C++ Softwrae Transactional memory

Zoltan Fuzesi

Contents

1	OST	M C++ :	Software Transactional Memory	1
	1.1	Object	Based Software Transactional Memory	1
		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	1
		1.1.2	Step 1: Download the archive file.	1
		1.1.3	Step 2: Unzip in the target destination	1
		1.1.4	Step 3: Copy the shared library (libostm.so) to the operating system folder where the other shared library are stored	1
		1.1.5	Step 4: Achieve the required class hierarchy between the OSTM library and your own class structure	1
		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	2
		1.1.7	Step 6: Now your application use transactional environment, that guarantees the consistency between object transactions	2
		1.1.8	Step 7: Run the application.	2
2	REA	DME		3
3	Hier	archica	l Index	5
	3.1	Class	Hierarchy	5
4	Clas	s Index		7
	4.1	Class	List	7

iv CONTENTS

5	Clas	s Docu	mentation	1	9
	5.1	AIB CI	ass Refere	ence	9
		5.1.1	Detailed	Description	10
		5.1.2	Construc	ctor & Destructor Documentation	10
			5.1.2.1	AIB()	10
			5.1.2.2	AIB(int accountNumber, double balance, std::string firstName, std::string last ← Name, std::string address)	10
			5.1.2.3	AIB(std::shared_ptr< BANK > obj, int _version, int _unique_id)	10
			5.1.2.4	AIB(const AIB &orig)	10
			5.1.2.5	~AIB()	10
		5.1.3	Member	Function Documentation	10
			5.1.3.1	$\label{eq:copy} \mbox{copy(std::shared_ptr} < \mbox{OSTM} > \mbox{to, std::shared_ptr} < \mbox{OSTM} > \mbox{from)} . \ . \ . \ . \ . \ . \ . \ . \ . \ .$	10
			5.1.3.2	getBaseCopy(std::shared_ptr< OSTM > object)	11
			5.1.3.3	operator=(const AIB &orig)	11
			5.1.3.4	toString()	11
	5.2	BANK	Class Ref	erence	12
		5.2.1	Detailed	Description	12
		5.2.2	Construc	ctor & Destructor Documentation	12
			5.2.2.1	BANK()	12
			5.2.2.2	BANK(int _version, int _unique_id)	13
			5.2.2.3	BANK(const BANK &orig)	13
			5.2.2.4	~BANK()	13
	5.3	BOA C	Class Refer	rence	13
		5.3.1	Detailed	Description	14
		5.3.2	Construc	ctor & Destructor Documentation	14
			5.3.2.1	BOA()	14
			5.3.2.2	BOA(int accountNumber, double balance, std::string firstName, std::string last ← Name, std::string address)	14
			5.3.2.3	BOA(std::shared_ptr< BANK > obj, int _version, int _unique_id)	15
			5.3.2.4	BOA(const BOA &orig)	15
			5.3.2.5	~BOA()	15

CONTENTS

	5.3.3	Member	Function Documentation	15
		5.3.3.1	$\label{eq:copy} \mbox{copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)} . \ . \ . \ . \ . \ . \ . \ . \ . \ .$	15
		5.3.3.2	getBaseCopy(std::shared_ptr< OSTM > object)	15
		5.3.3.3	operator=(const BOA &orig)	16
		5.3.3.4	toString()	16
5.4	BOI CI	ass Refere	ence	16
	5.4.1	Detailed	Description	17
	5.4.2	Construc	ctor & Destructor Documentation	17
		5.4.2.1	BOI()	17
		5.4.2.2	BOI(int accountNumber, double balance, std::string firstName, std::string last← Name, std::string address)	17
		5.4.2.3	BOI(std::shared_ptr< BOI > obj, int _version, int _unique_id)	17
		5.4.2.4	BOI(const BOI &orig)	17
		5.4.2.5	~BOI()	18
	5.4.3	Member	Function Documentation	18
		5.4.3.1	$\label{eq:copy} \mbox{copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)} . \ . \ . \ . \ . \ . \ . \ . \ . \ .$	18
		5.4.3.2	getBaseCopy(std::shared_ptr< OSTM > object)	18
		5.4.3.3	operator=(const BOI &orig)	18
		5.4.3.4	toString()	18
5.5	CARLO	DW_W Cla	ass Reference	19
	5.5.1	Detailed	Description	20
	5.5.2	Construc	etor & Destructor Documentation	20
		5.5.2.1	CARLOW_W()	20
		5.5.2.2	CARLOW_W(std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)	20
		5.5.2.3	${\tt CARLOW_W(std::shared_ptr} < {\tt WAREHOUSE} > {\tt obj, int_version, int_unique_id})$	20
		5.5.2.4	CARLOW_W(const CARLOW_W &orig)	20
		5.5.2.5	~CARLOW_W()	20
	5.5.3	Member	Function Documentation	20
		5.5.3.1	$\label{eq:copy} \mbox{copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)} . \ . \ . \ . \ . \ . \ . \ . \ . \ .$	20
		5.5.3.2	getBaseCopy(std::shared_ptr< OSTM > object)	21

