is a **BANK** (p. 12) type pointer for casting
obj	is a BANK∗ return type

Reimplemented from **OSTM** (p. 30).

Definition at line 21 of file CARPHONE_WAREHOUSE.cpp.

5.6.3.3 CARPHONE_WAREHOUSE CARPHONE_WAREHOUSE::operator= (const CARPHONE_WAREHOUSE & *orig*) `[inline]`

Operator

Definition at line 75 of file CARPHONE_WAREHOUSE.h.

5.6.3.4 void CARPHONE_WAREHOUSE::toString ( ) `[virtual]`

_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type

toString function, displays the object values in formatted way

Reimplemented from **OSTM** (p. 30).

Definition at line 60 of file CARPHONE_WAREHOUSE.cpp.

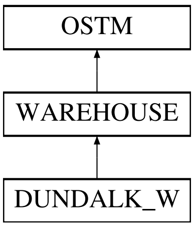
The documentation for this class was generated from the following files:

- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/↩ Test01/CARPHONE_WAREHOUSE.h
- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/↩ Test01/CARPHONE_WAREHOUSE.cpp

5.7 DUNDALK_W Class Reference

`#include <DUNDALK_W.h>`

Inheritance diagram for DUNDALK_W:



Public Member Functions

- **DUNDALK_W** ()
- **DUNDALK_W** (std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)
- **DUNDALK_W** (std::shared_ptr< **WAREHOUSE** > obj, int _version, int _unique_id)
- **DUNDALK_W** (const **DUNDALK_W** &orig)
- **DUNDALK_W operator=** (const **DUNDALK_W** &orig)
- virtual ∼**DUNDALK_W** ()
- virtual void **copy** (std::shared_ptr< **OSTM** > to, std::shared_ptr< **OSTM** > from)

  copy function, make deep copy of the object/pointer
- virtual std::shared_ptr< **OSTM** > **getBaseCopy** (std::shared_ptr< **OSTM** > object)

  getBaseCopy function, make deep copy of the object/pointer and Return a new BANK∗ type object
- virtual void **toString** ()

  _cast, is use to cast bak the std::shared_ptr<OSTM> to the required type
- virtual void **SetNumber_of_alcatel** (int _number_of_alcatel)
- virtual int **GetNumber_of_alcatel** ()
- virtual void **SetNumber_of_nokia** (int _number_of_nokia)
- virtual int **GetNumber_of_nokia** ()
- virtual void **SetNumber_of_huawei** (int _number_of_huawei)
- virtual int **GetNumber_of_huawei** ()
- virtual void **SetNumber_of_sony** (int _number_of_sony)
- virtual int **GetNumber_of_sony** ()
- virtual void **SetNumber_of_samsung** (int _number_of_samsung)
- virtual int **GetNumber_of_samsung** ()
- virtual void **SetNumber_of_iphones** (int _number_of_iphones)
- virtual int **GetNumber_of_iphones** ()
- virtual void **SetShop_name** (std::string _shop_name)
- virtual std::string **GetShop_name** ()
- virtual void **SetShop_address** (std::string _shop_address)
- virtual std::string **GetShop_address** ()

5.7.1 Detailed Description

Inherit from **WAREHOUSE** (p. 47)

Definition at line 19 of file DUNDALK_W.h.

5.7.2 Constructor & Destructor Documentation

5.7.2.1 DUNDALK_W::DUNDALK_W ( ) [inline]

Constructor

Definition at line 24 of file DUNDALK_W.h.

5.7.2.2 DUNDALK_W::DUNDALK_W (std::string *address,* std::string *shop_name,* int *iphone,* int *samsung,* int *sony,* int *huawei,* int *nokia,* int *alcatel*) [inline]

Custom constructor

Definition at line 38 of file DUNDALK_W.h.

5.7.2.3 DUNDALK_W::DUNDALK_W (std::shared_ptr< **WAREHOUSE** > *obj,* int *_version,* int *_unique_id*) [inline]

Custom constructor, used by the library for deep copying

Definition at line 55 of file DUNDALK_W.h.

5.7.2.4 DUNDALK_W::DUNDALK_W (const **DUNDALK_W** & *orig*)

Copy constructor

Definition at line 15 of file DUNDALK_W.cpp.

5.7.2.5 DUNDALK_W::∼DUNDALK_W ( ) [virtual]

de-constructor

Definition at line 12 of file DUNDALK_W.cpp.

5.7.3 Member Function Documentation

5.7.3.1 void DUNDALK_W::copy (std::shared_ptr< **OSTM** > *to,* std::shared_ptr< **OSTM** > *from*) [virtual]

copy function, make deep copy of the object/pointer

Parameters

objTO	is a BANK∗ type object casted back from std::shared_ptr<OSTM>
objFROM	is a BANK∗ type object casted back from std::shared_ptr<OSTM>

Reimplemented from **OSTM** (p. 30).

Definition at line 35 of file DUNDALK_W.cpp.

5.7.3.2 std::shared_ptr< OSTM > DUNDALK_W::getBaseCopy (std::shared_ptr< **OSTM** > *object*) [virtual]

getBaseCopy function, make deep copy of the object/pointer and Return a new BANK∗ type object

Parameters

objTO	is a **BANK** (p. 12) type pointer for casting
obj	is a BANK∗ return type

Reimplemented from **OSTM** (p. 30).

