

Filmtár

Generated by Doxygen 1.9.1

1 proghf	1
2 Namespace Index	3
2.1 Namespace List	3
3 Hierarchical Index	5
3.1 Class Hierarchy	5
4 Class Index	7
4.1 Class List	7
5 File Index	9
5.1 File List	9
6 Namespace Documentation	11
6.1 gtest_lite Namespace Reference	11
6.1.1 Detailed Description	12
6.1.2 Function Documentation	12
6.1.2.1 almostEQ()	12
6.1.2.2 eq()	12
6.1.2.3 EXPECTSTR()	12
7 Class Documentation	13
7.1 _Is_Types< F, T > Struct Template Reference	13
7.1.1 Detailed Description	13
7.2 _registry_item Struct Reference	14
7.3 call_t Struct Reference	14
7.4 CsaladiFilm Class Reference	14
7.4.1 Detailed Description	15
7.4.2 Constructor & Destructor Documentation	15
7.4.2.1 CsaladiFilm() [1/2]	15
7.4.2.2 CsaladiFilm() [2/2]	15
7.4.3 Member Function Documentation	15
7.4.3.1 getPlusData()	15
7.4.3.2 setPlusData()	15
7.5 DokumentumFilm Class Reference	16
7.5.1 Detailed Description	16
7.5.2 Constructor & Destructor Documentation	16
7.5.2.1 DokumentumFilm() [1/2]	16
7.5.2.2 DokumentumFilm() [2/2]	16
7.5.3 Member Function Documentation	17
7.5.3.1 getPlusData()	17
7.5.3.2 setPlusData()	17
7.6 Film Class Reference	17

7.6.1 Detailed Description	18
7.6.2 Constructor & Destructor Documentation	18
7.6.2.1 Film() [1/2]	18
7.6.2.2 Film() [2/2]	18
7.6.2.3 ~Film()	18
7.6.3 Member Function Documentation	19
7.6.3.1 filmModosit()	19
7.6.3.2 getHossz()	19
7.6.3.3 getJel()	19
7.6.3.4 getKiadas()	19
7.6.3.5 getKiirhossz()	19
7.6.3.6 getNev()	19
7.6.3.7 getPlusData()	20
7.6.3.8 kiir()	20
7.6.3.9 setHossz()	20
7.6.3.10 setJel()	20
7.6.3.11 setKiadas()	20
7.6.3.12 setNev()	20
7.6.3.13 setPlusData()	21
7.7 Filmek Class Reference	21
7.7.1 Detailed Description	21
7.7.2 Constructor & Destructor Documentation	21
7.7.2.1 Filmek()	22
7.7.2.2 ~Filmek()	22
7.7.3 Member Function Documentation	22
7.7.3.1 add() [1/2]	22
7.7.3.2 add() [2/2]	22
7.7.3.3 clear()	22
7.7.3.4 getFilm()	22
7.7.3.5 getFilmekPointer()	23
7.7.3.6 getFilmPointer()	23
7.7.3.7 getMeret()	23
7.7.3.8 hozzaadas()	23
7.7.3.9 keres()	23
7.7.3.10 kereslistaz()	23
7.7.3.11 keresveModosit()	23
7.7.3.12 keresveTorol()	24
7.7.3.13 kilistaz()	24
7.7.3.14 setFilm()	24
7.7.3.15 setFilmPointer()	24
7.7.3.16 torol() [1/2]	24
7.7.3.17 torol() [2/2]	24

