

Filmtár | Készítette: Funk Gábor

Készítette Doxygen 1.9.1

1. Névtérmutató	1
1.1. Névtérlista	1
2. Hierarchikus mutató	3
2.1. Osztályhierarchia	3
3. Osztálymutató	5
3.1. Osztálylista	5
4. Fájlmutató	7
4.1. Fájllista	7
5. Névterek dokumentációja	9
5.1. gtest_lite névtér-referencia	9
5.1.1. Részletes leírás	10
5.1.2. Függvények dokumentációja	10
5.1.2.1. almostEQ()	10
5.1.2.2. eq()	10
5.1.2.3. eqstr()	10
5.1.2.4. eqstrcase()	10
5.1.2.5. EXPECT_() [1/2]	11
5.1.2.6. EXPECT_() [2/2]	11
5.1.2.7. EXPECTSTR()	11
5.1.2.8. ge()	11
5.1.2.9. gt()	12
5.1.2.10. le()	12
5.1.2.11. lt()	12
5.1.2.12. ne()	12
5.1.2.13. nestr()	12
6. Osztályok dokumentációja	13
6.1. _Is_Types< F, T > struktúrasablon-referencia	13
6.1.1. Részletes leírás	13
6.1.2. Tagfüggvények dokumentációja	13
6.1.2.1. f() [1/2]	14
6.1.2.2. f() [2/2]	14
6.1.3. Adattagok dokumentációja	14
6.1.3.1. convertible	14
6.2. CsaladFilm osztályreferencia	14
6.2.1. Részletes leírás	15
6.2.2. Konstruktorok és destruktorkok dokumentációja	15
6.2.2.1. CsaladFilm()	15
6.2.3. Tagfüggvények dokumentációja	15
6.2.3.1. kiir()	15

6.2.3.2.	kiir_stream()	15
6.3.	DokumentumFilm osztályreferencia	16
6.3.1.	Részletes leírás	16
6.3.2.	Konstruktorok és destruktorok dokumentációja	16
6.3.2.1.	DokumentumFilm()	16
6.3.3.	Tagfüggvények dokumentációja	16
6.3.3.1.	kiir()	17
6.3.3.2.	kiir_stream()	17
6.4.	FileMegnyitasHiba osztályreferencia	17
6.4.1.	Konstruktorok és destruktorok dokumentációja	17
6.4.1.1.	FileMegnyitasHiba()	17
6.5.	Film osztályreferencia	18
6.5.1.	Részletes leírás	18
6.5.2.	Konstruktorok és destruktorok dokumentációja	19
6.5.2.1.	Film()	19
6.5.2.2.	~Film()	19
6.5.3.	Tagfüggvények dokumentációja	19
6.5.3.1.	getCim()	19
6.5.3.2.	getEv()	19
6.5.3.3.	kiir()	19
6.5.3.4.	kiir_stream()	20
6.5.3.5.	setCim()	20
6.5.3.6.	setEv()	20
6.5.4.	Barát és kapcsolódó függvények dokumentációja	20
6.5.4.1.	operator<<	20
6.5.5.	Adattagok dokumentációja	20
6.5.5.1.	cim	20
6.5.5.2.	kiadasiEv	21
6.5.5.3.	lejatszasildo	21
6.6.	FilmTar osztályreferencia	21
6.6.1.	Részletes leírás	21
6.6.2.	Konstruktorok és destruktorok dokumentációja	22
6.6.2.1.	~FilmTar()	22
6.6.3.	Tagfüggvények dokumentációja	22
6.6.3.1.	file_beolvas()	22
6.6.3.2.	file_ment()	22
6.6.3.3.	hozzaad()	22
6.6.3.4.	keres()	22
6.6.3.5.	kiir()	23
6.6.3.6.	mennyi_elem()	23
6.6.3.7.	modositas()	23
6.6.3.8.	torles()	23

6.7. gtest_lite::ostreamRedir osztályreferencia	23
6.7.1. Részletes leírás	24
6.7.2. Konstruktork és destruktorok dokumentációja	24
6.7.2.1. ostreamRedir()	24
6.7.2.2. ~ostreamRedir()	24
6.8. Tarolo< T > osztálysablon-referencia	24
6.8.1. Részletes leírás	24
6.8.2. Konstruktork és destruktorok dokumentációja	25
6.8.2.1. Tarolo()	25
6.8.2.2. ~Tarolo()	25
6.8.3. Tagfüggvények dokumentációja	25
6.8.3.1. hozzáad()	25
6.8.3.2. meret()	25
6.8.3.3. operator[]() [1/2]	26
6.8.3.4. operator[]() [2/2]	26
6.8.3.5. torol()	26
6.8.3.6. torol_egesz()	26
6.9. gtest_lite::Test struktúráreferencia	26
6.9.1. Részletes leírás	27
6.9.2. Konstruktork és destruktorok dokumentációja	27
6.9.2.1. ~Test()	28
6.9.3. Tagfüggvények dokumentációja	28
6.9.3.1. astatus()	28
6.9.3.2. begin()	28
6.9.3.3. end()	28
6.9.3.4. expect()	28
6.9.3.5. fail()	29
6.9.3.6. getTest()	29
6.9.4. Adattagok dokumentációja	29
6.9.4.1. ablocks	29
6.9.4.2. failed	29
6.9.4.3. name	29
6.9.4.4. null	29
6.9.4.5. os	30
6.9.4.6. status	30
6.9.4.7. sum	30
6.9.4.8. tmp	30
7. Fájlok dokumentációja	31
7.1. CsaladFilm.cpp fájlreferencia	31
7.2. CsaladFilm.hpp fájlreferencia	31
7.2.1. Részletes leírás	31

7.3. DinTomb.hpp fájlreferencia	32
7.3.1. Részletes leírás	32
7.4. DokumentumFilm.cpp fájlreferencia	32
7.5. DokumentumFilm.hpp fájlreferencia	32
7.5.1. Részletes leírás	33
7.6. Filehiba.hpp fájlreferencia	33
7.7. Film.cpp fájlreferencia	33
7.7.1. Függvények dokumentációja	33
7.7.1.1. operator<<()	33
7.8. Film.hpp fájlreferencia	34
7.8.1. Részletes leírás	34
7.9. Filmtar.cpp fájlreferencia	34
7.10. Filmtar.hpp fájlreferencia	34
7.10.1. Részletes leírás	35
7.11. gtest_lite.h fájlreferencia	35
7.11.1. Részletes leírás	38
7.11.2. Makródefiníciók dokumentációja	38
7.11.2.1. ADD_FAILURE	38
7.11.2.2. ASSERT_	39
7.11.2.3. ASSERT_EQ	39
7.11.2.4. ASSERT_NO_THROW [1/2]	39
7.11.2.5. ASSERT_NO_THROW [2/2]	39
7.11.2.6. ASSERTTHROW	39
7.11.2.7. CREATE_Has_	40
7.11.2.8. CREATE_Has_fn_	40
7.11.2.9. END	40
7.11.2.10. ENDM	40
7.11.2.11. ENDMsg	40
7.11.2.12. EXPECT_ANY_THROW	40
7.11.2.13. EXPECT_DOUBLE_EQ	40
7.11.2.14. EXPECT_ENVCASEEQ	41
7.11.2.15. EXPECT_ENVEQ	41
7.11.2.16. EXPECT_EQ	41
7.11.2.17. EXPECT_FALSE	41
7.11.2.18. EXPECT_FLOAT_EQ	41
7.11.2.19. EXPECT_GE	41
7.11.2.20. EXPECT_GT	42
7.11.2.21. EXPECT_LE	42
7.11.2.22. EXPECT_LT	42
7.11.2.23. EXPECT_NE	42
7.11.2.24. EXPECT_NO_THROW	42
7.11.2.25. EXPECT_STRCASEEQ	42

7.11.2.26.EXPECT_STRCASENE	42
7.11.2.27.EXPECT_STREQ	43
7.11.2.28.EXPECT_STRNE	43
7.11.2.29.EXPECT_THROW	43
7.11.2.30.EXPECT_THROW_THROW	43
7.11.2.31.EXPECT_TRUE	43
7.11.2.32.EXPECTTHROW	43
7.11.2.33.Nem célszerű közvetlenül használni, vagy módosítani	44
7.11.2.34.FAIL	44
7.11.2.35.GTEND	44
7.11.2.36.GTINIT	44
7.11.2.37.SUCCEED	44
7.11.2.38.TEST	44
7.11.3. Függvények dokumentációja	44
7.11.3.1. hasMember()	44
7.12. main.cpp fájlreferencia	45
7.12.1. Függvények dokumentációja	45
7.12.1.1. main()	45
7.13. memtrace.cpp fájlreferencia	45
7.14. memtrace.h fájlreferencia	45
7.15. Tarolo_impl.hpp fájlreferencia	45
Tárgymutató	47

1. fejezet

Névtérmutató

1.1. Névtérlista

Az összes névtér listája rövid leírásokkal:

gtest_lite	Gtest_lite: a keretrendszer függvényinek és objektumainak névtére	9
----------------------------	---	---