Definition at line 22 of file DUNDALK_W.cpp.

5.7.3.3 DUNDALK_W DUNDALK_W::operator= (const DUNDALK_W & *orig*) [inline]

Operator

Definition at line 75 of file DUNDALK_W.h.

5.7.3.4 void DUNDALK_W::toString ( ) [virtual]

_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type

toString function, displays the object values in formatted way

Reimplemented from **OSTM** (p. 30).

Definition at line 62 of file DUNDALK_W.cpp.

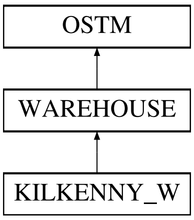
The documentation for this class was generated from the following files:

- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/↩ Test01/DUNDALK_W.h
- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/↩ Test01/DUNDALK_W.cpp

5.8 KILKENNY_W Class Reference

```
#include <KILKENNY_W.h>
```

Inheritance diagram for KILKENNY_W:



Public Member Functions

- **KILKENNY_W** ()
- **KILKENNY_W** (std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)
- **KILKENNY_W** (std::shared_ptr< **WAREHOUSE** > obj, int _version, int _unique_id)
- **KILKENNY_W** (const **KILKENNY_W** &orig)
- **KILKENNY_W operator=** (const **KILKENNY_W** &orig)
- virtual ∼**KILKENNY_W** ()
- virtual void **copy** (std::shared_ptr< **OSTM** > to, std::shared_ptr< **OSTM** > from)

  copy function, make deep copy of the object/pointer
- virtual std::shared_ptr< **OSTM** > **getBaseCopy** (std::shared_ptr< **OSTM** > object)

getBaseCopy function, make deep copy of the object/pointer and Return a new BANK∗ type object

- virtual void **toString** ()

  _cast, is use to cast bak the std::shared_ptr<OSTM> to the required type

- virtual void **SetNumber_of_alcatel** (int _number_of_alcatel)
- virtual int **GetNumber_of_alcatel** ()
- virtual void **SetNumber_of_nokia** (int _number_of_nokia)
- virtual int **GetNumber_of_nokia** ()
- virtual void **SetNumber_of_huawei** (int _number_of_huawei)
- virtual int **GetNumber_of_huawei** ()
- virtual void **SetNumber_of_sony** (int _number_of_sony)
- virtual int **GetNumber_of_sony** ()
- virtual void **SetNumber_of_samsung** (int _number_of_samsung)
- virtual int **GetNumber_of_samsung** ()
- virtual void **SetNumber_of_iphones** (int _number_of_iphones)
- virtual int **GetNumber_of_iphones** ()
- virtual void **SetShop_name** (std::string _shop_name)
- virtual std::string **GetShop_name** ()
- virtual void **SetShop_address** (std::string _shop_address)
- virtual std::string **GetShop_address** ()

5.8.1 Detailed Description

Inherit from **WAREHOUSE** (p. 47)

Definition at line 19 of file KILKENNY_W.h.

5.8.2 Constructor & Destructor Documentation

5.8.2.1 KILKENNY_W::KILKENNY_W ( ) [inline]

Constructor

Definition at line 24 of file KILKENNY_W.h.

5.8.2.2 KILKENNY_W::KILKENNY_W (std::string *address,* std::string *shop_name,* int *iphone,* int *samsung,* int *sony,* int *huawei,* int *nokia,* int *alcatel*) [inline]

Custom constructor

Definition at line 38 of file KILKENNY_W.h.

5.8.2.3 KILKENNY_W::KILKENNY_W (std::shared_ptr< WAREHOUSE > *obj,* int *_version,* int *_unique_id*) [inline]

Custom constructor, used by the library for deep copying

Definition at line 55 of file KILKENNY_W.h.

5.8.2.4 KILKENNY_W::KILKENNY_W (const KILKENNY_W & *orig*)

Copy constructor

Definition at line 15 of file KILKENNY_W.cpp.

5.8.2.5 KILKENNY_W::∼KILKENNY_W ( ) [virtual]

de-constructor

Definition at line 12 of file KILKENNY_W.cpp.

5.8.3 Member Function Documentation

5.8.3.1 void KILKENNY_W::copy (std::shared_ptr< OSTM > *to*, std::shared_ptr< OSTM > *from*) [virtual]

copy function, make deep copy of the object/pointer

Parameters

objTO	is a BANK∗ type object casted back from std::shared_ptr<OSTM>
objFROM	is a BANK∗ type object casted back from std::shared_ptr<OSTM>

Reimplemented from **OSTM** (p. 30).

Definition at line 35 of file KILKENNY_W.cpp.

5.8.3.2 std::shared_ptr< OSTM > KILKENNY_W::getBaseCopy (std::shared_ptr< OSTM > *object*) [virtual]

getBaseCopy function, make deep copy of the object/pointer and Return a new BANK∗ type object

Parameters

objTO	is a **BANK** (p. 12) type pointer for casting
obj	is a BANK∗ return type

Reimplemented from **OSTM** (p. 30).

Definition at line 22 of file KILKENNY_W.cpp.

5.8.3.3 KILKENNY_W KILKENNY_W::operator= (const KILKENNY_W & *orig*) [inline]

Operator

Definition at line 75 of file KILKENNY_W.h.

5.8.3.4 void KILKENNY_W::toString ( ) [virtual]

_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type

toString function, displays the object values in formatted way

Reimplemented from **OSTM** (p. 30).