7.8 gtest_lite::Test Struct Reference	25
7.8.1 Detailed Description	25
7.8.2 Member Function Documentation	25
7.8.2.1 getTest()	26
8 File Documentation	27
8.1 gtest_lite.h File Reference	27
8.1.1 Detailed Description	30
8.1.2 Macro Definition Documentation	30
8.1.2.1 ASSERT_	30
8.1.2.2 ASSERT_EQ	30
8.1.2.3 ASSERT_NO_THROW [1/2]	31
8.1.2.4 ASSERT_NO_THROW [2/2]	31
8.1.2.5 ASSERTTHROW	31
8.1.2.6 CREATE_Has_	31
8.1.2.7 ENDM	31
8.1.2.8 ENDMsg	31
8.1.2.9 EXPECT_ANY_THROW	32
8.1.2.10 EXPECT_NO_THROW	32
8.1.2.11 EXPECT_THROW	32
8.1.2.12 EXPECT_THROW_THROW	32
8.1.2.13 EXPECTTHROW	32
8.1.2.14 Nem célszerű közvetlenül használni, vagy módosítani	32
8.1.2.15 TEST	33
8.1.3 Function Documentation	33
8.1.3.1 hasMember()	33
8.2 main.cpp File Reference	33
8.2.1 Detailed Description	33
8.2.2 Function Documentation	33
8.2.2.1 test_0()	33
8.2.2.2 test_1()	34
8.2.2.3 test_2()	34
8.2.2.4 test_3()	34
8.2.2.5 test_4()	34
8.2.2.6 test_5()	34
8.3 segedfgvek.h File Reference	34
8.3.1 Detailed Description	34
8.3.2 Function Documentation	34
8.3.2.1 beolvas()	35
8.3.2.2 clrscreen()	35
8.3.2.3 customconst()	35
8.3.2.4 getjel()	35

8.3.2.5 getszam()	35
8.3.2.6 menukiir()	35
8.3.2.7 txtTorol()	35
8.3.2.8 visszair()	35

Index	37
--------------	-----------

Chapter 1

proghf

Fimtár

A feladatom:

A feladatomnak a "Filmtár" nevű programot választotta a feladatötlektől, egy nyílvántartó rendszert fogok létrehozni amiben filmeket tárolok. Minden film rendelkezik majd címmel, lejátszási idővel és kiadási évvel. Továbbá filmeket meg lehet majd különböztetni kategória alapján. (pl. családi, dokumentum) Ezek bónusz adatokkal is rendelkeznek.

Funkciók: -Listázás -Hozzáadás -Keresés -Törlés

A nyílvántartásra használt program működését egy konzolos feőleten lehet majd irányítani, számokat megadva az egyes funkciók eléréséhez, más adatot nem fogad el a program.

A programot egy menüben lehet majd irányítani ahova minden művelet után visszadob a rendszer, onnan lehet majd kiválasztani a következő lépést.

A tesztelést listázásra, keresésre, törlésre fogom kipróbálni.

Pontosított feladatspecifikáció

A feladat egy nyílvántartó program megírása amely megnyitáskor a forrás szöveges fájlból olvassa ki az adatokat, és dinamikusan tárolja őket objektumokba rendezve a program futásának végéig amikor visszaírja őket a szöveges fájlba. Minden műveletet (módosítás, törlés, hozzáadás) ezen a dinamikusan foglalt objektumon hajt végre a program. Az adatok megadásánál a megfelelő adattípusoknak megfelelően kell adatokat megadni, vagyis például a megjelenés évének csak számokat fogad el a program. Amennyiben nem létezik/üres a txt file, abban az esetben a program létrehozza a futásának végén.

Objektum terv

Az adatok tárolását a **Filmek** osztály egyik objektumában fogom megvalósítani. Minden filmet vagy sima, családi vagy dokumentumfilmként fog tárolni a program. A **Filmek** osztály tárolja az elemek számát így könnyebb kezelni a dinamikus foglalásokat és az adatok hozzáadását. A program elkészítése során igyekszem minnél könnyebben bővíthető objektumszerkezetet létrehozni, ennek érdekében egy új objektum megírásához mindössze konstruktor és getter, setter lesz szükséges, és 1 sor változtatás a kódban.

Algoritmusok

A feladatom egyik legösszetettebb algoritmus az adatok beolvasása amit egy külön függvény fog megvalósítani. A függvény argumentuma egy filmek objektum amit a mainben hoz létre a program. A beolvasás a txt fájl mintája szerint fog történni és annak egyedi mintája szerint minden sorból kiolvassa az odaillő elemeket. Minden beolvasás elején a txt fájlban van egy betű ami a film típusára utal, ennek megfelelően megtudja a program határozni az egyes

filmekhez tartozó adatok számát. A filmek bekérése az eof-ig történik. Addig minden filmből objektumot hoz létre a program amit behelyez a mainból kapott filmek típusú objektumba. A keresés filmek címei között zajlik, amelyekből azokat tekinti találatnak melyekben szerepel ugyanaz a részlet, kis és nagybetűket egyként kezeli.