2. fejezet

Hierarchikus mutató

2.1. Osztályhierarchia

Majdnem (de nem teljesen) betűrendbe szedett leszármazási lista:

<code>_Is_Types< F, T ></code>	13
<code>Film</code>	18
<code>CsaladFilm</code>	14
<code>DokumentumFilm</code>	16
<code>FilmTar</code>	21
<code>gtest_lite::ostreamRedir</code>	23
<code>std::runtime_error</code>	
<code>FileMegnyitasHiba</code>	17
<code>Tarolo< T ></code>	24
<code>Tarolo< Film * ></code>	24
<code>gtest_lite::Test</code>	26

3. fejezet

Osztálymutató

3.1. Osztálylista

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

_Is_Types< F, T >	
Segédsablon típuskonverzió futás közbeni ellenőrzésere	13
CsaladFilm	
Speciális korhatáros film	14
DokumentumFilm	
Speciális film leírással	16
FileMegnyitasHiba	17
Film	
Az alap film base osztály. Tartalmazza a film címét, meddig tart és kiadási évét. (Megjelenése)	18
FilmTar	
A filmek tárolásáért és elérését biztosító osztály	21
gtest_lite::ostreamRedir	23
Tarolo< T >	
Egy alap generikus dinamikus tároló	24
gtest_lite::Test	26

4. fejezet

Fájlmutató

4.1. Fájllista

Az összes fájl listája rövid leírásokkal:

CsaladFilm.cpp	31
CsaladFilm.hpp	
Ez az osztály felelős a családi filmek megvalósításáért	31
DinTomb.hpp	
Ebbe a fájlba van a dinamikus tömb osztály	32
DokumentumFilm.cpp	32
DokumentumFilm.hpp	
Ez az osztály felelős a dokumentum filmek megvalósításáért	32
Filehiba.hpp	33
Film.cpp	33
Film.hpp	
Ez az osztály felelős az alap film osztály	34
Filmtar.cpp	34
Filmtar.hpp	
Ez az osztály felelős a filmek tárolásáért, a keresésért, olvasásért	34
gtest_lite.h	35
main.cpp	45
memtrace.cpp	45
memtrace.h	45
Tarolo_impl.hpp	45

5. fejezet

Névterek dokumentációja

5.1. gtest_lite névtér-referencia

[gtest_lite](#): a keretrendszer függvényinek és objektumainak névtére

Osztályok

- struct [Test](#)
- class [ostreamRedir](#)

Függvények

- `template<typename T1 , typename T2 >`
`std::ostream & EXPECT_ (T1 exp, T2 act, bool(*pred)(T1, T1), const char *file, int line, const char *expr, const char *lhs="elvart", const char *rhs="aktual")`
általános sablon a várt értékhez.
- `template<typename T1 , typename T2 >`
`std::ostream & EXPECT_ (T1 *exp, T2 *act, bool(*pred)(T1 *, T1 *), const char *file, int line, const char *expr, const char *lhs="elvart", const char *rhs="aktual")`
pointerre specializált sablon a várt értékhez.
- `std::ostream & EXPECTSTR (const char *exp, const char *act, bool(*pred)(const char *, const char *), const char *file, int line, const char *expr, const char *lhs="elvart", const char *rhs="aktual")`
- `template<typename T >`
`bool eq (T a, T b)`
- `bool eqstr (const char *a, const char *b)`
- `bool eqstrcase (const char *a, const char *b)`
- `template<typename T >`
`bool ne (T a, T b)`
- `bool nestr (const char *a, const char *b)`
- `template<typename T >`
`bool le (T a, T b)`
- `template<typename T >`
`bool lt (T a, T b)`
- `template<typename T >`
`bool ge (T a, T b)`
- `template<typename T >`
`bool gt (T a, T b)`
- `template<typename T >`
`bool almostEQ (T a, T b)`

5.1.1. Részletes leírás

`gtest_lite`: a keretrendszer függvényinek és objektumainak névtére

5.1.2. Függvények dokumentációja

5.1.2.1. `almostEQ()`

```
template<typename T >
bool gtest_lite::almostEQ (
    T a,
    T b )
```

Segédsablon valós számok összehasonlításához Nem bombabiztos, de nekünk most jó lesz Elméleti hátér:
<http://www.cygnus-software.com/papers/comparingfloats/comparingfloats.htm>

5.1.2.2. `eq()`

```
template<typename T >
bool gtest_lite::eq (
    T a,
    T b )
```

segéd sablonok a relációkhoz. azért nem STL (algorithm), mert csak a függvény lehet, hogy menjen a deduckció

5.1.2.3. `eqstr()`

```
bool gtest_lite::eqstr (
    const char * a,
    const char * b ) [inline]
```

5.1.2.4. `eqstrcase()`

```
bool gtest_lite::eqstrcase (
    const char * a,
    const char * b ) [inline]
```

5.1.2.5. EXPECT_() [1/2]

```
template<typename T1 , typename T2 >
std::ostream& gtest_lite::EXPECT_ (
    T1 * exp,
    T2 * act,
    bool(*) (T1 *, T1 *) pred,
    const char * file,
    int line,
    const char * expr,
    const char * lhs = "elvart",
    const char * rhs = "aktual" )
```

pointerre specializált sablon a várt értékhez.

5.1.2.6. EXPECT_() [2/2]

```
template<typename T1 , typename T2 >
std::ostream& gtest_lite::EXPECT_ (
    T1 exp,
    T2 act,
    bool(*) (T1, T1) pred,
    const char * file,
    int line,
    const char * expr,
    const char * lhs = "elvart",
    const char * rhs = "aktual" )
```

általános sablon a várt értékhez.

5.1.2.7. EXPECTSTR()

```
std::ostream& gtest_lite::EXPECTSTR (
    const char * exp,
    const char * act,
    bool(*) (const char *, const char *) pred,
    const char * file,
    int line,
    const char * expr,
    const char * lhs = "elvart",
    const char * rhs = "aktual" ) [inline]
```

stringek összehasonlításához. azért nem spec. mert a sima EQ-ra másként kell működnie.

5.1.2.8. ge()

```
template<typename T >
bool gtest_lite::ge (
    T a,
    T b )
```

5.1.2.9. gt()

```
template<typename T >
bool gtest_lite::gt (
    T a,
    T b )
```

5.1.2.10. le()

```
template<typename T >
bool gtest_lite::le (
    T a,
    T b )
```

5.1.2.11. lt()

```
template<typename T >
bool gtest_lite::lt (
    T a,
    T b )
```

5.1.2.12. ne()

```
template<typename T >
bool gtest_lite::ne (
    T a,
    T b )
```

5.1.2.13. nestr()

```
bool gtest_lite::nestr (
    const char * a,
    const char * b ) [inline]
```

6. fejezet

Osztályok dokumentációja

6.1. `_Is_Types< F, T >` struktúrasablon-referencia

Segédsablon típuskonverzió futás közbeni ellenőrzésére.

```
#include <gtest_lite.h>
```

Statikus publikus tagfüggvények

- `template<typename D >`
`static char(& f (D))[1]`
- `template<typename D >`
`static char(& f (...))[2]`

Statikus publikus attribútumok

- `static bool const convertible = sizeof(f<T>(F())) == 1`

6.1.1. Részletes leírás

```
template<typename F, typename T>  
struct _Is_Types< F, T >
```

Segédsablon típuskonverzió futás közbeni ellenőrzésére.

6.1.2. Tagfüggvények dokumentációja

6.1.2.1. f() [1/2]

```
template<typename F , typename T >
template<typename D >
static char(& _Is_Types< F, T >::f (
    ... )) [2] [static]
```

6.1.2.2. f() [2/2]

```
template<typename F , typename T >
template<typename D >
static char(& _Is_Types< F, T >::f (
    D )) [1] [static]
```

6.1.3. Adattagok dokumentációja**6.1.3.1. convertible**

```
template<typename F , typename T >
bool const _Is_Types< F, T >::convertible = sizeof(f<T>(F())) == 1 [static]
```

Ez a dokumentáció a struktúráról a következő fájl alapján készült:

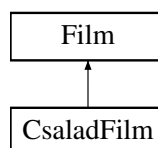
- [gtest_lite.h](#)

6.2. CsaladFilm osztályreferencia

Speciális korhatáros film.

```
#include <CsaladFilm.hpp>
```

A CsaladFilm osztály származási diagramja:

**Publikus tagfüggvények**

- [CsaladFilm](#) (const std::string &cim, int lejatszasisido, int kiadasiev, int korhatar)
Alap konstruktor ami a filmhez hasonló csak a korhatárt is meg kell adni.
- void [kiir](#) () const override
Kiírja a családfilm nevét, lejátszási idejét (meddig tart percben), korhatárát és címét.
- void [kiir_stream](#) (std::ostream &os) const override

További örökölt tagok

6.2.1. Részletes leírás

Speciális korhatáros film.

