

NHF

Sepsi László
K6NNN5
Szerda Május 8 2019

Tartalomjegyzék

Tartalom

Tartalomjegyzék	ii
Osztálymutató	2
Osztálylista	2
Fájlmutató	3
Fájllista	3
Osztályok dokumentációja	4
Auction osztályreferencia	4
Publikus tagfüggvények	4
Privát tagfüggvények	4
Privát attribútumok	5
Részletes leírás	5
Konstruktorok és destruktorok dokumentációja	5
Tagfüggvények dokumentációja	5
Adattagok dokumentációja	7
Customer osztályreferencia	9
Publikus tagfüggvények	9
Privát tagfüggvények	9
Privát attribútumok	9
Részletes leírás	10
Konstruktorok és destruktorok dokumentációja	10
Tagfüggvények dokumentációja	10
Adattagok dokumentációja	12
Item osztályreferencia	13
Publikus tagfüggvények	13
Privát attribútumok	13
Részletes leírás	13
Konstruktorok és destruktorok dokumentációja	13
Tagfüggvények dokumentációja	14
Adattagok dokumentációja	14
Vektor< T > osztálysablon-referencia	15
Publikus tagfüggvények	15
Privát attribútumok	15
Részletes leírás	16
Konstruktorok és destruktorok dokumentációja	16
Tagfüggvények dokumentációja	16
Adattagok dokumentációja	18
Fájlok dokumentációja	19
arveres.cpp fájlreferencia	19
Függvények	19
Változók	19
Függvények dokumentációja	19
Változók dokumentációja	21
arveres.h fájlreferencia	22
Osztályok	22
Enumerációk	22
Függvények	22
Enumerációk dokumentációja	22
Függvények dokumentációja	23
auction.cpp fájlreferencia	25
customer.cpp fájlreferencia	26
main.cpp fájlreferencia	27
Függvények	27

Változók	27
Függvények dokumentációja	27
Változók dokumentációja	27
Tárgymutató	28

Osztálymutató

Osztálylista

Az összes osztály, struktúra, unió és interfész listája rövid leírásokkal:

Auction (Osztály az aukció modellezésére)	4
Customer (Osztály egy vásárló modellezésére)	9
Item (Osztály egy tárgy modellezésére)	13
Vektor< T > (Egy template osztály megadható típusú elemek tárolására, vektorban, listafunkciókkal)	15

Fájlmutató

Fájllista

Az összes fájl listája rövid leírásokkal:

arveres.cpp	19
arveres.h	22
auction.cpp	25
customer.cpp	26
main.cpp	27

Osztályok dokumentációja

Auction osztályreferencia

Osztály az aukció modellezésére.

```
#include <arveres.h>
```

Publikus tagfüggvények

- **Auction ()**
Létrehoz egy aukció modellt.
- **void AddCustomer (const Customer &c)**
Vásárlót vesz fel az osztály vektorába.
- **void AddItemForSale (const Item &i)**
Fel vesz egy tárgyat az eleadandó tárgyak vektorába.
- **void Start ()**
Elindítja az árverést.
- **Vektor< Customer > getCustomers ()**
Egy másolatot ad vissza a vevőket tároló vektorról.
- **Vektor< Item > getItemsForsale ()**
Egy másolatot ad vissza az eladandó árukat tároló vektorról.
- **Vektor< Item > getItemsSold ()**
Egy másolatot ad vissza az eladott árukat tároló vektorról.
- **Vektor< Item > getItemsLeft ()**
Egy másolatot ad vissza a megmaradt árukat tároló vektorról.
- **~Auction ()**
Destruktor.

Privát tagfüggvények

- **void sellItem (Item item)**
A megfelelő árverezési módot hívja meg az adott árura.
- **void EnglisAuction (Item item)**
Angol akció elvégzése.
- **void DeutchAuction (Item item)**
Német aukció elvégzése.

- void **BlindAuction** (Item item)
Vak aukció elvégzése.
- void **transferItemToCustomer** (Item item, Customer &customer, double price)
A paraméterként megadott árut átadja a megfelelő vásárlónak ,akitől levonja a megadott árat.

Privát attribútumok

- Vektor< Customer > customers
- Vektor< Item > itemsForsale
- Vektor< Item > itemsSold
- Vektor< Item > itemsLeft

Részletes leírás

Oszály az aukció modellezésére.

Konstruktorok és destruktorok dokumentációja

Auction::Auction ()

Létrehoz egy aukció modellt.

Auction::~~Auction () [inline]

Destruktor.

