CppBolt

DoxyGen-nel generált dokumentáció (rtf->pdf)
Kézzel picit rövidítve, minimális mértékben csinosítva

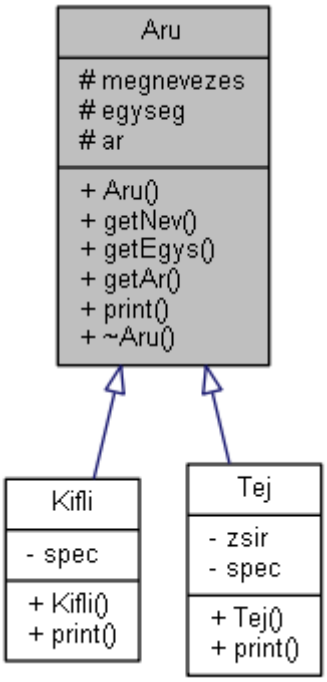
Osztályok dokumentációja

Aru osztályreferencia

Aru osztály.

`#include <aru.h>`

Az Aru osztály származási diagramja:



Publikus tagfüggvények

- Aru (const char *nev, const char *egyseg, double ar)
  Konstruktor beállítja az attribútumokat.
- String getNev () const
  Megnevezés lekérdezése.
- String getEgys () const
  Mennyiségi egység lekérdezése.
- double getAr () const
  Egységár lekérdezése.
- virtual std::ostream & print (std::ostream &os) const
  Attribútumok kiírása egy stream-re.
- virtual ~Aru ()
  Virtuális destruktor.

Védett attribútumok

- String megnevezes
  áru megnevezése.
- String egyseg
  áru mennyiségi egysége (db, liter, kg, ...)
- double ar
  áru egységára.

Részletes leírás

Aru osztály.

Konstruktorok és destruktorok dokumentációja

Aru::Aru (const char * *nev*, const char * *egyseg*, double *ar*) [inline]

Konstruktor beállítja az attribútumokat.

Paraméterek:

nev	- áru megnevezése
egység	- áru mennyiségi egysége szövegesen (db, liter, kg, ...)
ar	- áru egységára

Tagfüggvények dokumentációja

double Aru::getAr () const [inline]

Egységár lekérdezése.

Visszatérési érték:

- egységár

String Aru::getEgys () const [inline]

Mennyiségi egység lekérdezése.

Visszatérési érték:

- mennyiségi egység

String Aru::getNev () const [inline]

Megnevezés lekérdezése.

Visszatérési érték:

- megnevezés

virtual std::ostream& Aru::print (std::ostream & *os*) const [inline], [virtual]

Attribútumok kiírása egy stream-re.

Paraméterek:

os	- output stream referencia

Visszatérési érték:

output stream referencia

Újraimplementáló leszármazottak: Tej és Kifli.

Datum osztályreferencia

Publikus tagfüggvények

- Datum ()
  Paraméter nélkül hívható konstruktor.
- Datum (int ev, int ho, int nap)
  Adott napot beállító konstruktor.
- int getEv () const
  Év lekérdezése.
- int getHo () const
  Hónap lekérdezése.
- int getNap () const
  Nap lekérdezése.
- bool operator== (const Datum &d) const
  Két dátum egyezõségét vizsgálja.

Privát attribútumok

- int ev
  év.
- int ho
  hónap
- int nap
  nap

Konstruktorok és destruktorok dokumentációja

Datum::Datum ()

Paraméter nélkül hívható konstruktor.

Mai napot állítja be

Datum::Datum (int *ev*, int *ho*, int *nap*) `[inline]`

Adott napot beállító konstruktor.

Paraméterek:

ev	- év
hó	- hónap
nap	- nap

Tagfüggvények dokumentációja

int Datum::getEv () const `[inline]`

Év lekérdezése.

Visszatérési érték:

ev

int Datum::getHo () const `[inline]`

Hónap lekérdezése.

Visszatérési érték:

honap

int Datum::getNap () const `[inline]`

Nap lekérdezése.

Visszatérési érték:

nap

bool Datum::operator== (const Datum & *d*) const `[inline]`

Két dátum egyezõségét vizsgálja.

Paraméterek:

d	- jobb oldali operandus

Visszatérési érték:

true, ha egyezik a két dátum

Kassza osztályreferencia

Kassza osztály.

```
#include <kassza.h>
```

Publikus tagfüggvények

- Kassza ()
  taroló. Tetel-eket tárol
- void elad (double mennyi, const Aru &mit, const Datum &mikor=Datum())
  Eladás.
- void list (std::ostream &os) const
  Kassza tartalmának kilistázása.
- void list (std::ostream &os, const Datum &mikor) const
  Eladások listázása egy adott napon.
- double napiOsszeg (const Datum &mikor=Datum())
  Eladások összege egy adott napra.