6.2.2. Konstruktorkorok és destruktorkorok dokumentációja

6.2.2.1. CsaladFilm()

```
CsaladFilm::CsaladFilm (
    const std::string & cim,
    int lejatszasiIdo,
    int kiadasiEv,
    int korhatar )
```

Alap konstruktor ami a filmhez hasonló csak a korhatárt is meg kell adni.

6.2.3. Tagfüggvények dokumentációja

6.2.3.1. kiir()

```
void CsaladFilm::kiir ( ) const [override], [virtual]
```

Kiírja a családfilm nevét, lejátszási idejét (meddig tart percben), korhatárát és címét.

Újraimplementált ősök: [Film](#).

6.2.3.2. kiir_stream()

```
void CsaladFilm::kiir_stream (
    std::ostream & os ) const [override], [virtual]
```

Újraimplementált ősök: [Film](#).

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

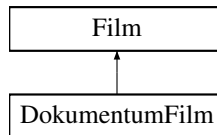
- [CsaladFilm.hpp](#)
- [CsaladFilm.cpp](#)

6.3. DokumentumFilm osztályreferencia

Speciális film leírással.

```
#include <DokumentumFilm.hpp>
```

A DokumentumFilm osztály származási diagramja:



Publikus tagfüggvények

- **DokumentumFilm** (const std::string &cim, int lejatszasiIdo, int kiadasiEv, const std::string &leiras)
Alap konstruktor ami a filmhez hasonló csak a leírást is meg kell adni.
- void **kiir** () const override
Kiírja a családfilm nevét, lejátszási idejét (meddíg tart percben), leírását és címét.
- void **kiir_stream** (std::ostream &os) const override

További örökölt tagok

6.3.1. Részletes leírás

Speciális film leírással.

6.3.2. Konstruktorok és destruktorok dokumentációja

6.3.2.1. DokumentumFilm()

```
DokumentumFilm::DokumentumFilm (
    const std::string & cim,
    int lejatszasiIdo,
    int kiadasiEv,
    const std::string & leiras )
```

Alap konstruktor ami a filmhez hasonló csak a leírást is meg kell adni.

6.3.3. Tagfüggvények dokumentációja

6.3.3.1. kiir()

```
void DokumentumFilm::kiir ( ) const [override], [virtual]
```

Kiírja a családfilm nevét, lejátszási idejét (meddíg tart percben), leírását és címét.

Újrimplementált ősök: [Film](#).

6.3.3.2. kiir_stream()

```
void DokumentumFilm::kiir_stream (
    std::ostream & os ) const [override], [virtual]
```

Újrimplementált ősök: [Film](#).

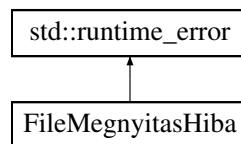
Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [DokumentumFilm.hpp](#)
- [DokumentumFilm.cpp](#)

6.4. FileMegnyitasHiba osztályreferencia

```
#include <Filehiba.hpp>
```

A FileMegnyitasHiba osztály származási diagramja:



Publikus tagfüggvények

- [FileMegnyitasHiba](#) (const std::string &uzenet)

6.4.1. Konstruktorok és destruktorok dokumentációja

6.4.1.1. FileMegnyitasHiba()

```
FileMegnyitasHiba::FileMegnyitasHiba (
    const std::string & uzenet ) [inline], [explicit]
```

Ez a dokumentáció az osztályról a következő fájl alapján készült:

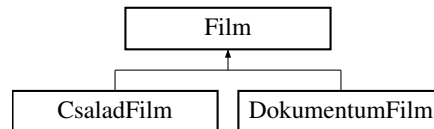
- [Filehiba.hpp](#)

6.5. Film osztályreferencia

Az alap film base osztály. Tartalmazza a film címét, meddig tart és kiadási évét. (Megjelenése)

```
#include <Film.hpp>
```

A Film osztály származási diagramja:



Publikus tagfüggvények

- `Film` (`const std::string &cim`, `int lejatszasildo`, `int kiadasiEv`)
A film konstruktora ami címet, megjelenést és időtartamot kér.
- `virtual void kiir () const`
Kiírja a családfilm nevét, lejátszási idejét (meddig tart percben) és címét.
- `void setEv (int ujEv)`
Setter a megjelenési évre.
- `int getEv () const`
Visszaadja a megjelenési évet.
- `void setCim (const std::string &ujCim)`
Setter a címre.
- `std::string getCim () const`
Visszaadja a címet.
- `virtual void kiir_stream (std::ostream &os) const`
- `virtual ~Film ()=default`
Virtuális destruktorként fogadjuk az öröklődni osztályok a `Film` osztályból.

Védett attribútumok

- `std::string cim`
A film címe.
- `int lejatszasildo`
Meddig tart a film percben.
- `int kiadasiEv`
A film megjelenési ideje évben.

Barátok

- `std::ostream & operator<< (std::ostream &os, const Film &film)`

6.5.1. Részletes leírás

Az alap film base osztály. Tartalmazza a film címét, meddig tart és kiadási évét. (Megjelenése)

6.5.2. Konstruktorkok és destruktorkok dokumentációja

6.5.2.1. Film()

```
Film::Film (
    const std::string & cim,
    int   lejatszasiIdo,
    int   kiadasiEv )
```

A film konstruktora ami címet, megjelenést és időtartamot kér.

6.5.2.2. ~Film()

```
virtual Film::~Film ( ) [virtual], [default]
```

Virtuális destruktork mert fognak öröklődni osztályok a [Film](#) osztályból.

6.5.3. Tagfüggvények dokumentációja

6.5.3.1. getCim()

```
std::string Film::getCim ( ) const
```

Visszaadja a címet.

6.5.3.2. getEv()

```
int Film::getEv ( ) const
```

Visszaadja a megjelenési évet.

6.5.3.3. kiir()

```
void Film::kiir ( ) const [virtual]
```

Kiírja a családfilm nevét, lejátszási idejét (meddig tart percben) és címét.

Újraimplementáló leszármazottak: [DokumentumFilm](#) és [CsaladFilm](#).

6.5.3.4. kiir_stream()

```
void Film::kiir_stream (
    std::ostream & os ) const [virtual]
```

Újraimplementáló leszármazottak: [DokumentumFilm](#) és [CsaladFilm](#).

6.5.3.5. setCim()

```
void Film::setCim (
    const std::string & ujCim )
```

Setter a címre.

6.5.3.6. setEv()

```
void Film::setEv (
    int ujEv )
```

Setter a megjelenési évre.

6.5.4. Barát és kapcsolódó függvények dokumentációja

6.5.4.1. operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const Film & film ) [friend]
```

6.5.5. Adattagok dokumentációja

6.5.5.1. cim

```
std::string Film::cim [protected]
```

A film címe.

6.5.5.2. kiadasiEv

```
int Film::kiadasiEv [protected]
```

A film megjelenési ideje évben.

6.5.5.3. lejatszasiIdo

```
int Film::lejatszasiIdo [protected]
```

Meddig tart a film percben.

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [Film.hpp](#)
- [Film.cpp](#)

6.6. FilmTar osztályreferencia

A filmek tárolásáért és elérését biztosító osztály.

```
#include <Filmtar.hpp>
```

Publikus tagfüggvények

- void [hozzaad](#) ([Film](#) *film)
Hozzáad egy filmet a tárolóba.
- void [kiir](#) () const
Kiírja a filmtárolóba lévő összes filmet.
- [Tarolo](#)< [Film](#) * > [keres](#) (const std::string &keresett) const
Megkeresi a paraméterként kapott rész stringből a cím alapján a filmeket és egy dinamikus tömböt ad vissza amely azokat tartalmazza.
- void [modositas](#) (size_t index, const std::string &uj_cim, int uj_ev)
Modosítja az index-edik filmet új címmel és új évszámmal.
- void [torles](#) (int index)
Törli az index-edik elemet a tárolóból.
- size_t [mennyi_elem](#) () const
Megadja hogy hány film van a tárolóba.
- [~FilmTar](#) ()
Destruktor, felszabadítja a tárolt elemeket.
- void [file_ment](#) (const std::string &fajlnev) const
- void [file_beolvas](#) (const std::string &fajlnev)

6.6.1. Részletes leírás

A filmek tárolásáért és elérését biztosító osztály.

6.6.2. Konstruktorkok és destruktorkok dokumentációja

6.6.2.1. ~FilmTar()

```
FilmTar::~~FilmTar ( )
```

Destruktor, felszabadítja a tárolt elemeket.

6.6.3. Tagfüggvények dokumentációja

6.6.3.1. file_beolvas()

```
void FilmTar::file_beolvas (
    const std::string & fajlnev )
```

6.6.3.2. file_ment()

```
void FilmTar::file_ment (
    const std::string & fajlnev ) const
```

6.6.3.3. hozzaad()

```
void FilmTar::hozzaad (
    Film * film )
```

Hozzáad egy filmet a tárolóba.

6.6.3.4. keres()