vi

		5.5.3.3	operator=(const CARLOW_W &orig)	21
		5.5.3.4	toString()	21
5.6	CARP	HONE_WA	AREHOUSE Class Reference	22
	5.6.1	Detailed	Description	22
	5.6.2	Construc	ctor & Destructor Documentation	23
		5.6.2.1	CARPHONE_WAREHOUSE()	23
		5.6.2.2	CARPHONE_WAREHOUSE(std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)	23
		5.6.2.3	CARPHONE_WAREHOUSE(std::shared_ptr< WAREHOUSE > obj, int _ correction, int _unique_id)	23
		5.6.2.4	CARPHONE_WAREHOUSE(const CARPHONE_WAREHOUSE & orig)	23
		5.6.2.5	~CARPHONE_WAREHOUSE()	23
	5.6.3	Member	Function Documentation	23
		5.6.3.1	$\label{eq:copy} \mbox{copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)} $	23
		5.6.3.2	getBaseCopy(std::shared_ptr< OSTM > object)	24
		5.6.3.3	operator=(const CARPHONE_WAREHOUSE &orig)	24
		5.6.3.4	toString()	24
5.7	DUND	ALK_W CI	lass Reference	24
	5.7.1	Detailed	Description	25
	5.7.2	Construc	ctor & Destructor Documentation	25
		5.7.2.1	DUNDALK_W()	25
		5.7.2.2	DUNDALK_W(std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)	25
		5.7.2.3	DUNDALK_W(std::shared_ptr< WAREHOUSE > obj, int _version, int _unique ← _id)	26
		5.7.2.4	DUNDALK_W(const DUNDALK_W &orig)	26
		5.7.2.5	~DUNDALK_W()	26
	5.7.3	Member	Function Documentation	26
		5.7.3.1	$\label{eq:copy} \mbox{copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)} $	26
		5.7.3.2	getBaseCopy(std::shared_ptr< OSTM > object)	26
		5.7.3.3	operator=(const DUNDALK_W &orig)	27
		5.7.3.4	toString()	27

CONTENTS vii

5.8	KILKEN	NNY_W CI	ass Reference	27
	5.8.1	Detailed I	Description	28
	5.8.2	Construc	tor & Destructor Documentation	28
		5.8.2.1	KILKENNY_W()	28
		5.8.2.2	KILKENNY_W(std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)	28
		5.8.2.3	KILKENNY_W(std::shared_ptr< WAREHOUSE > obj, int _version, int _unique ← _id)	28
		5.8.2.4	KILKENNY_W(const KILKENNY_W &orig)	29
		5.8.2.5	~KILKENNY_W()	29
	5.8.3	Member I	Function Documentation	29
		5.8.3.1	$\label{eq:copy} \mbox{copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)} $	29
		5.8.3.2	getBaseCopy(std::shared_ptr< OSTM > object)	29
		5.8.3.3	operator=(const KILKENNY_W &orig)	29
		5.8.3.4	toString()	30
5.9	OSTM	Class Refe	erence	30
	5.9.1	Detailed I	Description	31
5.10	SLIGO	_W Class	Reference	31
	5.10.1	Detailed I	Description	32
	5.10.2	Construc	tor & Destructor Documentation	32
		5.10.2.1	SLIGO_W()	32
		5.10.2.2	SLIGO_W(std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)	32
		5.10.2.3	${\sf SLIGO_W(std::shared_ptr} < {\sf WAREHOUSE} > {\sf obj, int_version, int_unique_id}) \ \ .$	33
		5.10.2.4	SLIGO_W(const SLIGO_W &orig)	33
		5.10.2.5	~SLIGO_W()	33
	5.10.3	Member I	Function Documentation	33
		5.10.3.1	$\label{eq:copy} \mbox{copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)} $	33
		5.10.3.2	getBaseCopy(std::shared_ptr< OSTM > object)	33
		5.10.3.3	operator=(const SLIGO_W &orig)	34
		5.10.3.4	toString()	34
5.11	SWBPI	_C Class F	Reference	34