Privát attribútumok

- size_t db
  tarolt darabszám
- Tetel **tetelek** [maxdb]

Statikus privát attribútumok

- static const size_t maxdb = 20
  méret

Részletes leírás

Kassza osztály.

Fix darabszámú Tetel tárolására képes. Ha betelik, const char* kivételt dob

Konstruktorok és destruktorok dokumentációja

Kassza::Kassza () [inline]

taroló. Tetel-eket tárol

Üres kassza

Tagfüggvények dokumentációja

void Kassza::elad (double *mennyi*, const Aru & *mit*, const Datum & *mikor* = Datum())

Eladás.

Paraméterek:

mennyi	- eladott mennyiség
mit	- referencia az eladott árura (Kompatibilitás kihasználása)
mikor	- eladás dátuma

Visszatérési érték:

kivételt dob, ha nem fér be

void Kassza::list (std::ostream & *os*) const

Kassza tartalmának kilistázása.

Paraméterek:

os	- output stream

void Kassza::list (std::ostream & *os*, const Datum & *mikor*) const

Eladások listázása egy adott napon.

Paraméterek:

os	- output stream
mikor	- melyik nap

double Kassza::napiOsszeg (const Datum & *mikor* = Datum())

Eladások összege egy adott napra.

Eladások összege egy adott nara.

Paraméterek:

mikor	- melyik nap

Visszatérési érték:

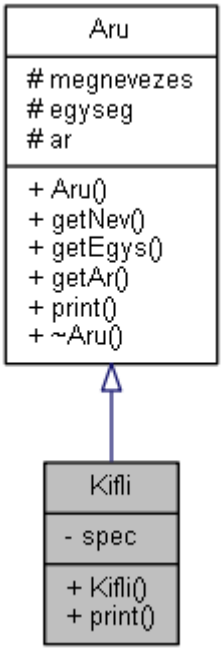
- összeg

Kifli osztályreferencia

Kifli osztály.

`#include <kifli.h>`

A Kifli osztály származási diagramja:



Publikus tagfüggvények

- Kifli (double ar, const char *spec="")
  Konstruktor beállítja az attribútumokat (õsosztályét is)
- std::ostream & print (std::ostream &os) const
  Attribútumok kiírása egy stream-re.

Privát attribútumok

- String spec
  kifli fajta

Részletes leírás

Kifli osztály.

spec attribútuma a kifli fajtája

Konstruktorok és destruktorok dokumentációja

Kifli::Kifli (double *ar*, const char * *spec* = "")[inline]

Konstruktor beállítja az attribútumokat (õsosztályét is)

Paraméterek:

ar	- kifli egységára
spec	- kifli fajtája

Tagfüggvények dokumentációja

std::ostream& Kifli::print (std::ostream & *os*) const `[inline]`, `[virtual]`

Attribútumok kiírása egy stream-re.

Paraméterek:

os	- output stream referencia

Visszatérési érték:

output stream referencia

Újraimplementált õsök: Aru.

String osztályreferencia

`#include <string5.h>`

Publikus tagfüggvények

- size_t size () const
  hossz lezáró nulla nélkül
- String ()
  Default konstruktor:
- const char * c_str () const
  C-stringet ad vissza.
- String (char ch)
  Konstruktor: egy char karakterbõl (createStrFromChar)
- String (const char *p)
  Konstruktor: egy nullával lezárt char sorozatból (createStringFromCharStr)
- String (const String &s1)
  MÁSOLÓ konstruktor, ami a createStringFromString-bõl keletkezett.
- ~String ()
  Destruktor (disposeString)
- void printDbg (const char *txt="") const
  Egyéb tagfüggvények:
- String & operator= (const String &rhs_s)
  Operátorok:
- String & **operator+=** (const String &rhs_s)
- String operator+ (const String &rhs_s) const
  Két Stringet összefûz (concatString)
- String operator+ (char rhs_c) const
  Sztrinhez karaktert összefûz (concatString)
- char & operator[] (unsigned int idx)
  A string egy megadott indexû elemének REFERENCIÁJÁVAL tér vissza.
- const char & operator[] (unsigned int idx) const
  A string egy megadott indexû elemének REFERENCIÁJÁVAL tér vissza.
- void **erase** ()

Privát attribútumok

- char * **pData**
- size_t len
  pointer az adatra

Konstruktorok és destruktorok dokumentációja

String::String (const String & *s1*)

MÁSOLÓ konstruktor, ami a createStringFromString-bõl keletkezett.