```
Tarolo< Film * > FilmTar::keres (
    const std::string & keresett ) const
```

Megkeresi a paraméterként kapott rész stringből a cím alapján a filmeket és egy dinamikus tömböt ad vissza amely azokat tartalmazza.

6.6.3.5. kiir()

```
void FilmTar::kiir ( ) const
```

Kiírja a filmtárolóba lévő összes filmet.

6.6.3.6. mennyi_elem()

```
size_t FilmTar::mennyi_elem ( ) const
```

Megadja hogy hány film van a tárolóba.

6.6.3.7. modositas()

```
void FilmTar::modositas (
    size_t index,
    const std::string & uj_cim,
    int uj_ev )
```

Modosítja az index-edik filmet új címmel és új évszámmal.

6.6.3.8. torles()

```
void FilmTar::torles (
    int index )
```

Törli az index-edik elemet a tárolóból.

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [Filmtar.hpp](#)
- [Filmtar.cpp](#)

6.7. gtest_lite::ostreamRedir osztályreferencia

```
#include <gtest_lite.h>
```

Publikus tagfüggvények

- [ostreamRedir](#) (std::ostream &src, std::ostream &dst)
- [~ostreamRedir](#) ()

6.7.1. Részletes leírás

Segédsablon ostream átirányításához A destruktor visszaállít

6.7.2. Konstruktorok és destruktorok dokumentációja

6.7.2.1. ostreamRedir()

```
gtest_lite::ostreamRedir::ostreamRedir (
    std::ostream & src,
    std::ostream & dst ) [inline]
```

6.7.2.2. ~ostreamRedir()

```
gtest_lite::ostreamRedir::~~ostreamRedir ( ) [inline]
```

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- [gtest_lite.h](#)

6.8. Tarolo< T > osztálysablon-referencia

Egy alap generikus dinamikus tároló.

```
#include <DinTomb.hpp>
```

Publikus tagfüggvények

- [Tarolo](#) (bool birtokolt__{__}=true)
- [~Tarolo](#) ()
- void [hozzaad](#) (const T &elem)
Hozzáad egy új elemet a tömbhöz, ha nincs elég hely, foglal még és nagyobbodik a tömb.
- size_t [meret](#) () const
Visszaadja, hogy mennyi elem van a tömbben.
- void [torol](#) (size_t index)
Töröl egy indexedik elemet a tömbből.
- void [torol_egesz](#) ()
Törli az összes elemet a tömbből.
- T & [operator\[\]](#) (size_t index)
Visszaadja a tömb [i]-edik elemét.
- const T & [operator\[\]](#) (size_t index) const
Visszaadja a tömb [i]-edik elemét. Ez konstans hívásra kell.

6.8.1. Részletes leírás

```
template<typename T>
class Tarolo< T >
```

Egy alap generikus dinamikus tároló.

Sablon paraméterek

<i>T</i>	mit tároljon a tömb?
----------	----------------------

6.8.2. Konstruktorok és destruktorok dokumentációja

6.8.2.1. Tarolo()

```
template<typename T >
Tarolo< T >::Tarolo (
    bool birtokolt_ = true )
```

6.8.2.2. ~Tarolo()

```
template<typename T >
Tarolo< T >::~~Tarolo
```

6.8.3. Tagfüggvények dokumentációja

6.8.3.1. hozzaad()

```
template<typename T >
void Tarolo< T >::hozzaad (
    const T & elem )
```

Hozzáad egy új elemet a tömbhöz, ha nincs elég hely, foglal még és nagyobbodik a tömb.

6.8.3.2. meret()

```
template<typename T >
size_t Tarolo< T >::meret
```

Visszaadja, hogy mennyi elem van a tömbben.

6.8.3.3. operator[]() [1/2]

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

Visszaadja a tömb [i]-edik elemét.

6.8.3.4. operator[]() [2/2]

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

Visszaadja a tömb [i]-edik elemét. Ez konstans hívásra kell.

6.8.3.5. torol()

```
template<typename T >
void Tarolo< T >::torol (
    size_t index )
```

Töröl egy indexedik elemet a tömbből.

6.8.3.6. torol_egesz()

```
template<typename T >
void Tarolo< T >::torol_egesz
```

Törli az összes elemet a tömbből.

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [DinTomb.hpp](#)
- [Tarolo_impl.hpp](#)

6.9. gtest_lite::Test struktúrareferencia

```
#include <gtest_lite.h>
```

Publikus tagfüggvények

- void `begin` (const char *n)
Teszt kezdete.
- std::ostream & `end` (bool memchk=false)
Teszt vége.
- bool `fail` ()
- bool `astatus` ()
- std::ostream & `expect` (bool st, const char *file, int line, const char *expr, bool pr=false)
Eredményt adminisztráló tagfüggvény True a jó eset.
- `~Test` ()
Destruktor.

Statikus publikus tagfüggvények

- static `Test` & `getTest` ()

Publikus attribútumok

- int `sum`
tesztek számlálója
- int `failed`
hibás tesztek
- int `ablocks`
allokált blokkok száma
- bool `status`
éppen futó teszt státusza.
- bool `tmp`
temp a kivételkezeléshez;
- std::string `name`
éppen futó teszt neve.
- std::fstream `null`
nyelő, ha nem kell kiírni semmit
- std::ostream & `os`
ide írunk

6.9.1. Részletes leírás

Tesztek állapotát tároló osztály. Egyetlen egy statikus példány keletkezik, aminek a destruktora a futás végén hívódik meg.

6.9.2. Konstruktorok és destruktorok dokumentációja

6.9.2.1. ~Test()

```
gtest_lite::Test::~~Test ( ) [inline]
```

Destruktor.

6.9.3. Tagfüggvények dokumentációja

6.9.3.1. astatus()

```
bool gtest_lite::Test::astatus ( ) [inline]
```

6.9.3.2. begin()

```
void gtest_lite::Test::begin (
    const char * n ) [inline]
```

Teszt kezdete.

6.9.3.3. end()

```
std::ostream& gtest_lite::Test::end (
    bool memchk = false ) [inline]
```

Teszt vége.

6.9.3.4. expect()

```
std::ostream& gtest_lite::Test::expect (
    bool st,
    const char * file,
    int line,
    const char * expr,
    bool pr = false ) [inline]
```

Eredményt adminisztráló tagfüggvény True a jó eset.

6.9.3.5. fail()

```
bool gtest_lite::Test::fail ( ) [inline]
```

6.9.3.6. getTest()

```
static Test& gtest_lite::Test::getTest ( ) [inline], [static]
```

< egyedüli (singleton) példány

6.9.4. Adattagok dokumentációja

6.9.4.1. ablocks

```
int gtest_lite::Test::ablocks
```

allokált blokkok száma

6.9.4.2. failed

```
int gtest_lite::Test::failed
```

hibás tesztek

6.9.4.3. name

```
std::string gtest_lite::Test::name
```

éppen futó teszt neve.

6.9.4.4. null

```
std::fstream gtest_lite::Test::null
```

nyelő, ha nem kell kiírni semmit

6.9.4.5. os

```
std::ostream& gtest_lite::Test::os
```

ide írunk

6.9.4.6. status

```
bool gtest_lite::Test::status
```

éppen futó teszt státusza.

6.9.4.7. sum

```
int gtest_lite::Test::sum
```

tesztek számlálója

6.9.4.8. tmp

```
bool gtest_lite::Test::tmp
```

temp a kivételkezeléshez;

Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- [gtest_lite.h](#)

7. fejezet

Fájlok dokumentációja

7.1. CsaladFilm.cpp fájlreferencia

```
#include "CsaladFilm.hpp"
```

7.2. CsaladFilm.hpp fájlreferencia

Ez az osztály felelős a családi filmek megvalósításáért.

```
#include "Film.hpp"
```

Osztályok

- class [CsaladFilm](#)
Speciális korhatáros film.

7.2.1. Részletes leírás

Ez az osztály felelős a családi filmek megvalósításáért.

Szerző

Funk Gábor

Dátum

2025-04-27

7.3. DinTomb.hpp fájlreferencia

Ebbe a fájlba van a dinamikus tömb osztály.

```
#include <iostream>
#include <stdexcept>
#include <algorithm>
#include <cctype>
#include <string>
#include "Tarolo_impl.hpp"
```

Osztályok

- class `Tarolo< T >`
Egy alap generikus dinamikus tároló.

7.3.1. Részletes leírás

Ebbe a fájlba van a dinamikus tömb osztály.

Szerző

Funk Gábor

Dátum

2025-04-27

7.4. DokumentumFilm.cpp fájlreferencia

```
#include "DokumentumFilm.hpp"
```

7.5. DokumentumFilm.hpp fájlreferencia

Ez az osztály felelős a dokumentum filmek megvalósításáért.