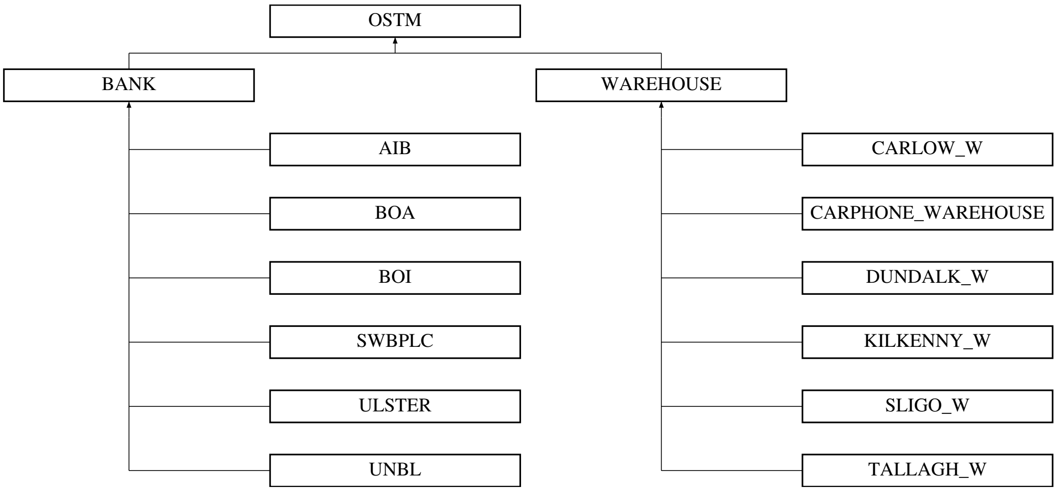
Definition at line 62 of file KILKENNY_W.cpp.

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

- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/↩ Test01/KILKENNY_W.h
- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/↩ Test01/KILKENNY_W.cpp

5.9 OSTM Class Reference

Inheritance diagram for OSTM:



Public Member Functions

- **OSTM** ()

  OSTM *(p. 30) Constructor.*
- **OSTM** (int _version_number_, int _unique_id_)

  OSTM *(p. 30) Custom Constructor.*
- virtual ∼**OSTM** ()

  De-constructor.
- virtual void **copy** (std::shared_ptr< **OSTM** > from, std::shared_ptr< **OSTM** > to)

  OSTM *(p. 30) required virtual method for deep copy.*
- virtual std::shared_ptr< **OSTM** > **getBaseCopy** (std::shared_ptr< **OSTM** > object)

  OSTM *(p. 30) required virtual method for returning a pointer that is copy of the original pointer.*
- virtual void **toString** ()

OSTM (p. 30) required virtual method for display object.

- void **Set_Unique_ID** (int uniqueID)

  setter for unique id
- int **Get_Unique_ID** () const

  getter for unique id
- void **Set_Version** (int version)

  setter for version number
- int **Get_Version** () const

  getter for version number
- void **increase_VersionNumber** ()

  commit time increase version number to child object
- bool **Is_Can_Commit** () const

  NOT USED YET.
- void **Set_Can_Commit** (bool canCommit)

  NOT USED YET.
- void **Set_Abort_Transaction** (bool abortTransaction)

  NOT USED YET.
- bool **Is_Abort_Transaction** () const

  NOT USED YET.
- void **lock_Mutex** ()

  object unique lock, locks mutex
- void **unlock_Mutex** ()

  object unique lock, unlocks mutex
- bool **is_Locked** ()

  object unique lock, try locks mutex return boolean value depends on the lock state

5.9.1 Detailed Description

Definition at line 17 of file OSTM.h.

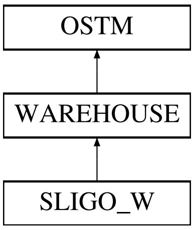
The documentation for this class was generated from the following file:

- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/↩ Test01/OSTM.h

5.10 SLIGO_W Class Reference

```
#include <SLIGO_W.h>
```

Inheritance diagram for SLIGO_W:



Public Member Functions

- **SLIGO_W** ()
- **SLIGO_W** (std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)
- **SLIGO_W** (std::shared_ptr< **WAREHOUSE** > obj, int _version, int _unique_id)
- **SLIGO_W** (const **SLIGO_W** &orig)
- **SLIGO_W operator=** (const **SLIGO_W** &orig)
- virtual ∼**SLIGO_W** ()
- virtual void **copy** (std::shared_ptr< **OSTM** > to, std::shared_ptr< **OSTM** > from)

  copy function, make deep copy of the object/pointer
- virtual std::shared_ptr< **OSTM** > **getBaseCopy** (std::shared_ptr< **OSTM** > object)

  getBaseCopy function, make deep copy of the object/pointer and Return a new BANK∗ type object
- virtual void **toString** ()

  _cast, is use to cast bak the std::shared_ptr<OSTM> to the required type
- virtual void **SetNumber_of_alcatel** (int _number_of_alcatel)
- virtual int **GetNumber_of_alcatel** ()
- virtual void **SetNumber_of_nokia** (int _number_of_nokia)
- virtual int **GetNumber_of_nokia** ()
- virtual void **SetNumber_of_huawei** (int _number_of_huawei)
- virtual int **GetNumber_of_huawei** ()
- virtual void **SetNumber_of_sony** (int _number_of_sony)
- virtual int **GetNumber_of_sony** ()
- virtual void **SetNumber_of_samsung** (int _number_of_samsung)
- virtual int **GetNumber_of_samsung** ()
- virtual void **SetNumber_of_iphones** (int _number_of_iphones)
- virtual int **GetNumber_of_iphones** ()
- virtual void **SetShop_name** (std::string _shop_name)
- virtual std::string **GetShop_name** ()
- virtual void **SetShop_address** (std::string _shop_address)
- virtual std::string **GetShop_address** ()

5.10.1 Detailed Description

Inherit from **WAREHOUSE** (p. 47)

Definition at line 19 of file SLIGO_W.h.