Paraméterek:

s1	- String, amibõl létrehozzuk az új String-et

Tagfüggvények dokumentációja

String String::operator+ (const String & *rhs_s*) const

Két Stringet összefûz (concatString)

Paraméterek:

rhs_s	- jobboldali String

Visszatérési érték:

új String, ami tartalmazza a két stringet egmás után

String String::operator+ (char *rhs_c*) const `[inline]`

Sztrinhez karaktert összefûz (concatString)

Paraméterek:

rhs_c	- jobboldali karakter

Visszatérési érték:

új String, ami tartalmazza a két sztringet egymás után

String& String::operator= (const String & *rhs_s*)

Operátorok:

Értékadó operátor is kell !

Paraméterek:

rhs_s	- jobboldali String

Visszatérési érték:

baoldali (módosított) string (referenciája)

char& String::operator[] (unsigned int *idx*)

A string egy megadott indexû elemének REFERENCIÁJÁVAL tér vissza.

charAtString-bõl keletkezett, de ezt bal oldalon is lehet használni

Paraméterek:

idx	- charakter indexe

Visszatérési érték:

karakter (referencia) Indexelési hiba esetén const char* kivételt dob (assert helyett).

const char& String::operator[] (unsigned int *idx*) const

A string egy megadott indexû elemének REFERENCIÁJÁVAL tér vissza.

charAtString-bõl keletkezett. Konstans stringre alkalmazható. Indexelési hiba esetén const char* kivételt dob (assert helyett).

Paraméterek:

idx	- karakter indexe

Visszatérési érték:

karakter (referencia) Indexelési hiba esetén const char* kivételt dob (assert helyett).

void String::printDbg (const char * *txt* = `""`) const `[inline]`

Egyéb tagfüggvények:

Kiírunk egy Stringet (debug célokra) (ez kész) Elötte kiírunk egy tetszõleges szöveget.

Paraméterek:

txt	- nullával lezárt szövegre mutató pointer

size_t String::size () const `[inline]`

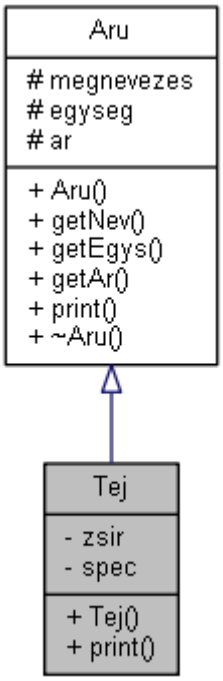
hossz lezáró nulla nélkül

Tej osztályreferencia

Tej osztály.

`#include <tej.h>`

A Tej osztály származási diagramja:



Publikus tagfüggvények

- Tej (double zsir, double ar, const char *spec="")
  Konstruktor beállítja az attribútumokat (õsosztályét is)
- std::ostream & print (std::ostream &os) const
  Attribútumok kiírása egy stream-re.

Privát attribútumok

- double zsir
  tej zsírtartalma %%)
- String spec
  tej fajtája

Additional Inherited Members

Részletes leírás

Tej osztály.

spec attribútuma a tej fajtája, és szírtartalma

Konstruktorok és destruktorok dokumentációja

Tej::Tej (double *zsir*, double *ar*, const char * *spec* = "") [inline]

Konstruktor beállítja az attribútumokat (õsosztályét is)

Paraméterek:

zsir	- tej zsírtartalma
ar	- tej egységára
spec	- tej fajtája

Tagfüggvények dokumentációja

std::ostream& Tej::print (std::ostream & *os*) const [inline], [virtual]

Attribútumok kiírása egy stream-re.

Paraméterek:

os	- output stream referencia

Visszatérési érték:

output stream referencia

Újraimplementált õsök: Aru.

Tetel struktúrareferencia

Tetel publikus osztály: Kassza ilyen tételeket tárol.

```
#include <kassza.h>
```

Publikus tagfüggvények

- **Tetel** (double mennyi, const Aru *aru, const Datum &d)

Publikus attribútumok

- double mennyiseg
  eladott áru mennyisége
- const Aru * aru
  eladott áru. POINTER az õs osztályra!
- Datum datum
  eladás dátuma

Részletes leírás

Tetel publikus osztály: Kassza ilyen tételeket tárol.

Kassza belsõ osztálya is lehetne, de talán így jobban olvasható