```
#include "Film.hpp"
```

Osztályok

- class `DokumentumFilm`
Speciális film leírással.

7.5.1. Részletes leírás

Ez az osztály felelős a dokumentum filmek megvalósításáért.

Szerző

Funk Gábor

Dátum

2025-04-27

7.6. Filehiba.hpp fájlreferencia

```
#include <stdexcept>
#include <string>
```

Osztályok

- class [FileMegnyitasHiba](#)

7.7. Film.cpp fájlreferencia

```
#include "Film.hpp"
```

Függvények

- `std::ostream & operator<< (std::ostream &os, const Film &film)`

7.7.1. Függvények dokumentációja

7.7.1.1. operator<<()

```
std::ostream& operator<< (
    std::ostream & os,
    const Film & film )
```

7.8. Film.hpp fájlreferencia

Ez az osztály felelős az alap film osztály.

```
#include <string>
#include <iostream>
```

Osztályok

- class [Film](#)

Az alap film base osztály. Tartalmazza a film címét, meddig tart és kiadási évét. (Megjelenése)

7.8.1. Részletes leírás

Ez az osztály felelős az alap film osztály.

Szerző

Funk Gábor

Dátum

2025-04-27

7.9. Filmtar.cpp fájlreferencia

```
#include "Filmtar.hpp"
```

7.10. Filmtar.hpp fájlreferencia

Ez az osztály felelős a filmek tárolásáért, a keresésért, olvasásért.

```
#include "Film.hpp"
#include "CsaladFilm.hpp"
#include "DokumentumFilm.hpp"
#include <iostream>
#include <string>
#include <algorithm>
#include <cctype>
#include <fstream>
#include <sstream>
#include "Filehiba.hpp"
#include "DinTomb.hpp"
```

Osztályok

- class [FilmTar](#)

A filmek tárolásáért és elérését biztosító osztály.

7.10.1. Részletes leírás

Ez az osztály felelős a filmek tárolásáért, a keresésért, olvasásért.

Szerző

Funk Gábor

Dátum

2025-04-27

7.11. gtest_lite.h fájlreferencia

```
#include <iostream>
#include <cassert>
#include <cmath>
#include <cstring>
#include <limits>
#include <cstdlib>
#include <string>
#include <fstream>
```

Osztályok

- struct [_Is_Types< F, T >](#)
Segédsablon típuskonverzió futás közbeni ellenőrzésére.
- struct [gtest_lite::Test](#)
- class [gtest_lite::ostreamRedir](#)

Névterek

- [gtest_lite](#)
[gtest_lite](#): a keretrendszer függvényinek és objektumainak névtére

Makródefiníciók

- `#define TEST(C, N) do { gtest_lite::test.begin(#C"."#N);`
- `#define END gtest_lite::test.end(); } while (false);`
- Tesztelés vége.*
- `#define ENDM gtest_lite::test.end(true); } while (false);`
- `#define ENDMsg(t) gtest_lite::test.end(true) << t << std::endl; } while (false);`
- `#define SUCCEED() gtest_lite::test.expect(true, __FILE__, __LINE__, "SUCCEED()", true)`
- Sikeres teszt makrója.*
- `#define FAIL() gtest_lite::test.expect(false, __FILE__, __LINE__, "FAIL()", true)`
- Sikertelen teszt fatális hiba makrója.*
- `#define ADD_FAILURE() gtest_lite::test.expect(false, __FILE__, __LINE__, "ADD_FAILURE()", true)`
- Sikertelen teszt makrója.*
- `#define EXPECT_EQ(expected, actual) gtest_lite::EXPECT_(expected, actual, gtest_lite::eq, __FILE__, __LINE__, "EXPECT_EQ(" #expected ", " #actual ")")`
- Azonosságot elváró makró*
- `#define EXPECT_NE(expected, actual) gtest_lite::EXPECT_(expected, actual, gtest_lite::ne, __FILE__, __LINE__, "EXPECT_NE(" #expected ", " #actual ")", "etalon")`
- Eltérést elváró makró*
- `#define EXPECT_LE(expected, actual) gtest_lite::EXPECT_(expected, actual, gtest_lite::le, __FILE__, __LINE__, "EXPECT_LE(" #expected ", " #actual ")", "etalon")`
- Kisebb, vagy egyenlő relációt elváró makró*
- `#define EXPECT_LT(expected, actual) gtest_lite::EXPECT_(expected, actual, gtest_lite::lt, __FILE__, __LINE__, "EXPECT_LT(" #expected ", " #actual ")", "etalon")`
- Kisebb, mint relációt elváró makró*
- `#define EXPECT_GE(expected, actual) gtest_lite::EXPECT_(expected, actual, gtest_lite::ge, __FILE__, __LINE__, "EXPECT_GE(" #expected ", " #actual ")", "etalon")`
- Nagyobb, vagy egyenlő relációt elváró makró*
- `#define EXPECT_GT(expected, actual) gtest_lite::EXPECT_(expected, actual, gtest_lite::gt, __FILE__, __LINE__, "EXPECT_GT(" #expected ", " #actual ")", "etalon")`
- Nagyobb, mint relációt elváró makró*
- `#define EXPECT_TRUE(actual) gtest_lite::EXPECT_(true, actual, gtest_lite::eq, __FILE__, __LINE__, "EXPECT_TRUE(" #actual ")")`
- Igaz értéket elváró makró*
- `#define EXPECT_FALSE(actual) gtest_lite::EXPECT_(false, actual, gtest_lite::eq, __FILE__, __LINE__, "EXPECT_FALSE(" #actual ")")`
- Hamis értéket elváró makró*
- `#define EXPECT_FLOAT_EQ(expected, actual) gtest_lite::EXPECT_(expected, actual, gtest_lite::almostEQ, __FILE__, __LINE__, "EXPECT_FLOAT_EQ(" #expected ", " #actual ")")`
- Valós számok azonosságát elváró makró*
- `#define EXPECT_DOUBLE_EQ(expected, actual) gtest_lite::EXPECT_(expected, actual, gtest_lite::almostEQ, __FILE__, __LINE__, "EXPECT_DOUBLE_EQ(" #expected ", " #actual ")")`
- Valós számok azonosságát elváró makró*
- `#define EXPECT_STREQ(expected, actual) gtest_lite::EXPECTSTR(expected, actual, gtest_lite::eqstr, __FILE__, __LINE__, "EXPECT_STREQ(" #expected ", " #actual ")")`
- C stringek (const char *) azonosságát tesztelő makró*
- `#define EXPECT_STRNE(expected, actual) gtest_lite::EXPECTSTR(expected, actual, gtest_lite::nestr, __FILE__, __LINE__, "EXPECT_STRNE(" #expected ", " #actual ")", "etalon")`
- C stringek (const char *) eltérést tesztelő makró*
- `#define EXPECT_STRCASEEQ(expected, actual) gtest_lite::EXPECTSTR(expected, actual, gtest_lite::eqstrcase, __FILE__, __LINE__, "EXPECT_STRCASEEQ(" #expected ", " #actual ")")`
- C stringek (const char *) azonosságát tesztelő makró (kisbetű/nagybetű azonos)*

- `#define EXPECT_STRCASENE(expected, actual) gtest_lite::EXPECTSTR(expected, actual, gtest_lite::nestrcase, __FILE__, __LINE__, "EXPECT_STRCASENE(" #expected ", " #actual ")", "etalon")`
*C stringek (const char *) eltérését tesztelő makró (kisbetű/nagybetű azonos)*
- `#define EXPECT_THROW(statement, exception_type)`
Kivételt várunk.
- `#define EXPECT_ANY_THROW(statement)`
Kivételt várunk.
- `#define EXPECT_NO_THROW(statement)`
Nem várunk kivételt.
- `#define ASSERT_NO_THROW(statement)`
Nem várunk kivételt.
- `#define EXPECT_THROW_THROW(statement, exception_type)`
Kivételt várunk és továbbdobjuk – ilyen nincs a gtest-ben.
- `#define EXPECT_ENVEQ(expected, actual) gtest_lite::EXPECTSTR(std::getenv(expected), actual, gtest_lite::eqstr, __FILE__, __LINE__, "EXPECT_ENVEQ(" #expected ", " #actual ")")`
Környezeti változóhoz hasonlít – ilyen nincs a gtest-ben.
- `#define EXPECT_ENVCASEEQ(expected, actual) gtest_lite::EXPECTSTR(std::getenv(expected), actual, gtest_lite::eqstrcase, __FILE__, __LINE__, "EXPECT_ENVCASEEQ(" #expected ", " #actual ")")`
Környezeti változóhoz hasonlít – ilyen nincs a gtest-ben (kisbetű/nagybetű azonos)
- `#define ASSERT_EQ(expected, actual) gtest_lite::ASSERT_(expected, actual, gtest_lite::eq, "ASSER_EQ")`
Azonosságot elváró makró
- `#define ASSERT_NO_THROW(statement)`
Nem várunk kivételt.
- `#define CREATE_Has_(X)`
- `#define CREATE_Has_fn_(X, S)`
- `#define EXPECTTHROW(statement, exp, act)`
EXPECTTHROW: kivételkezelés.
- `#define ASSERTTHROW(statement, exp, act)`
- `#define ASSERT_(expected, actual, fn, op)`
- `#define GTINIT(IS)`
- `#define GTEND(os)`

Függvények

- `void hasMember (...)`
- `template<typename T1, typename T2>
std::ostream & gtest_lite::EXPECT_(T1 exp, T2 act, bool(*pred)(T1, T1), const char *file, int line, const char *expr, const char *lhs="elvt", const char *rhs="aktual")`
általános sablon a várt értékhez.
- `template<typename T1, typename T2>
std::ostream & gtest_lite::EXPECT_(T1 *exp, T2 *act, bool(*pred)(T1 *, T1 *), const char *file, int line, const char *expr, const char *lhs="elvt", const char *rhs="aktual")`
pointerre specializált sablon a várt értékhez.
- `std::ostream & gtest_lite::EXPECTSTR(const char *exp, const char *act, bool(*pred)(const char *, const char *), const char *file, int line, const char *expr, const char *lhs="elvt", const char *rhs="aktual")`
- `template<typename T>
bool gtest_lite::eq(T a, T b)`
- `bool gtest_lite::eqstr(const char *a, const char *b)`
- `bool gtest_lite::eqstrcase(const char *a, const char *b)`
- `template<typename T>
bool gtest_lite::ne(T a, T b)`
- `bool gtest_lite::nestr(const char *a, const char *b)`

- `template<typename T>`
`bool gtest_lite::le (T a, T b)`
- `template<typename T>`
`bool gtest_lite::lt (T a, T b)`
- `template<typename T>`
`bool gtest_lite::ge (T a, T b)`
- `template<typename T>`
`bool gtest_lite::gt (T a, T b)`
- `template<typename T>`
`bool gtest_lite::almostEQ (T a, T b)`

7.11.1. Részletes leírás

(v4/2022)

Google gtest keretrendszerhez hasonló rendszer. Sz.l. 2015., 2016., 2017. (*Has_X*) Sz.l. 2018 (*template*), *ENDM*, *ENDMsg*, *nullptr_t* Sz.l. 2019 *singleton* Sz.l. 2021 *ASSERT..*, *STRCASE...* Sz.l. 2021 *EXPECT_REGEXP*, *CREATE_Has_fn*, *cmp w. NULL*, *EXPECT_param fix V.B.*, Sz.l. 2022 *almostEQ fix*, Sz.l. 2022. *EXPECT_THROW fix*

A tesztelés legalapvetőbb funkcióit támogató függvények és makrók. Nem szálbiztos megvalósítás.

Szabadon felhasználható, bővíthető.

Használati példa: Teszteljük az $f(x)=2*x$ függvényt: `int f(int x) { return 2*x; }`