5.10.2 Constructor & Destructor Documentation

5.10.2.1 SLIGO_W::SLIGO_W ( ) [inline]

Constructor

Definition at line 24 of file SLIGO_W.h.

5.10.2.2 SLIGO_W::SLIGO_W (std::string *address,* std::string *shop_name,* int *iphone,* int *samsung,* int *sony,* int *huawei,* int *nokia,* int *alcatel*) [inline]

Custom constructor

Definition at line 38 of file SLIGO_W.h.

5.10.2.3 SLIGO_W::SLIGO_W (std::shared_ptr< **WAREHOUSE** > *obj,* int *_version,* int *_unique_id*) [inline]

Custom constructor, used by the library for deep copying

Definition at line 55 of file SLIGO_W.h.

5.10.2.4 SLIGO_W::SLIGO_W (const **SLIGO_W** & *orig*)

Copy constructor

Definition at line 15 of file SLIGO_W.cpp.

5.10.2.5 SLIGO_W::∼SLIGO_W ( ) [virtual]

de-constructor

Definition at line 12 of file SLIGO_W.cpp.

5.10.3 Member Function Documentation

5.10.3.1 void SLIGO_W::copy (std::shared_ptr< **OSTM** > *to,* std::shared_ptr< **OSTM** > *from*) [virtual]

copy function, make deep copy of the object/pointer

Parameters

objTO	is a BANK∗ type object casted back from std::shared_ptr<OSTM>
objFROM	is a BANK∗ type object casted back from std::shared_ptr<OSTM>

Reimplemented from **OSTM** (p. 30).

Definition at line 35 of file SLIGO_W.cpp.

5.10.3.2 std::shared_ptr< **OSTM** > SLIGO_W::getBaseCopy (std::shared_ptr< **OSTM** > *object*) [virtual]

getBaseCopy function, make deep copy of the object/pointer and Return a new BANK∗ type object

Parameters

objTO	is a **BANK** (p. 12) type pointer for casting
obj	is a BANK∗ return type

Reimplemented from **OSTM** (p. 30).

Definition at line 22 of file SLIGO_W.cpp.

5.10.3.3 SLIGO_W SLIGO_W::operator= (const SLIGO_W & *orig*) `[inline]`

Operator

Definition at line 75 of file SLIGO_W.h.

5.10.3.4 void SLIGO_W::toString ( ) `[virtual]`

_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type

toString function, displays the object values in formatted way

Reimplemented from **OSTM** (p. 30).

Definition at line 62 of file SLIGO_W.cpp.

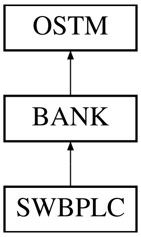
The documentation for this class was generated from the following files:

- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/←↩ Test01/SLIGO_W.h
- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/←↩ Test01/SLIGO_W.cpp

5.11 SWBPLC Class Reference

`#include <SWBPLC.h>`

Inheritance diagram for SWBPLC:



Public Member Functions

- **SWBPLC** ()
- **SWBPLC** (int accountNumber, double balance, std::string firstName, std::string lastName, std::string address)
- **SWBPLC** (std::shared_ptr< **BANK** > obj, int _version, int _unique_id)
- **SWBPLC** (const **SWBPLC** &orig)
- **SWBPLC operator=** (const **SWBPLC** &orig)
- virtual ∼**SWBPLC** ()
- virtual void **copy** (std::shared_ptr< **OSTM** > to, std::shared_ptr< **OSTM** > from)

    copy function, make deep copy of the object/pointer
- virtual std::shared_ptr< **OSTM** > **getBaseCopy** (std::shared_ptr< **OSTM** > object)

getBaseCopy function, make deep copy of the object/pointer and Return a new std::shared_ptr<BANK> type object
- virtual void **toString** ()

  _cast, is use to cast bak the std::shared_ptr<OSTM> to the required type
- virtual void **SetAddress** (std::string address)
- virtual std::string **GetAddress** () const
- virtual void **SetBalance** (double balance)
- virtual double **GetBalance** () const
- virtual void **SetAccountNumber** (int accountNumber)
- virtual int **GetAccountNumber** () const
- virtual void **SetLastName** (std::string lastName)
- virtual std::string **GetLastName** () const
- virtual void **SetFirstName** (std::string firstName)
- virtual std::string **GetFirstName** () const
- virtual void **SetFullname** (std::string fullname)
- virtual std::string **GetFullname** () const

5.11.1 Detailed Description

Inherit from **BANK** (p. 12)

Definition at line 19 of file SWBPLC.h.

5.11.2 Constructor & Destructor Documentation

5.11.2.1 SWBPLC::SWBPLC ( ) [inline]

Constructor

Definition at line 24 of file SWBPLC.h.