viii CONTENTS

	5.11.1	Detailed I	Description	35
	5.11.2	Construc	tor & Destructor Documentation	35
		5.11.2.1	SWBPLC()	35
		5.11.2.2	SWBPLC(int accountNumber, double balance, std::string firstName, std::string lastName, std::string address)	35
		5.11.2.3	SWBPLC(std::shared_ptr< BANK > obj, int _version, int _unique_id)	35
		5.11.2.4	SWBPLC(const SWBPLC &orig)	35
		5.11.2.5	~SWBPLC()	36
	5.11.3	Member I	Function Documentation	36
		5.11.3.1	$\label{eq:copy} \mbox{copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)} $	36
		5.11.3.2	getBaseCopy(std::shared_ptr< OSTM > object)	36
		5.11.3.3	operator=(const SWBPLC &orig)	36
		5.11.3.4	toString()	36
5.12	TALLA	GH_W Cla	ass Reference	37
	5.12.1	Detailed I	Description	38
	5.12.2	Construc	tor & Destructor Documentation	38
		5.12.2.1	TALLAGH_W()	38
		5.12.2.2	TALLAGH_W(std::string address, std::string shop_name, int iphone, int samsung, int sony, int huawei, int nokia, int alcatel)	38
		5.12.2.3	TALLAGH_W(std::shared_ptr< WAREHOUSE > obj, int _version, int _unique ← _id)	38
		5.12.2.4	TALLAGH_W(const TALLAGH_W &orig)	38
		5.12.2.5	\sim TALLAGH_W()	38
	5.12.3	Member I	Function Documentation	38
		5.12.3.1	$\label{eq:copy} \mbox{copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)} . \ . \ . \ . \ . \ . \ . \ .$	38
		5.12.3.2	getBaseCopy(std::shared_ptr< OSTM > object)	39
		5.12.3.3	operator=(const TALLAGH_W &orig)	39
		5.12.3.4	toString()	39
5.13	TM Cla	ss Refere	nce	40
	5.13.1	Detailed I	Description	40
	5.13.2	Member I	Function Documentation	40
		5.13.2.1	print_all()	40