Paraméterek:

--	--

Tagfüggvények dokumentációja

void Auction::AddCustomer (const Customer & c)

Vásárlót vesz fel az osztály vektorába.

Paraméterek:

Customer&c	const
------------	-------

Visszatérési érték:

void

void Auction::AddItemForSale (const Item & i)

Fel vesz egy tárgyat az eleadandó tárgyak vektorába.

Paraméterek:

<i>i</i>	const Item &
----------	---------------------

Visszatérési érték:

void

void Auction::BlindAuction (Item *item*) [private]

Vak aukció elvégzése.

Paraméterek:

<i>item</i>	Item
-------------	-------------

Visszatérési érték:

void

void Auction::DeutchAuction (Item *item*) [private]

Német aukció elvégzése.

Paraméterek:

<i>item</i>	Item
-------------	-------------

Visszatérési érték:

void

void Auction::EnglisAuction (Item *item*) [private]

Angol aukció elvégzése.

Paraméterek:

<i>item</i>	Item
-------------	-------------

Visszatérési érték:

void

Vektor< Customer > Auction::getCustomers ()

Egy másolatot ad vissza a vevőket tároló vektorról.

Vektor< Item > Auction::getItemsForsale ()

Egy másolatot ad vissza az eladandó árukat tároló vektorról.

Vektor< Item > Auction::getItemsLeft ()

Egy másolatot ad vissza a megmaradt árukat tároló vektorról.

Vektor< Item > Auction::getItemsSold ()

Egy másolatot ad vissza az eladott árukat tároló vektorról.

void Auction::sellItem (Item item) [private]

A megfelelő árvezetési módot hívja meg az adott árura.

Paraméterek:

<i>item</i>	Item
-------------	-------------

Visszatérési érték:

void

void Auction::Start ()

Elindítja az árverést.

Visszatérési érték:

void

void Auction::transferItemToCustomer (Item item, Customer & customer, double price) [private]

A paraméterként megadott árut átadja a megfelelő vásárlónak ,akitől levonja a megadott árat.

Paraméterek:

<i>item</i>	Item
<i>customer</i>	Customer&
<i>price</i>	double

Visszatérési érték:

void

Adattagok dokumentációja

Vektor<Customer> Auction::customers [private]

Vektor<Item> Auction::itemsForsale [private]

Vektor<Item> Auction::itemsLeft [private]

Vektor<Item> Auction::itemsSold [private]

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- `arveres.h`
- `auction.cpp`

Customer osztályreferencia

Oszály egy vásárló modellezésére.

```
#include <arveres.h>
```

Publikus tagfüggvények

- **Customer** (string **name**="Generic_Name", double **balance**=1000)
Konstruktor default paraméterekkel ellátva, létrehozás után az értékek nem változtathatók.
- void **AddItem** (const **Item** &i)
A vásárlóhoz tartozó tárgy tárolóhoz ad egy tárgyat.
- **Item PopItem** ()
A vásárló utolsó tárgyát törli és egy másolatot ad vissza róla.
- string **getName** ()
Lekérdezi a vásárló nevét.
- double **getBalance** ()
lekérdezi a vásárló egyenlegét
- **Vektor< Item > getItems** ()
Egy másolatot ad a vásárló tárgyairól.
- double **Bid** (double askingprice)
A vásárló eldönti, hogy szeretne-e licitálni, ha igen egy ajánlatot ad vissza, ha nem akkor 0-át.
- void **Pay** (double price)
A vásárló egyenlegéből levonásra kerül a megadott összeg.
- bool **AcceptOffer** (double price)

Privát tagfüggvények

- double **fRand** (double fMin, double fMax)
ffMin-fMax tartományban generál egy véletlen számot
- bool **isIntrested** ()
Igaz érték ad vissza, ha a vásárló szeretne résztvenni az árverésen.

Privát attribútumok

- string **name**
 - double **balance**
 - **Vektor< Item > items**
-

Részletes leírás

Oszály egy vásárló modellezésére.

Konstruktorok és destruktorok dokumentációja

Customer::Customer (string *name* = "Generic_Name", double *balance* = 1000)

Konstruktor default paraméterekkel ellátva, létrehozás után az értékek nem változtathatók.

Paraméterek:

<i>Generic_Name</i>	string name=
<i>1000</i>	double balance=

Tagfüggvények dokumentációja

bool Customer::AcceptOffer (double *price*)

void Customer::AddItem (const Item & *i*)

A vásárlóhoz tartozó tárgy tárolóhoz ad egy tárgyat.