5.11.2.2 SWBPLC::SWBPLC (int *accountNumber,* double *balance,* std::string *firstName,* std::string *lastName,* std::string *address*) [inline]

Custom constructor

Definition at line 35 of file SWBPLC.h.

5.11.2.3 SWBPLC::SWBPLC (std::shared_ptr< BANK > *obj,* int *_version,* int *_unique_id*) [inline]

Custom constructor, used by the library for deep copying

Definition at line 46 of file SWBPLC.h.

5.11.2.4 SWBPLC::SWBPLC (const SWBPLC & *orig*)

Copy constructor

Definition at line 12 of file SWBPLC.cpp.

5.11.2.5 SWBPLC::∼SWBPLC ( ) [virtual]

de-constructor

Definition at line 15 of file SWBPLC.cpp.

5.11.3 Member Function Documentation

5.11.3.1 void SWBPLC::copy (std::shared_ptr< OSTM > *to,* std::shared_ptr< OSTM > *from*) [virtual]

copy function, make deep copy of the object/pointer

Parameters

objTO	is a std::shared_ptr<BANK> type object casted back from std::shared_ptr<OSTM>
objFROM	is a std::shared_ptr<BANK> type object casted back from std::shared_ptr<OSTM>

Reimplemented from **OSTM** (p. 30).

Definition at line 34 of file SWBPLC.cpp.

5.11.3.2 std::shared_ptr< OSTM > SWBPLC::getBaseCopy (std::shared_ptr< OSTM > *object*) [virtual]

getBaseCopy function, make deep copy of the object/pointer and Return a new std::shared_ptr<BANK> type object

Parameters

objTO	is a **BANK** (p. 12) type pointer for casting
obj	is a std::shared_ptr<BANK> return type

Reimplemented from **OSTM** (p. 30).

Definition at line 22 of file SWBPLC.cpp.

5.11.3.3 SWBPLC SWBPLC::operator= (const SWBPLC & *orig*) [inline]

Operator

Definition at line 63 of file SWBPLC.h.

5.11.3.4 void SWBPLC::toString ( ) [virtual]

_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type

toString function, displays the object values in formatted way

Reimplemented from **OSTM** (p. 30).

Definition at line 55 of file SWBPLC.cpp.

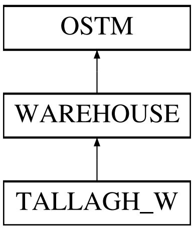
The documentation for this class was generated from the following files:

- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/↩ Test01/SWBPLC.h
- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/↩ Test01/SWBPLC.cpp

5.12 TALLAGH_W Class Reference

```
#include <TALLAGH_W.h>
```

Inheritance diagram for TALLAGH_W:



Public Member Functions

- **TALLAGH_W** ()
- **TALLAGH_W** (std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)
- **TALLAGH_W** (std::shared_ptr< **WAREHOUSE** > obj, int _version, int _unique_id)
- **TALLAGH_W** (const **TALLAGH_W** &orig)
- **TALLAGH_W operator=** (const **TALLAGH_W** &orig)
- virtual ∼**TALLAGH_W** ()
- virtual void **copy** (std::shared_ptr< **OSTM** > to, std::shared_ptr< **OSTM** > from)

  copy function, make deep copy of the object/pointer
- virtual std::shared_ptr< **OSTM** > **getBaseCopy** (std::shared_ptr< **OSTM** > object)

  getBaseCopy function, make deep copy of the object/pointer and Return a new BANK∗ type object
- virtual void **toString** ()

  _cast, is use to cast bak the std::shared_ptr<OSTM> to the required type
- virtual void **SetNumber_of_alcatel** (int _number_of_alcatel)
- virtual int **GetNumber_of_alcatel** ()
- virtual void **SetNumber_of_nokia** (int _number_of_nokia)
- virtual int **GetNumber_of_nokia** ()
- virtual void **SetNumber_of_huawei** (int _number_of_huawei)
- virtual int **GetNumber_of_huawei** ()
- virtual void **SetNumber_of_sony** (int _number_of_sony)
- virtual int **GetNumber_of_sony** ()
- virtual void **SetNumber_of_samsung** (int _number_of_samsung)
- virtual int **GetNumber_of_samsung** ()
- virtual void **SetNumber_of_iphones** (int _number_of_iphones)
- virtual int **GetNumber_of_iphones** ()
- virtual void **SetShop_name** (std::string _shop_name)
- virtual std::string **GetShop_name** ()
- virtual void **SetShop_address** (std::string _shop_address)
- virtual std::string **GetShop_address** ()

5.12.1 Detailed Description

Inherit from **WAREHOUSE** (p. 47)

Definition at line 19 of file TALLAGH_W.h.

5.12.2 Constructor & Destructor Documentation

5.12.2.1 TALLAGH_W::TALLAGH_W ( ) [inline]

Constructor

Definition at line 24 of file TALLAGH_W.h.

5.12.2.2 TALLAGH_W::TALLAGH_W (std::string *address,* std::string *shop_name,* int *iphone,* int *samsung,* int *sony,* int *huawei,* int *nokia,* int *alcatel*) [inline]

Custom constructor

Definition at line 38 of file TALLAGH_W.h.