CONTENTS

TX Clas	ss Referer	nce	40
5.14.1	Detailed I	Description	41
5.14.2	Member I	Function Documentation	41
	5.14.2.1	load(std::shared_ptr< OSTM > object)	41
5.14.3	Friends A	and Related Function Documentation	41
	5.14.3.1	TM	41
5.14.4	Member I	Data Documentation	41
	5.14.4.1	test_counter	41
ULSTE	R Class R	eference	42
5.15.1	Detailed I	Description	43
5.15.2	Construct	tor & Destructor Documentation	43
	5.15.2.1	ULSTER()	43
	5.15.2.2	ULSTER(int accountNumber, double balance, std::string firstName, std::string lastName, std::string address)	43
	5.15.2.3	$\label{eq:ULSTER} {\sf ULSTER}({\sf std::shared_ptr} < {\sf BANK} > {\sf obj, int_version, int_unique_id}) $	43
	5.15.2.4	ULSTER(const ULSTER &orig)	43
	5.15.2.5	\sim ULSTER()	43
5.15.3	Member I	Function Documentation	43
	5.15.3.1	$\label{eq:copy} \mbox{copy(std::shared_ptr} < \mbox{OSTM} > \mbox{to, std::shared_ptr} < \mbox{OSTM} > \mbox{from)} . \ . \ . \ . \ . \ .$	43
	5.15.3.2	$getBaseCopy(std::shared_ptr < OSTM > object) \ \ . \ \ \ \ . \ \ . \ \ . \ \ . \ \ . \ \ \ . \ \ . \ \ . \ \ . \ \ . \ \ \ \ . \ \ . \ \ . \ \ . \ \ . \ \ . \ \ . \ \ . \ \ . \ \ . \ \$	44
	5.15.3.3	operator=(const ULSTER &orig)	44
	5.15.3.4	toString()	44
UNBL (Class Refe	rence	45
5.16.1	Detailed I	Description	45
5.16.2	Construct	tor & Destructor Documentation	46
	5.16.2.1	UNBL()	46
	5.16.2.2	UNBL(int accountNumber, double balance, std::string firstName, std::string last ← Name, std::string address)	46
	5.16.2.3	$\label{eq:unblumble} {\tt UNBL(std::shared_ptr}{<}~{\tt BANK} > {\tt obj, int_version, int_unique_id})~\dots~\dots~.$	46
	5.16.2.4	UNBL(const UNBL &orig)	46
	5.16.2.5	\sim UNBL()	46
5.16.3	Member I	Function Documentation	46
	5.16.3.1	$\label{eq:copy} \mbox{copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from)} $	46
	5.16.3.2	${\tt getBaseCopy(std::shared_ptr} < {\tt OSTM} > {\tt object)} \ \ \ldots \ \ldots \ \ldots \ \ldots \ \ldots$	47
	5.16.3.3	operator=(const UNBL &orig)	47
	5.16.3.4	toString()	47
WARE	HOUSE CI	lass Reference	47
5.17.1	Detailed I	Description	48
5.17.2	Construct	tor & Destructor Documentation	48
	5.17.2.1	WAREHOUSE()	48
	5.17.2.2	WAREHOUSE(int _version, int _unique_id)	48
	5.17.2.3	WAREHOUSE(const WAREHOUSE &orig)	48
	5.17.2.4	\sim WAREHOUSE()	49
	5.14.1 5.14.2 5.14.3 5.14.4 ULSTE 5.15.1 5.15.2 5.15.3 UNBL 0 5.16.1 5.16.2	5.14.1 Detailed II 5.14.2 Member II 5.14.3 Friends A 5.14.3.1 5.14.4 Member II 5.14.4.1 ULSTER Class R 5.15.1 Detailed II 5.15.2 Construct 5.15.2.3 5.15.2.4 5.15.2.5 5.15.3 Member II 5.16.3.1 5.16.2.2 5.16.2.1 5.16.2.2 5.16.2.1 5.16.2.2 5.16.2.3 5.16.2.4 5.16.2.2 5.16.3.3 5.16.3.4 WAREHOUSE Cl 5.17.2.1 5.17.2.2 5.17.2.3	5.15.2.3 ULSTER(std::shared_ptr< BANK > obj, int_version, int_unique_id) 5.15.2.4 ULSTER() 5.15.2.5 ~ULSTER() 5.15.3.1 copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from) 5.15.3.2 getBaseCopy(std::shared_ptr< OSTM > object) 5.15.3.3 operator=(const ULSTER &orig) 5.15.3.4 toString() UNBL Class Reference 5.16.1 Detailed Description 5.16.2.2 UNBL() 5.16.2.2 UNBL() 5.16.2.3 UNBL(std::shared_ptr< BANK > obj, int_version, int_unique_id) 5.16.2.4 UNBL(const UNBL &orig) 5.16.2.5 ~UNBL() 5.16.3.1 copy(std::shared_ptr< BANK > obj, int_version, int_unique_id) 5.16.2.5 ~UNBL() 5.16.3.1 copy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from) 5.16.3.2 getBaseCopy(std::shared_ptr< OSTM > to, std::shared_ptr< OSTM > from) 5.16.3.3 operator=(const UNBL &orig) 5.16.3.4 toString() WAREHOUSE Class Reference 5.17.1 Detailed Description 5.17.2 Constructor & Destructor Documentation 5.16.3.1 copyright: Shared_ptr< OSTM > object) 5.16.3.3 operator=(const UNBL &orig) 5.16.3.4 toString()

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 \leftarrow

Start values: Double value: 500 to each.

Nesting level 4.
After transactions:

4 README

- 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:

OSTM	30
BANK	12
AIB	ç
BOA	13
BOI	16
SWBPLC	34
ULSTER	42
UNBL	45
WAREHOUSE	47
CARLOW_W	19
CARPHONE_WAREHOUSE	22
DUNDALK_W	24
KILKENNY_W	27
SLIGO_W	31
TALLAGH_W	37
TM	40
TX	40

6 Hierarchical Index

Chapter 4

Class Index

4.1 Class List

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

AIB																							
BANK																							
BOA																							
BOI																							
CARLOW_W																							
CARPHONE_V																							
DUNDALK_W																							
${\bf KILKENNY_W}$																							
OSTM											 	 											30
SLIGO_W																							
SWBPLC											 	 											34
TALLAGH_W																							37
TM																							40
TX																							
ULSTER																							
UNBL											 	 											45
WAREHOUSE											 	 											47

8 Class Index

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::\simAIB( ) [virtual]
```

de-constructor

Definition at line 17 of file AIB.cpp.

5.1.3 Member Function Documentation

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

copy function, make deep copy of the object/pointer

5.1 AIB Class Reference 11

Parameters

objTO	is a std::shared_ptr <bank> type object casted back from std::shared_ptr<ostm></ostm></bank>
objFROM	is a std::shared_ptr <bank> type object casted back from std::shared_ptr<ostm></ostm></bank>

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

Definition at line 37 of file AIB.cpp.