```
int main() { TEST(TeszEsetNeve, TesztNeve) EXPECT_EQ(0, f(0)); EXPECT_EQ(4, f(2)) << "A függvény hibás
eredményt adott" << std::endl; ... END ... // Fatális hiba esetén a tesztet nem fut tovább. Ezek az AS-
SERT... makrók. // Nem lehet a kiírásukhoz további üzenetet fűzni. PL: TEST(TeszEsetNeve, TesztNeve)
ASSERT_NO_THROW(f(0)); // itt nem lehet << "duma" EXPECT_EQ(4, f(2)) << "A függvény hibás eredményt
adott" << std::endl; ... END ...
```

A működés részleteinek megértése szorgalmi feladat.

7.11.2. Makródefiníciók dokumentációja

7.11.2.1. ADD_FAILURE

```
#define ADD_FAILURE( ) gtest_lite::test.expect(false, __FILE__, __LINE__, "ADD_FAILURE()",
true)
```

Sikertelen teszt makrója.

7.11.2.2. ASSERT_

```
#define ASSERT_(
    expected,
    actual,
    fn,
    op )
```

Érték:

```
EXPECT_(expected, actual, fn, __FILE__, __LINE__, #op "(" #expected ", " #actual ")" ); \
if (!gtest_lite::test.status) { gtest_lite::test.end(); break; }
```

7.11.2.3. ASSERT_EQ

```
#define ASSERT_EQ(
    expected,
    actual ) gtest_lite::ASSERT_(expected, actual, gtest_lite::eq, "ASER_EQ")
```

Azonosságot elváró makró

ASSERT típusú ellenőrzések. CSak 1-2 van megvalósítva. Nem ostream& -val térnek vissza !!! Kivételt várunk

7.11.2.4. ASSERT_NO_THROW [1/2]

```
#define ASSERT_NO_THROW(
    statement )
```

Érték:

```
try { gtest_lite::test.tmp = true; statement; } \
catch (...) { gtest_lite::test.tmp = false; }\
ASSERTTHROW(statement, "nem dob kivetelt.", "kivetelt dobott.")
```

Nem várunk kivételt.

7.11.2.5. ASSERT_NO_THROW [2/2]

```
#define ASSERT_NO_THROW(
    statement )
```

Érték:

```
try { gtest_lite::test.tmp = true; statement; } \
catch (...) { gtest_lite::test.tmp = false; }\
ASSERTTHROW(statement, "nem dob kivetelt.", "kivetelt dobott.")
```

Nem várunk kivételt.

7.11.2.6. ASSERTTHROW

```
#define ASSERTTHROW(
    statement,
    exp,
    act )
```

Érték:

```
gtest_lite::test.expect(gtest_lite::test.tmp, __FILE__, __LINE__, #statement) \
« "*** Az utasítás " « (act) \
« "\n** Azt vartuk, hogy " « (exp) « std::endl; if (!gtest_lite::test.status) { gtest_lite::test.end(); \
break; }
```

7.11.2.7. CREATE_Has_

```
#define CREATE_Has_(
    X )
```

Érték:

```
template<typename T> struct _Has_##X { \
    struct Fallback { int X; }; \
    struct Derived : T, Fallback {}; \
    template<typename C, C> struct ChT; \
    template<typename D> static char (&f(ChT<int Fallback::*, &D::X>*)) [1]; \
    template<typename D> static char (&f(...)) [2]; \
    static bool const member = sizeof(f<Derived>(0)) == 2; \
};
```

Segédmakró egy adattag, vagy tagfüggvény létezésének tesztelésére futási időben Ötlet: <https://cpptalk.wordpress.com/2009/09/12/substitution-failure-is-not-an-error-2>

Használat: `CREATE_Has_(size) ... if (_Has_size<std::string>::member)...`

7.11.2.8. CREATE_Has_fn_

```
#define CREATE_Has_fn_(
    X,
    S )
```

Érték:

```
template<typename R, typename T> struct _Has_fn_##X##_##S { \
    template<typename C, R (C::*f)() S> struct ChT; \
    template<typename D> static char (&f(ChT<D, &D::X>*)) [1]; \
    template<typename D> static char (&f(...)) [2]; \
    static bool const fn = sizeof(f<T>(0)) == 1; \
};
```

7.11.2.9. END

```
#define END gtest_lite::test.end(); } while (false);
```

Tesztelés vége.

7.11.2.10. ENDM

```
#define ENDM gtest_lite::test.end(true); } while (false);
```

Tesztelés vége allokált blokkok számának összehasonlításával Ez az ellenőrzés nem bomba biztos.

7.11.2.11. ENDMsg

```
#define ENDMsg(
    t ) gtest_lite::test.end(true) << t << std::endl; } while (false);
```

Tesztelés vége allokált blokkok számának összehasonlításával Ez az ellenőrzés nem bomba biztos. Ha hiba van kiírja az üzenetet.

7.11.2.12. EXPECT_ANY_THROW

```
#define EXPECT_ANY_THROW(
    statement )
```

Érték:

```
try { gtest_lite::test.tmp = false; statement; } \
catch (...) { gtest_lite::test.tmp = true; } \
EXPECT_THROW(statement, "kivetelt dob.", "nem dobott kivetelt.")
```

Kivételt várunk.

7.11.2.13. EXPECT_DOUBLE_EQ

```
#define EXPECT_DOUBLE_EQ(
    expected,
```



```

        actual ) gtest_lite::EXPECT_(expected, actual, gtest_lite::almostEQ, __FILE__, ↵
__LINE__, "EXPECT_DOUBLE_EQ(" #expected ", " #actual ")") )

```

Valós számok azonosságát elváró makró

7.11.2.14. EXPECT_ENVCASEEQ

```

#define EXPECT_ENVCASEEQ(
    expected,
    actual ) gtest_lite::EXPECTSTR(std::getenv(expected), actual, gtest_lite::eqstrcase,
__FILE__, __LINE__, "EXPECT_ENVCASEEQ(" #expected ", " #actual ")") )

```

Környezeti változóhoz hasonlít – ilyen nincs a gtest-ben (kisbetű/nagybetű azonos)

7.11.2.15. EXPECT_ENVEQ