5.12.2.3 TALLAGH_W::TALLAGH_W (std::shared_ptr< WAREHOUSE > *obj,* int *_version,* int *_unique_id*) [inline]

Custom constructor, used by the library for deep copying

Definition at line 55 of file TALLAGH_W.h.

5.12.2.4 TALLAGH_W::TALLAGH_W (const TALLAGH_W & *orig*)

Copy constructor

Definition at line 15 of file TALLAGH_W.cpp.

5.12.2.5 TALLAGH_W::∼TALLAGH_W ( ) [virtual]

de-constructor

Definition at line 12 of file TALLAGH_W.cpp.

5.12.3 Member Function Documentation

5.12.3.1 void TALLAGH_W::copy (std::shared_ptr< OSTM > *to,* std::shared_ptr< OSTM > *from*) [virtual]

copy function, make deep copy of the object/pointer

Parameters

objTO	is a BANK∗ type object casted back from std::shared_ptr<OSTM>
objFROM	is a BANK∗ type object casted back from std::shared_ptr<OSTM>

Reimplemented from **OSTM** (p. 30).

Definition at line 35 of file TALLAGH_W.cpp.

5.12.3.2 std::shared_ptr< OSTM > TALLAGH_W::getBaseCopy (std::shared_ptr< OSTM > *object*) [virtual]

getBaseCopy function, make deep copy of the object/pointer and Return a new BANK∗ type object

Parameters

objTO	is a **BANK** (p. 12) type pointer for casting
obj	is a BANK∗ return type

Reimplemented from **OSTM** (p. 30).

Definition at line 22 of file TALLAGH_W.cpp.

5.12.3.3 TALLAGH_W TALLAGH_W::operator= (const TALLAGH_W & *orig*) [inline]

Operator

Definition at line 75 of file TALLAGH_W.h.

5.12.3.4 void TALLAGH_W::toString ( ) [virtual]

_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type

toString function, displays the object values in formatted way

Reimplemented from **OSTM** (p. 30).

Definition at line 62 of file TALLAGH_W.cpp.

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

- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/Test01/TALLAGH_W.h
- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/Test01/TALLAGH_W.cpp

5.13 TM Class Reference

Public Member Functions

- std::shared_ptr< **TX** > const **_get_tx** ()

  _get_tx std::shared_ptr<TX>, returning a shared pointer with the transaction
- void **_TX_EXIT** ()

  _TX_EXIT void, the thread calls the ostm_exit function in the transaction, and clear all elements from the shared global collection associated with the main process
- void **print_all** ()

Static Public Member Functions

- static **TM** & **Instance** ()

  Scott Meyer's Singleton creation, what is thread safe.

5.13.1 Detailed Description

Definition at line 56 of file TM.h.

5.13.2 Member Function Documentation

5.13.2.1 void TM::print_all ( )

ONLY FOR TESTING print_all void, prints all object in the txMap

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

- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/↩ Test01/TM.h

5.14 TX Class Reference

Public Member Functions

- **TX** (std::thread::id id)

  Constructor.
- ∼**TX** ()

  De-constructor.
- **TX** (const **TX** &orig)

  Default copy constructor.
- void **ostm_exit** ()

  Delete all map entries associated with the main process.
- void **_register** (std::shared_ptr< **OSTM** > object)

  *Register **OSTM** (p. 30) pointer into STM library.*
- std::shared_ptr< **OSTM** > **load** (std::shared_ptr< **OSTM** > object)

- void **store** (std::shared_ptr< **OSTM** > object)

  Store transactional changes.
- bool **commit** ()

  Commit transactional changes.
- void **_increase_tx_nesting** ()

  *Add **TX** (*p. *40) nesting level by one.*
- void **_decrease_tx_nesting** ()

  *Remove **TX** (*p. *40) nesting level by one.*
- int **getTest_counter** ()
- void **_print_all_tx** ()

Static Public Attributes

- static int **test_counter**

Friends

- class **TM**

5.14.1 Detailed Description

Definition at line 24 of file TX.h.

5.14.2 Member Function Documentation

5.14.2.1 std::shared_ptr<OSTM> TX::load (std::shared_ptr< OSTM > *object*)

Register **OSTM** (p. 30) pointer into STM library

5.14.3 Friends And Related Function Documentation

5.14.3.1 friend class TM [friend]

Only **TM** (p. 40) Transaction Manager can create instance of **TX** (p. 40) Transaction

Definition at line 70 of file TX.h.

5.14.4 Member Data Documentation

5.14.4.1 int TX::test_counter [static]

Parameters

test_counter	int ONLY FOR TESTING!!!

Definition at line 78 of file TX.h.

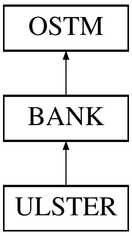
The documentation for this class was generated from the following file:

- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/↩ Test01/TX.h

5.15 ULSTER Class Reference

```
#include <ULSTER.h>
```

Inheritance diagram for ULSTER:



Public Member Functions

- **ULSTER** ()
- **ULSTER** (int accountNumber, double balance, std::string firstName, std::string lastName, std::string address)
- **ULSTER** (std::shared_ptr< **BANK** > obj, int _version, int _unique_id)
- **ULSTER** (const **ULSTER** &orig)
- **ULSTER operator=** (const **ULSTER** &orig)
- virtual ∼**ULSTER** ()
- virtual void **copy** (std::shared_ptr< **OSTM** > to, std::shared_ptr< **OSTM** > from)

    copy function, make deep copy of the object/pointer
- virtual std::shared_ptr< **OSTM** > **getBaseCopy** (std::shared_ptr< **OSTM** > object)

    getBaseCopy function, make deep copy of the object/pointer and Return a new std::shared_ptr<BANK> type object
- virtual void **toString** ()