```
5.1.3.2 std::shared_ptr< OSTM > AlB::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</bank>

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.

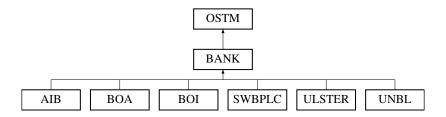
The documentation for this class was generated from the following files:

- /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.

5.3 BOA Class Reference

```
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::\simBANK( ) [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 \sim **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 BOA Class Reference 15

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></ostm></bank>
objFROM	is a std::shared_ptr <bank> type object casted back from std::shared_ptr<ostm></ostm></bank>

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</bank>

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

5.4 BOI Class Reference 17

- 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></ostm>
objFROM	is a BANK* type object casted back from std::shared_ptr <ostm></ostm>

Reimplemented from OSTM (p. 30).

Definition at line 35 of file BOI.cpp.

```
5.4.3.2 std::shared_ptr< OSTM > BOl::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
ſ	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.

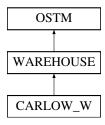
The documentation for this class was generated from the following files:

- /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></ostm>
objFROM	is a BANK* type object casted back from std::shared_ptr <ostm></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></ostm>
objFROM	is a BANK* type object casted back from std::shared_ptr <ostm></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
Ī	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.

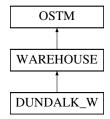
The documentation for this class was generated from the following files:

- /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></ostm>
objFROM	is a BANK* type object casted back from std::shared_ptr <ostm></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.

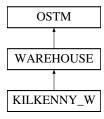
The documentation for this class was generated from the following files:

- /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></ostm>
objFROM	is a BANK* type object casted back from std::shared_ptr <ostm></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).

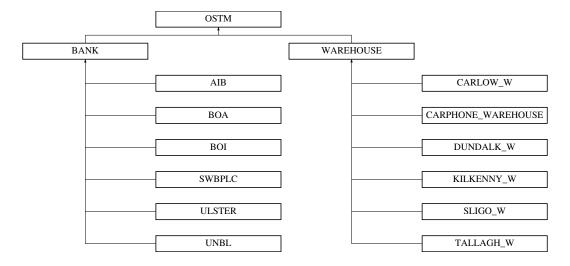
Definition at line 62 of file KILKENNY W.cpp.

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 \sim 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.

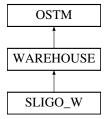
The documentation for this class was generated from the following file:

/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></ostm>
objFROM	is a BANK* type object casted back from std::shared_ptr <ostm></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.

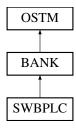
The documentation for this class was generated from the following files:

- /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></ostm></bank>
objFROM	is a std::shared_ptr <bank> type object casted back from std::shared_ptr<ostm></ostm></bank>

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</bank>

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.

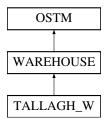
The documentation for this class was generated from the following files:

- /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></ostm>
objFROM	is a BANK* type object casted back from std::shared_ptr <ostm></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.

• \sim 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< ${\bf OSTM} > {\bf load}$ (std::shared_ptr< ${\bf OSTM} > {\bf object})$

5.14 TX Class Reference 41

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.

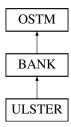
The documentation for this class was generated from the following file:

/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></ostm></bank>
objFROM	is a std::shared_ptr <bank> type object casted back from std::shared_ptr<ostm></ostm></bank>

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</bank>

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:

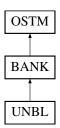
- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/← Test01/ULSTER.h
- /media/zoltan/Data/00_2018_ITCarlow/00_Modules/06_Project/Documents/Git_Sync/Main Tests for Linux/← Test01/ULSTER.cpp

5.16 UNBL Class Reference 45

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 ${f copy}$ (std::shared_ptr< ${f OSTM}$ > to, std::shared_ptr< ${f OSTM}$ > 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></ostm></bank>	
objFROM	is a std::shared_ptr <bank> type object casted back from std::shared_ptr<ostm></ostm></bank>	

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</bank>	

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.

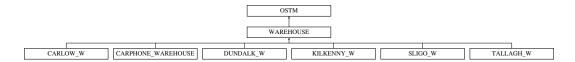
The documentation for this class was generated from the following files:

- /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