A tesztprogram működése

A tesztprogram egy előre elkészített eseménysorozatot fog lejátszani és ezek helyességét fogja kiértékelni. A tesztesetek a program működését fogják szimulálni, mivel az adatok beolvasása nem kivételt hanem újbóli bekérést eredményez ezért azokra nem terjed ki. (viszont ezeket is megvalósítja a program) A tesztet a jporta definiálásával lehet elindítani de lehetőség lesz a program kipróbálásra is.

A tesztprogram lefuttat a lehető legtöbb esetet, de g_test segítségével csak az előre megírt tesztesetek futnak. ↩
Viszont a nem ellenőrzött tesztesetek eredménye az hogy a program majdnem minden függvénye meghívódik és a filmlista.txt változatlan marad hiszen a teszt a törlést is meghívja. (Erre a hibás bekérések miatti ellenőrzésre van szükség.)

A dokumentáció

A doxygen program segítségével. Ékezetek nélkül, mivel különböző karaktereket nem jelenít meg a program.

Bővíthetőség

A program bővítéséhez a [filmtipusok.h](#)-ban kell egy új classt definiálni getter, setter, konstruktorral és a segédfgvek. ↩
cpp-ben a customconst fgvben beleírni hogy a megfelelő jelnél a jó konstruktor hívódjon meg.

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

gtest_lite	Gtest_lite: a keretrendszer függvényinek és objektumainak névtére	11
----------------------------	-----------------------------------------------------------------------------	----

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

<code>_Is_Types< F, T ></code>	13
<code>_registry_item</code>	14
<code>call_t</code>	14
<code>Film</code>	17
<code>CsaladiFilm</code>	14
<code>DokumentumFilm</code>	16
<code>Filmek</code>	21
<code>gtest_lite::Test</code>	25

Chapter 4

Class Index

4.1 Class List

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

_Is_Types< F, T >	
Segédsablon típuskonverzió futás közbeni ellenőrzésere	13
_registry_item	14
call_t	14
CsaladiFilm	14
DokumentumFilm	
DokumentumFilm osztály,film leszármazottja egy plusz adatot tarol.Jelölése:D	16
Film	
Ez a film osztály.Innen származnak le a különbozo filmtípusok,filmek adatait tarolja	17
Filmek	
Filmek osztály a filmek tarolasara.Film pontereket tarol dinamikusán	21
gtest_lite::Test	25

Chapter 5

File Index

5.1 File List

Here is a list of all documented files with brief descriptions:

film.h	??
filmtipusok.h	??
gtest_lite.h	27
main.cpp	33
memtrace.h	??
segedfgvek.h	34

Chapter 6

Namespace Documentation

6.1 gtest_lite Namespace Reference

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

Classes

- struct [Test](#)

Functions

- `template<typename T1 , typename T2 >
std::ostream & EXPECT_ (T1 exp, T2 act, bool(*pred)(T1, T2), 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 *, T2 *), 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 T1 , typename T2 >
bool eq (T1 a, T2 b)`
- `bool eqstr (const char *a, const char *b)`
- `bool eqstrcase (const char *a, const char *b)`
- `template<typename T1 , typename T2 >
bool ne (T1 a, T2 b)`
- `bool nestr (const char *a, const char *b)`
- `template<typename T1 , typename T2 >
bool le (T1 a, T2 b)`
- `template<typename T1 , typename T2 >
bool lt (T1 a, T2 b)`
- `template<typename T1 , typename T2 >
bool ge (T1 a, T2 b)`
- `template<typename T1 , typename T2 >
bool gt (T1 a, T2 b)`
- `template<typename T >
bool almostEQ (T a, T b)`

6.1.1 Detailed Description

`gtest_lite`: a keretrendszer függvényinek és objektumainak névtére

6.1.2 Function Documentation

6.1.2.1 `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>

6.1.2.2 `eq()`

```
template<typename T1 , typename T2 >
bool gtest_lite::eq (
    T1 a,
    T2 b )
```

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

6.1.2.3 `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.

Chapter 7

Class Documentation

7.1 `_Is_Types< F, T >` Struct Template Reference

Segédsablon típuskonverzió futás közbeni ellenőrzésére.

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

Static Public Member Functions

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

Static Public Attributes

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

7.1.1 Detailed Description

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

Segédsablon típuskonverzió futás közbeni ellenőrzésére.