    _cast, is use to cast bak the std::shared_ptr<OSTM> to the required type
- virtual void **SetAddress** (std::string address)
- virtual std::string **GetAddress** () const
- virtual void **SetBalance** (double balance)
- virtual double **GetBalance** () const
- virtual void **SetAccountNumber** (int accountNumber)
- virtual int **GetAccountNumber** () const
- virtual void **SetLastName** (std::string lastName)
- virtual std::string **GetLastName** () const
- virtual void **SetFirstName** (std::string firstName)
- virtual std::string **GetFirstName** () const
- virtual void **SetFullname** (std::string fullname)
- virtual std::string **GetFullname** () const

5.15.1 Detailed Description

Inherit from **BANK** (p. 12)

Definition at line 19 of file ULSTER.h.

5.15.2 Constructor & Destructor Documentation

5.15.2.1 ULSTER::ULSTER ( ) [inline]

Constructor

Definition at line 24 of file ULSTER.h.

5.15.2.2 ULSTER::ULSTER (int *accountNumber,* double *balance,* std::string *firstName,* std::string *lastName,* std::string *address*) [inline]

Custom constructor

Definition at line 35 of file ULSTER.h.

5.15.2.3 ULSTER::ULSTER (std::shared_ptr< BANK > *obj,* int *_version,* int *_unique_id*) [inline]

Custom constructor, used by the library for deep copying

Definition at line 46 of file ULSTER.h.

5.15.2.4 ULSTER::ULSTER (const ULSTER & *orig*)

Copy constructor

Definition at line 15 of file ULSTER.cpp.

5.15.2.5 ULSTER::∼ULSTER ( ) [virtual]

de-constructor

Definition at line 18 of file ULSTER.cpp.

5.15.3 Member Function Documentation

5.15.3.1 void ULSTER::copy (std::shared_ptr< OSTM > *to,* std::shared_ptr< OSTM > *from*) [virtual]

copy function, make deep copy of the object/pointer

Parameters

objTO	is a std::shared_ptr<BANK> type object casted back from std::shared_ptr<OSTM>
objFROM	is a std::shared_ptr<BANK> type object casted back from std::shared_ptr<OSTM>

Reimplemented from **OSTM** (p. 30).

Definition at line 37 of file ULSTER.cpp.

5.15.3.2 std::shared_ptr< **OSTM** > ULSTER::getBaseCopy (std::shared_ptr< **OSTM** > *object*) [virtual]

getBaseCopy function, make deep copy of the object/pointer and Return a new std::shared_ptr<BANK> type object

Parameters

objTO	is a **BANK** (p. 12) type pointer for casting
obj	is a std::shared_ptr<BANK> return type

Reimplemented from **OSTM** (p. 30).

Definition at line 25 of file ULSTER.cpp.

5.15.3.3 **ULSTER** ULSTER::operator= (const **ULSTER** & *orig*) [inline]

Operator

Definition at line 62 of file ULSTER.h.

5.15.3.4 void ULSTER::toString ( ) [virtual]

_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type

toString function, displays the object values in formatted way

Reimplemented from **OSTM** (p. 30).

Definition at line 58 of file ULSTER.cpp.

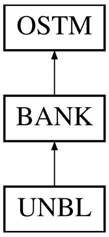
The documentation for this class was generated from the following files:

- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/Test01/ULSTER.h
- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/Test01/ULSTER.cpp

5.16 UNBL Class Reference

```
#include <UNBL.h>
```

Inheritance diagram for UNBL:



Public Member Functions

- **UNBL** ()
- **UNBL** (int accountNumber, double balance, std::string firstName, std::string lastName, std::string address)
- **UNBL** (std::shared_ptr< **BANK** > obj, int _version, int _unique_id)
- **UNBL** (const **UNBL** &orig)
- **UNBL operator=** (const **UNBL** &orig)
- virtual ∼**UNBL** ()
- virtual void **copy** (std::shared_ptr< **OSTM** > to, std::shared_ptr< **OSTM** > from)

  copy function, make deep copy of the object/pointer
- virtual std::shared_ptr< **OSTM** > **getBaseCopy** (std::shared_ptr< **OSTM** > object)

  getBaseCopy function, make deep copy of the object/pointer and Return a new std::shared_ptr<BANK> type object
- virtual void **toString** ()

  _cast, is use to cast bak the std::shared_ptr<OSTM> to the required type
- virtual void **SetAddress** (std::string address)
- virtual std::string **GetAddress** () const
- virtual void **SetBalance** (double balance)
- virtual double **GetBalance** () const
- virtual void **SetAccountNumber** (int accountNumber)
- virtual int **GetAccountNumber** () const
- virtual void **SetLastName** (std::string lastName)
- virtual std::string **GetLastName** () const
- virtual void **SetFirstName** (std::string firstName)
- virtual std::string **GetFirstName** () const
- virtual void **SetFullname** (std::string fullname)
- virtual std::string **GetFullname** () const

5.16.1 Detailed Description

Inherit from **BANK** (p. 12)

Definition at line 19 of file UNBL.h.