```

#define EXPECT_ENVEQ(
    expected,
    actual ) gtest_lite::EXPECTSTR(std::getenv(expected), actual, gtest_lite::eqstr,
__FILE__, __LINE__, "EXPECT_ENVEQ(" #expected ", " #actual ")") )

```

Környezeti változóhoz hasonlít – ilyen nincs a gtest-ben.

7.11.2.16. EXPECT_EQ

```

#define EXPECT_EQ(
    expected,
    actual ) gtest_lite::EXPECT_(expected, actual, gtest_lite::eq, __FILE__, __LINE↵
__, "EXPECT_EQ(" #expected ", " #actual ")") )

```

Azonosságot elváró makró

7.11.2.17. EXPECT_FALSE

```

#define EXPECT_FALSE(
    actual ) gtest_lite::EXPECT_(false, actual, gtest_lite::eq, __FILE__, __LINE__,
"EXPECT_FALSE(" #actual ")") )

```

Hamis értéket elváró makró

7.11.2.18. EXPECT_FLOAT_EQ

```

#define EXPECT_FLOAT_EQ(
    expected,
    actual ) gtest_lite::EXPECT_(expected, actual, gtest_lite::almostEQ, __FILE__, ↵
__LINE__, "EXPECT_FLOAT_EQ(" #expected ", " #actual ")") )

```

Valós számok azonosságát elváró makró

7.11.2.19. EXPECT_GE

```

#define EXPECT_GE(
    expected,
    actual ) gtest_lite::EXPECT_(expected, actual, gtest_lite::ge, __FILE__, __LINE↵
__, "EXPECT_GE(" #expected ", " #actual ")", "etalon") )

```

Nagyobb, vagy egyenlő relációt elváró makró

7.11.2.20. EXPECT_GT

```
#define EXPECT_GT(  
    expected,  
    actual ) gtest_lite::EXPECT_(expected, actual, gtest_lite::gt, __FILE__, __LINE__  
    __, "EXPECT_GT(" #expected ", " #actual ")", "etalon" )
```

Nagyobb, mint relációt elváró makró

7.11.2.21. EXPECT_LE

```
#define EXPECT_LE(  
    expected,  
    actual ) gtest_lite::EXPECT_(expected, actual, gtest_lite::le, __FILE__, __LINE__  
    __, "EXPECT_LE(" #expected ", " #actual ")", "etalon" )
```

Kisebb, vagy egyenlő relációt elváró makró

7.11.2.22. EXPECT_LT

```
#define EXPECT_LT(  
    expected,  
    actual ) gtest_lite::EXPECT_(expected, actual, gtest_lite::lt, __FILE__, __LINE__  
    __, "EXPECT_LT(" #expected ", " #actual ")", "etalon" )
```

Kisebb, mint relációt elváró makró

7.11.2.23. EXPECT_NE

```
#define EXPECT_NE(  
    expected,  
    actual ) gtest_lite::EXPECT_(expected, actual, gtest_lite::ne, __FILE__, __LINE__  
    __, "EXPECT_NE(" #expected ", " #actual ")", "etalon" )
```

Eltérést elváró makró

7.11.2.24. EXPECT_NO_THROW

```
#define EXPECT_NO_THROW(  
    statement )
```

Érték:

```
try { gtest_lite::test.tmp = true; statement; } \  
catch (...) { gtest_lite::test.tmp = false; }\  
EXPECT_THROW(statement, "nem dob kivetelt.", "kivetelt dobott.")
```

Nem várunk kivételt.

7.11.2.25. EXPECT_STRCASEEQ

```
#define EXPECT_STRCASEEQ(  
    expected,  
    actual ) gtest_lite::EXPECTSTR(expected, actual, gtest_lite::eqstrcase, __FILE__  
    __, __LINE__, "EXPECT_STRCASEEQ(" #expected ", " #actual ")" )
```

C stringek (const char *) azonosságát tesztelő makró (kisbetű/nagybetű azonos)

7.11.2.26. EXPECT_STRCASENE

```
#define EXPECT_STRCASENE(  
    expected,
```

```

        actual ) gtest_lite::EXPECTSTR(expected, actual, gtest_lite::nestr, __FILE__, ↵
_, __LINE__, "EXPECT_STRCASENE(" #expected ", " #actual ")", "etalon" )

```

C stringek (const char *) eltéréset tesztelő makró (kisbetű/nagybetű azonos)

7.11.2.27. EXPECT_STREQ

```

#define EXPECT_STREQ(
    expected,
    actual ) gtest_lite::EXPECTSTR(expected, actual, gtest_lite::eqstr, __FILE__, ↵
__LINE__, "EXPECT_STREQ(" #expected ", " #actual " )" )

```

C stringek (const char *) azonosságát tesztelő makró

7.11.2.28. EXPECT_STRNE

```

#define EXPECT_STRNE(
    expected,
    actual ) gtest_lite::EXPECTSTR(expected, actual, gtest_lite::nestr, __FILE__, ↵
__LINE__, "EXPECT_STRNE(" #expected ", " #actual ")", "etalon" )

```

C stringek (const char *) eltéréset tesztelő makró

7.11.2.29. EXPECT_THROW

```

#define EXPECT_THROW(
    statement,
    exception_type )

```

Érték:

```

try { gtest_lite::test.tmp = false; statement; } \
catch (exception_type &e) { gtest_lite::test.tmp = true; } \
catch (...) { } \
EXPECTTHROW(statement, "kivetelt dob.", "nem dobott '" #exception_type "' kivetelt.")

```

Kivetelt várunk.

7.11.2.30. EXPECT_THROW_THROW

```

#define EXPECT_THROW_THROW(
    statement,
    exception_type )

```

Érték:

```

try { gtest_lite::test.tmp = false; statement; } \
catch (exception_type &e) { gtest_lite::test.tmp = true; throw; } \
EXPECTTHROW(statement, "kivetelt dob.", "nem dobott '" #exception_type "' kivetelt.")

```

Kivetelt várunk és továbbdobjuk – ilyen nincs a gtest-ben.

7.11.2.31. EXPECT_TRUE

```

#define EXPECT_TRUE(
    actual ) gtest_lite::EXPECT_(true, actual, gtest_lite::eq, __FILE__, __LINE__ ↵
, "EXPECT_TRUE(" #actual " )" )

```

Igaz értéket elváró makró

7.11.2.32. EXPECTTHROW

```

#define EXPECTTHROW(
    statement,
    exp,
    act )

```

Érték:

```
gtest_lite::test.expect(gtest_lite::test.tmp, __FILE__, __LINE__, #statement) \
« "*** Az utasítás " « (act) \
« "\n** Azt vártuk, hogy " « (exp) « std::endl
```

EXPECTTHROW; kivételkezelés.

Belső megvalósításhoz tartozó makrók, és osztályok.

7.11.2.33. Nem célszerű közvetlenül használni, vagy módosítani**7.11.2.34. FAIL**

```
#define FAIL( ) gtest_lite::test.expect(false, __FILE__, __LINE__, "FAIL()", true)
```

Sikertelen teszt fatális hiba makrója.

7.11.2.35. GTEND

```
#define GTEND(
    os )
```

7.11.2.36. GTINIT

```
#define GTINIT(
    IS )
```

7.11.2.37. SUCCEED

```
#define SUCCEED( ) gtest_lite::test.expect(true, __FILE__, __LINE__, "SUCCEED()", true)
```

Sikeres teszt makrója.

7.11.2.38. TEST

```
#define TEST(
    C,
    N ) do { gtest_lite::test.begin(#C"."#N);
```

Teszt kezdete. A makró paraméterezése hasonlít a gtest paraméterezéséhez. Így az itt elkészített tesztek könnyen átemelhetők a gtest keretrendszerbe.

Paraméterek

<i>C</i>	- teszt eset neve (csak a gtest kompatibilitás miatt van külön neve az eseteknek)
<i>N</i>	- teszt neve

7.11.3. Függvények dokumentációja**7.11.3.1. hasMember()**

```
void hasMember (
    ... ) [inline]
```

Segédfüggvény egy publikus adattag, vagy tagfüggvény létezésének tesztelésére fordítási időben

7.12. main.cpp fájlreferencia

```
#include "Filmtar.hpp"  
#include "gtest_lite.h"  
#include "memtrace.h"
```

Függvények

- int `main` ()

7.12.1. Függvények dokumentációja

7.12.1.1. main()