Paraméterek:

<i>i</i>	const Item&
----------	-------------

Visszatérési érték:

void

double Customer::Bid (double *askingprice*)

A vásárló eldönti, hogy szeretne-e licitálni, ha igen egy ajánlatot ad vissza, ha nem akkor 0-át.

Paraméterek:

<i>askingprice</i>	double
--------------------	--------

Visszatérési érték:

double

double Customer::fRand (double *fMin*, double *fMax*) [private]

fMin-fMax tartományban generál egy véletlen számot

Paraméterek:

<i>fMin</i>	double
<i>fMax</i>	double

Visszatérési érték:

double

double Customer::getBalance ()

lekérdezi a vásárló egyenlegét

Visszatérési érték:

double

Vektor< Item > Customer::getItems ()

Egy másolatot ad a vásárló tárgyairól.

Visszatérési érték:

Vektor<Item>

string Customer::getName ()

Lekérdezi a vásárló nevét.

Visszatérési érték:

string

bool Customer::isIntrested () [private]

Igaz értéket ad vissza, ha a vásárló szeretne résztvenni az árverésen.

Visszatérési érték:

bool

void Customer::Pay (double *price*)

A vásárló egyenlegéből levonásra kerül a megadott összeg.

Paraméterek:

<i>price</i>	double
--------------	--------

Visszatérési érték:

void

Item Customer::PopItem ()

A vásárló utolsó tárgyát törli és egy másolatot ad vissza róla.

Visszatérési érték:

Item

Adattagok dokumentációja

`double Customer::balance [private]`

`Vektor<Item> Customer::items [private]`

`string Customer::name [private]`

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- `arveres.h`
- `customer.cpp`

Item osztályreferencia

Osztály egy tárgy modellezésére.
`#include <arveres.h>`

Publikus tagfüggvények

- **Item** (string **name**="Generic_item", double **price**=10, string **description**="Generic_description", SellingMetod **sellingMethod**=English)
Konstruktor default paraméterekkel ellátva, létrehozás után az értékek nem változtathatók.
- string **getName** ()
Lekérdezi a tárgy nevét.
- double **getPrice** ()
Lekérdezi a tárgy árát.
- string **getDescription** ()
Lekérdezi a tárgy leírását.
- **SellingMetod** **getSellingMethod** ()
Lekérdezi a tárgy eladási módját.

Privát attribútumok

- string **name**
- double **price**
- string **description**
- **SellingMetod** **sellingMethod**

Részletes leírás

Osztály egy tárgy modellezésére.

Konstruktorok és destruktorok dokumentációja

Item::Item (string **name** = "Generic_item", double **price** = 10, string **description** = "Generic_description", **SellingMetod** **sellingMethod** = English)

Konstruktor default paraméterekkel ellátva, létrehozás után az értékek nem változtathatók.

Paraméterek:

<i>Generic_item</i>	string name=
<i>10</i>	double price=
<i>Generic_description</i>	string description=
<i>English</i>	SellingMetod sellingMethod=

Tagfüggvények dokumentációja

string Item::getDescription ()

Lekérdezi a tárgy leírását.

Visszatérési érték:

string

string Item::getName ()

Lekérdezi a tárgy nevét.

Visszatérési érték:

string

double Item::getPrice ()

Lekérdezi a tárgy árát.

Visszatérési érték:

double

SellingMetod Item::getSellingMethod ()

Lekérdezi a tárgy eladási módját.

Visszatérési érték:

SellingMetod

Adattagok dokumentációja

string Item::description [private]

string Item::name [private]

double Item::price [private]

SellingMetod Item::sellingMethod [private]

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- arveres.h

Vektor< T > osztálysablon-referencia

Egy template osztály megadható típusú elemek tárolására, vektorban, listafunkciókkal.
`#include <arveres.h>`

Publikus tagfüggvények

- **Vektor** (size_t meret=0)
Létrehoz egy tárolót a megadott mérettel.
- **Vektor** (const **Vektor** &v)
Másolókonstruktor.
- **Vektor & operator=** (const **Vektor** &v)
Értékadó operátor.
- size_t **size** () const
Lekérdezi a tároló méretét.
- T & **operator[]** (size_t index)
Indexelő operátor nem konstans adattagokra.
- T const & **operator[]** (size_t index) const
Indexelőoperátor konstans adattagokra.
- void **Add** (T const &new_val)
Új elemet ad a vektorhoz.
- void **Delete** (size_t index)
Töri a paraméterként kapott indexű elemet.
- T **Pop** ()
Az utolsó elemet töri és a másolatát visszaszadja.
- ~**Vektor** ()
Felszabadítja a foglalt memóriát.