5.16.2 Constructor & Destructor Documentation

5.16.2.1 UNBL::UNBL ( ) [inline]

Constructor

Definition at line 24 of file UNBL.h.

5.16.2.2 UNBL::UNBL (int *accountNumber,* double *balance,* std::string *firstName,* std::string *lastName,* std::string *address*) [inline]

Custom constructor

Definition at line 35 of file UNBL.h.

5.16.2.3 UNBL::UNBL (std::shared_ptr< **BANK** > *obj,* int *_version,* int *_unique_id*) [inline]

Custom constructor, used by the library for deep copying

Definition at line 46 of file UNBL.h.

5.16.2.4 UNBL::UNBL (const **UNBL** & *orig*)

Copy constructor

Definition at line 11 of file UNBL.cpp.

5.16.2.5 UNBL::∼UNBL ( ) [virtual]

de-constructor

Definition at line 14 of file UNBL.cpp.

5.16.3 Member Function Documentation

5.16.3.1 void UNBL::copy (std::shared_ptr< **OSTM** > *to,* std::shared_ptr< **OSTM** > *from*) [virtual]

copy function, make deep copy of the object/pointer

Parameters

objTO	is a std::shared_ptr<BANK> type object casted back from std::shared_ptr<OSTM>
objFROM	is a std::shared_ptr<BANK> type object casted back from std::shared_ptr<OSTM>

Reimplemented from **OSTM** (p. 30).

Definition at line 33 of file UNBL.cpp.

5.16.3.2 std::shared_ptr< OSTM > UNBL::getBaseCopy (std::shared_ptr< OSTM > *object*) [virtual]

getBaseCopy function, make deep copy of the object/pointer and Return a new std::shared_ptr<BANK> type object

Parameters

objTO	is a **BANK** (p. 12) type pointer for casting
obj	is a std::shared_ptr<BANK> return type

Reimplemented from **OSTM** (p. 30).

Definition at line 21 of file UNBL.cpp.

5.16.3.3 UNBL UNBL::operator= (const UNBL & *orig*) [inline]

Operator

Definition at line 62 of file UNBL.h.

5.16.3.4 void UNBL::toString ( ) [virtual]

_cast, is use to cast bak the std::shared_ptr<OSTM> to the required type

toString function, displays the object values in formatted way

Reimplemented from **OSTM** (p. 30).

Definition at line 53 of file UNBL.cpp.

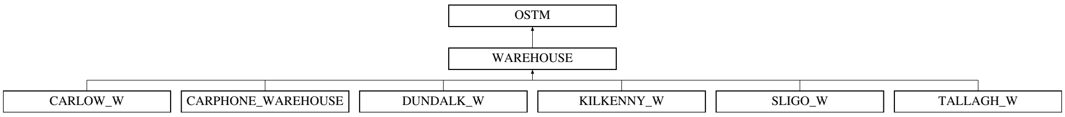
The documentation for this class was generated from the following files:

- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/Test01/UNBL.h
- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/Test01/UNBL.cpp

5.17 WAREHOUSE Class Reference

`#include <WAREHOUSE.h>`

Inheritance diagram for WAREHOUSE:



Public Member Functions

- **WAREHOUSE** ()
- **WAREHOUSE** (int _version, int _unique_id)
- **WAREHOUSE** (const **WAREHOUSE** &orig)
- virtual ∼**WAREHOUSE** ()
- virtual void **SetNumber_of_alcatel** (int _number_of_alcatel)
- virtual int **GetNumber_of_alcatel** ()
- virtual void **SetNumber_of_nokia** (int _number_of_nokia)
- virtual int **GetNumber_of_nokia** ()
- virtual void **SetNumber_of_huawei** (int _number_of_huawei)
- virtual int **GetNumber_of_huawei** ()
- virtual void **SetNumber_of_sony** (int _number_of_sony)
- virtual int **GetNumber_of_sony** ()
- virtual void **SetNumber_of_samsung** (int _number_of_samsung)
- virtual int **GetNumber_of_samsung** ()
- virtual void **SetNumber_of_iphones** (int _number_of_iphones)
- virtual int **GetNumber_of_iphones** ()
- virtual void **SetShop_name** (std::string _shop_name)
- virtual std::string **GetShop_name** ()
- virtual void **SetShop_address** (std::string _shop_address)
- virtual std::string **GetShop_address** ()

5.17.1 Detailed Description

WAREHOUSE (p. 47) inherit from **OSTM** (p. 30) library

Definition at line 16 of file WAREHOUSE.h.

5.17.2 Constructor & Destructor Documentation

5.17.2.1 WAREHOUSE::WAREHOUSE ( ) `[inline]`

Constructor

Definition at line 21 of file WAREHOUSE.h.

5.17.2.2 WAREHOUSE::WAREHOUSE (int *_version,* int *_unique_id*) `[inline]`

Custom Constructor

Definition at line 27 of file WAREHOUSE.h.

5.17.2.3 WAREHOUSE::WAREHOUSE (const WAREHOUSE & *orig*)

Copy constructor

Definition at line 12 of file WAREHOUSE.cpp.

5.17.2.4 WAREHOUSE::∼WAREHOUSE ( ) [virtual]

de-constructor

Definition at line 15 of file WAREHOUSE.cpp.

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

- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/↩ Test01/WAREHOUSE.h
- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/↩ Test01/WAREHOUSE.cpp