```
int main ( )
```

7.13. memtrace.cpp fájlreferencia

```
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <time.h>  
#include <ctype.h>
```

7.14. memtrace.h fájlreferencia

7.15. Tarolo_impl.hpp fájlreferencia

Tárgymutató

[_Is_Types< F, T >, 13](#)
 [convertable, 14](#)
 [f, 13, 14](#)
[~Film](#)
 [Film, 19](#)
[~FilmTar](#)
 [FilmTar, 22](#)
[~Tarolo](#)
 [Tarolo< T >, 25](#)
[~Test](#)
 [gtest_lite::Test, 27](#)
[~ostreamRedir](#)
 [gtest_lite::ostreamRedir, 24](#)

[ablocks](#)
 [gtest_lite::Test, 29](#)
[ADD_FAILURE](#)
 [gtest_lite.h, 38](#)
[almostEQ](#)
 [gtest_lite, 10](#)
[ASSERT_](#)
 [gtest_lite.h, 38](#)
[ASSERT_EQ](#)
 [gtest_lite.h, 39](#)
[ASSERT_NO_THROW](#)
 [gtest_lite.h, 39](#)
[ASSERTTHROW](#)
 [gtest_lite.h, 39](#)
[astatus](#)
 [gtest_lite::Test, 28](#)

[begin](#)
 [gtest_lite::Test, 28](#)

[cim](#)
 [Film, 20](#)
[convertable](#)
 [_Is_Types< F, T >, 14](#)
[CREATE_Has_](#)
 [gtest_lite.h, 39](#)
[CREATE_Has_fn_](#)
 [gtest_lite.h, 40](#)
[CsaladFilm, 14](#)
 [CsaladFilm, 15](#)
 [kiir, 15](#)
 [kiir_stream, 15](#)
[CsaladFilm.cpp, 31](#)
[CsaladFilm.hpp, 31](#)

[DinTomb.hpp, 32](#)

[DokumentumFilm, 16](#)
 [DokumentumFilm, 16](#)
 [kiir, 16](#)
 [kiir_stream, 17](#)
[DokumentumFilm.cpp, 32](#)
[DokumentumFilm.hpp, 32](#)

[END](#)
 [gtest_lite.h, 40](#)
[end](#)
 [gtest_lite::Test, 28](#)
[ENDM](#)
 [gtest_lite.h, 40](#)
[ENDMsg](#)
 [gtest_lite.h, 40](#)
[eq](#)
 [gtest_lite, 10](#)
[eqstr](#)
 [gtest_lite, 10](#)
[eqstrcase](#)
 [gtest_lite, 10](#)
[expect](#)
 [gtest_lite::Test, 28](#)
[EXPECT_](#)
 [gtest_lite, 10, 11](#)
[EXPECT_ANY_THROW](#)
 [gtest_lite.h, 40](#)
[EXPECT_DOUBLE_EQ](#)
 [gtest_lite.h, 40](#)
[EXPECT_ENVCASEEQ](#)
 [gtest_lite.h, 41](#)
[EXPECT_ENVEQ](#)
 [gtest_lite.h, 41](#)
[EXPECT_EQ](#)
 [gtest_lite.h, 41](#)
[EXPECT_FALSE](#)
 [gtest_lite.h, 41](#)
[EXPECT_FLOAT_EQ](#)
 [gtest_lite.h, 41](#)
[EXPECT_GE](#)
 [gtest_lite.h, 41](#)
[EXPECT_GT](#)
 [gtest_lite.h, 41](#)
[EXPECT_LE](#)
 [gtest_lite.h, 42](#)
[EXPECT_LT](#)
 [gtest_lite.h, 42](#)
[EXPECT_NE](#)
 [gtest_lite.h, 42](#)

EXPECT_NO_THROW
 gtest_lite.h, 42
 EXPECT_STRCASEEQ
 gtest_lite.h, 42
 EXPECT_STRCASENE
 gtest_lite.h, 42
 EXPECT_STREQ
 gtest_lite.h, 43
 EXPECT_STRNE
 gtest_lite.h, 43
 EXPECT_THROW
 gtest_lite.h, 43
 EXPECT_THROW_THROW
 gtest_lite.h, 43
 EXPECT_TRUE
 gtest_lite.h, 43
 EXPECTSTR
 gtest_lite, 11
 EXPECTTHROW
 gtest_lite.h, 43

 f
 _Is_Types< F, T >, 13, 14
 FAIL
 gtest_lite.h, 44
 fail
 gtest_lite::Test, 28
 failed
 gtest_lite::Test, 29
 file_beolvas
 FilmTar, 22
 file_ment
 FilmTar, 22
 Filehiba.hpp, 33
 FileMegnyitasHiba, 17
 FileMegnyitasHiba, 17
 Film, 18
 ~Film, 19
 cim, 20
 Film, 19
 getCim, 19
 getEv, 19
 kiadasiEv, 20
 kiir, 19
 kiir_stream, 19
 lejatszasilido, 21
 operator<<, 20
 setCim, 20
 setEv, 20
 Film.cpp, 33
 operator<<, 33
 Film.hpp, 34
 FilmTar, 21
 ~FilmTar, 22
 file_beolvas, 22
 file_ment, 22
 hozzaad, 22
 keres, 22
 kiir, 22
 mennyi_elem, 23
 modositas, 23
 torles, 23
 Filmtar.cpp, 34
 Filmtar.hpp, 34

 ge
 gtest_lite, 11
 getCim
 Film, 19
 getEv
 Film, 19
 getTest
 gtest_lite::Test, 29
 gt
 gtest_lite, 11
 GTEND
 gtest_lite.h, 44
 gtest_lite, 9
 almostEQ, 10
 eq, 10
 eqstr, 10
 eqstrcase, 10
 EXPECT_, 10, 11
 EXPECTSTR, 11
 ge, 11
 gt, 11
 le, 12
 lt, 12
 ne, 12
 nestr, 12
 gtest_lite.h, 35
 ADD_FAILURE, 38
 ASSERT_, 38
 ASSERT_EQ, 39
 ASSERT_NO_THROW, 39
 ASSERTTHROW, 39
 CREATE_Has_, 39
 CREATE_Has_fn_, 40
 END, 40
 ENDM, 40
 ENDMsg, 40
 EXPECT_ANY_THROW, 40
 EXPECT_DOUBLE_EQ, 40
 EXPECT_ENVCASEEQ, 41
 EXPECT_ENVEQ, 41
 EXPECT_EQ, 41
 EXPECT_FALSE, 41
 EXPECT_FLOAT_EQ, 41
 EXPECT_GE, 41
 EXPECT_GT, 41
 EXPECT_LE, 42
 EXPECT_LT, 42
 EXPECT_NE, 42
 EXPECT_NO_THROW, 42
 EXPECT_STRCASEEQ, 42
 EXPECT_STRCASENE, 42
 EXPECT_STREQ, 43
 EXPECT_STRNE, 43

- EXPECT_THROW, 43
- EXPECT_THROW_THROW, 43
- EXPECT_TRUE, 43
- EXPECTTHROW, 43
- FAIL, 44
- GTEND, 44
- GTINIT, 44
- hasMember, 44
- SUCCEED, 44
- TEST, 44
- gtest_lite::ostreamRedir, 23
 - ~ostreamRedir, 24
 - ostreamRedir, 24
- gtest_lite::Test, 26
 - ~Test, 27
 - ablocks, 29
 - astatus, 28
 - begin, 28
 - end, 28
 - expect, 28
 - fail, 28
 - failed, 29
 - getTest, 29
 - name, 29
 - null, 29
 - os, 29
 - status, 30
 - sum, 30
 - tmp, 30
- GTINIT
 - gtest_lite.h, 44
- hasMember
 - gtest_lite.h, 44
- hozzaad
 - FilmTar, 22
 - Tarolo< T >, 25
- keres
 - FilmTar, 22
- kiadasiEv
 - Film, 20
- kiir
 - CsaladFilm, 15
 - DokumentumFilm, 16
 - Film, 19
 - FilmTar, 22
- kiir_stream
 - CsaladFilm, 15
 - DokumentumFilm, 17
 - Film, 19
- le
 - gtest_lite, 12
- lejatszasiIdo
 - Film, 21
- lt
 - gtest_lite, 12
- main
 - main.cpp, 45
- main.cpp, 45
 - main, 45
- memtrace.cpp, 45
- memtrace.h, 45
- mennyi_elem
 - FilmTar, 23
- meret
 - Tarolo< T >, 25
- modositas
 - FilmTar, 23
- name
 - gtest_lite::Test, 29
- ne
 - gtest_lite, 12
- nestr
 - gtest_lite, 12
- null
 - gtest_lite::Test, 29
- operator<<
 - Film, 20
 - Film.cpp, 33
- operator[]
 - Tarolo< T >, 25, 26
- os
 - gtest_lite::Test, 29
- ostreamRedir
 - gtest_lite::ostreamRedir, 24
- setCim
 - Film, 20
- setEv
 - Film, 20
- status
 - gtest_lite::Test, 30
- SUCCEED
 - gtest_lite.h, 44
- sum
 - gtest_lite::Test, 30
- Tarolo
 - Tarolo< T >, 25
- Tarolo< T >, 24
 - ~Tarolo, 25
 - hozzaad, 25
 - meret, 25
 - operator[], 25, 26
 - Tarolo, 25
 - torol, 26
 - torol_egesz, 26
- Tarolo_impl.hpp, 45
- TEST
 - gtest_lite.h, 44
- tmp
 - gtest_lite::Test, 30
- torles

FilmTar, [23](#)
torol
Tarolo< T >, [26](#)
torol_egesz
Tarolo< T >, [26](#)