Privát attribútumok

- T * **data**
 - size_t **siz**
-

Részletes leírás

`template<typename T>`

`class Vektor< T >`

Egy template osztály megadható típusú elemek tárolására, vektorban, listafunkciókkal.

Paraméterek:

--	--

Konstruktorok és destruktorok dokumentációja

`template<typename T > Vektor< T >::Vektor (size_t meret = 0)[explicit]`

Létrehoz egy tárolót a megadott mérettel.

Paraméterek:

<i>0</i>	size_t <i>meret</i> =
----------	-----------------------

Visszatérési érték:

explicit

`template<typename T > Vektor< T >::Vektor (const Vektor< T > & v)`

Másolókonstruktor.

Paraméterek:

<i>v</i>	const Vektor &
----------	-----------------------

`template<typename T > Vektor< T >::~~Vektor ()`

Felszabadítja a foglalt memóriát.

Tagfüggvények dokumentációja

`template<typename T> void Vektor< T >::Add (T const & new_val)`

Új elemet ad a vektorhoz.

Paraméterek:

<i>new_val</i>	T const&
----------------	----------

Visszatérési érték:

void

template<typename T > void Vektor< T >::Delete (size_t index)

Törli a paraméterként kapott indexű elemet.

Paraméterek:

<i>index</i>	size_t
--------------	--------

Visszatérési érték:

void

template<typename T > Vektor< T > & Vektor< T >::operator= (const Vektor< T > & v)

Értékadó operátor.

Paraméterek:

v	const Vektor&
---	---------------

Visszatérési érték:

Vektor& operator

template<typename T > T & Vektor< T >::operator[] (size_t index)

Indexelő operátor nem konstans adattagokra.

Paraméterek:

<i>index</i>	size_t
--------------	--------

Visszatérési érték:

T&

template<typename T > T const & Vektor< T >::operator[] (size_t index) const

Indexelőoperátor konstans adattagokra.

Paraméterek:

<i>index</i>	size_t
--------------	--------

Visszatérési érték:

T const&

template<typename T > T Vektor< T >::Pop ()

Az utolsó elemet törli és a másolatát visszaszadja.

Visszatérési érték:

T

template<typename T > size_t Vektor< T >::size () const

Lekérdezi a tároló méretét.

Visszatérési érték:

size_t

Adattagok dokumentációja

template<typename T> T* Vektor< T >::data [private]

template<typename T> size_t Vektor< T >::siz [private]

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- arveres.h

Fájlok dokumentációja

arveres.cpp fájlreferencia

```
#include "arveres.h"
```

Függvények

- ostream & **operator<<** (ostream &os, **Item** i)
inserter operátor egy tárgy kiírásához
- ostream & **operator<<** (ostream &os, **Customer** c)
inserter operátor egy vásárló kiírásához
- ostream & **operator<<** (ostream &os, **SellingMetod** s)
inserter operátor egy eladási mód kiírásához
- ostream & **operator<<** (ostream &os, **Vektor< Item >** items)
inserter operátor egy tárgy vektor kiírásához
- ostream & **operator<<** (ostream &os, **Vektor< Customer >** customers)
inserter operátor egy vásárló vektor kiírásához
- ostream & **operator<<** (ostream &os, **Auction** a)
inserter operátor egy aukció kiírásához

Változók

- string **separator** ="-----"
- string **MethodNames** [] = { { "English" }, { "Dutch" }, { "Blind" } }

Függvények dokumentációja

ostream& operator<< (ostream & os, Item i)

inserter operátor egy tárgy kiírásához

Paraméterek:

<i>os</i>	ostream&
<i>i</i>	Item

Visszatérési érték:

ostream& operator

ostream& operator<< (ostream & os, Customer c)

inserter operátor egy vásárló kiírásához

Paraméterek:

<i>os</i>	ostream&
<i>c</i>	Customer

Visszatérési érték:

ostream& operator

ostream& operator<< (ostream & os, SellingMetod s)

insertter operátor egy eladási mód kiírásához

Paraméterek:

<i>os</i>	ostream&
<i>s</i>	SellingMetod

Visszatérési érték:

ostream& operator

ostream& operator<< (ostream & os, Vektor< Item > items)

insertter operátor egy tárgy vektor kiírásához

Paraméterek:

<i>os</i>	ostream&
<i>Vektor<Item>item</i>	
<i>s</i>	

Visszatérési érték:

ostream& operator

ostream& operator<< (ostream & os, Vektor< Customer > items)

insertter operátor egy vásárló vektor kiírásához

Paraméterek:

<i>os</i>	ostream&
<i>Vektor<Customer>items</i>	

Visszatérési érték:

ostream& operator

ostream& operator<< (ostream & os, Auction a)

insertter operátor egy aukció kiírásához

Paraméterek:

<i>os</i>	ostream&
<i>a</i>	Auction

Visszatérési érték:

ostream& operator

Változók dokumentációja

```
string MethodNames[] = { { "English" }, { "Dutch" }, { "Blind" } }
```

```
string separator ="-----"
```


arveres.h fájlreferencia

```
#include "arveres.h"
#include <iostream>
#include <string>
#include "memtrace.h"
```

Osztályok

- class **Item**
Osztály egy tárgy modellezésére.
- class **Vektor< T >**
Egy template osztály megadható típusú elemek tárolására, vektorban, listafunkciókkal.
- class **Customer**
Osztály egy vásárló modellezésére.
- class **Auction**
Osztály az aukció modellezésére.

Enumerációk

- enum **SellingMetod** { **English**, **Dutch**, **Blind** }

Függvények

- ostream & **operator<<** (ostream &os, **Item** i)
inserter operátor egy tárgy kiírásához
- ostream & **operator<<** (ostream &os, **Customer** c)
inserter operátor egy vásárló kiírásához
- ostream & **operator<<** (ostream &os, **SellingMetod** s)
inserter operátor egy eladási mód kiírásához
- ostream & **operator<<** (ostream &os, **Auction** a)
inserter operátor egy aukció kiírásához
- ostream & **operator<<** (ostream &os, **Vektor< Item >**items)
inserter operátor egy tárgy vektor kiírásához
- ostream & **operator<<** (ostream &os, **Vektor< Customer >**items)
inserter operátor egy vásárló vektor kiírásához

Enumerációk dokumentációja

enum SellingMetod

Enumeráció-értékek:

English	
Dutch	

Blind	
-------	--

Függvények dokumentációja

ostream& operator<< (ostream & os, Item i)

inserter operátor egy tárgy kiírásához

Paraméterek:

<i>os</i>	ostream&
<i>i</i>	Item

Visszatérési érték:

ostream& operator

ostream& operator<< (ostream & os, Customer c)

inserter operátor egy vásárló kiírásához

Paraméterek:

<i>os</i>	ostream&
<i>c</i>	Customer

Visszatérési érték:

ostream& operator

ostream& operator<< (ostream & os, SellingMetod s)

inserter operátor egy eladási mód kiírásához

Paraméterek:

<i>os</i>	ostream&
<i>s</i>	SellingMetod

Visszatérési érték:

ostream& operator

ostream& operator<< (ostream & os, Auction a)

inserter operátor egy aukció kiírásához

Paraméterek:

<i>os</i>	ostream&
<i>a</i>	Auction

Visszatérési érték:

ostream& operator

ostream& operator<< (ostream & os, Vektor< Item > items)

inserter operátor egy tárgy vektor kiírásához

Paraméterek:

<i>os</i>	ostream&
<i>Vektor<Item>item</i> <i>s</i>	

Visszatérési érték:

ostream& operator

ostream& operator<< (ostream & os, Vektor< Customer > items)

inserter operátor egy vásárló vektor kiírásához

Paraméterek:

<i>os</i>	ostream&
<i>Vektor<Customer</i> <i>>items</i>	

Visszatérési érték:

ostream& operator

auction.cpp fájlreferencia

```
#include "arveres.h"
```

customer.cpp fájlreferencia

```
#include "arveres.h"  
#include <stdlib.h>  
#include <time.h>
```

main.cpp fájlreferencia

```
#include <iostream>
#include "arveres.h"
#include <stdlib.h>
#include <time.h>
#include <string>
#include "gtest_lite.h"
#include "memtrace.h"
```

Függvények

- void **PrintOptions** ()
- **Item** **getItemFromConsole** ()
- **Customer** **getCustomerFromConsole** ()
- void **SeedAuction** ()
- void **UserInteractionHandler** ()
- void **Test** ()
- int **main** ()

Változók

- Auction **auction**
-

Függvények dokumentációja

Customer **getCustomerFromConsole** ()

Item **getItemFromConsole** ()

int **main** ()

void **PrintOptions** ()

void **SeedAuction** ()

void **Test** ()

void **UserInteractionHandler** ()

Változók dokumentációja

Auction **auction**

Tárgymutató

INDEX