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

- [gtest_lite.h](#)

7.2 `_registry_item` Struct Reference

Public Attributes

- `void * p`
- `size_t size`
- `call_t call`
- `struct _registry_item * next`

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

- `memtrace.cpp`

7.3 `call_t` Struct Reference

Public Attributes

- `int f`
- `int line`
- `char * par_txt`
- `char * file`

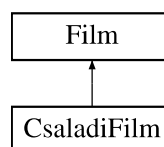
The documentation for this struct was generated from the following file:

- `memtrace.cpp`

7.4 `CsaladiFilm` Class Reference

```
#include <filmtipusok.h>
```

Inheritance diagram for `CsaladiFilm`:



Public Member Functions

- `CsaladiFilm (Film &myfilm)`
- `CsaladiFilm (std::string nev, int hossz, int kiadas, char jeloles, std::string plusdata)`
- `std::string getPlusData ()`
- `void setPlusData (std::string uj)`

7.4.1 Detailed Description

@ brief [CsaladiFilm](#) osztaly,film leszarmazottja tarolja a korhatart is.Jelolese:C

7.4.2 Constructor & Destructor Documentation

7.4.2.1 CsaladiFilm() [1/2]

```
CsaladiFilm::CsaladiFilm (
    Film & myfilm ) [inline]
```

Konstruktor a CsaladiFilmhez.Egy filmbol hoz létre dokumentumfilmet.

7.4.2.2 CsaladiFilm() [2/2]

```
CsaladiFilm::CsaladiFilm (
    std::string nev,
    int hossz,
    int kiadas,
    char jeloles,
    std::string plusdata ) [inline]
```

Konstruktor a CsaladiFilmhez.A megadott adatok alapján hoz létre egy dokumentumfilmet.

7.4.3 Member Function Documentation

7.4.3.1 getPlusData()

```
std::string CsaladiFilm::getPlusData ( ) [inline], [virtual]
```

Getter a csaladifilmek plusz adatahoz,stringgel ter vissza.

Reimplemented from [Film](#).

7.4.3.2 setPlusData()

```
void CsaladiFilm::setPlusData (
    std::string uj ) [inline], [virtual]
```

Setter a csaladifilmek plusz adatahoz,string bemenetet allit be korhatarnak.

Reimplemented from [Film](#).

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

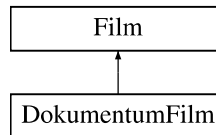
- filmtipusok.h

7.5 DokumentumFilm Class Reference

[DokumentumFilm](#) osztaly,film leszarmazottja egy plusz adatot tarol.Jelolese:D.

```
#include <filmtipusok.h>
```

Inheritance diagram for DokumentumFilm:



Public Member Functions

- [DokumentumFilm](#) ([Film](#) &myfilm)
- [DokumentumFilm](#) (std::string nev, int hossz, int kiadas, char jeloles, std::string plusdata)
- std::string [getPlusData](#) ()
- void [setPlusData](#) (std::string uj)

7.5.1 Detailed Description

[DokumentumFilm](#) osztaly,film leszarmazottja egy plusz adatot tarol.Jelolese:D.

7.5.2 Constructor & Destructor Documentation

7.5.2.1 DokumentumFilm() [1/2]

```
DokumentumFilm::DokumentumFilm (  
    Film & myfilm ) [inline]
```

Konstruktor a DokumentumFilmhez.Egy filmbol hoz létre dokumentumfilmet.

7.5.2.2 DokumentumFilm() [2/2]

```
DokumentumFilm::DokumentumFilm (  
    std::string nev,  
    int hossz,  
    int kiadas,  
    char jeloles,  
    std::string plusdata ) [inline]
```

Konstruktor a DokumentumFilmhez.A megadott adatok alapján hoz létre egy dokumentumfilmet.

7.5.3 Member Function Documentation

7.5.3.1 getPlusData()

```
std::string DokumentumFilm::getPlusData ( ) [inline], [virtual]
```

Getter a dokumentumfilmek plusz adatahoz.

Reimplemented from [Film](#).

7.5.3.2 setPlusData()

```
void DokumentumFilm::setPlusData (
    std::string uj ) [inline], [virtual]
```

Setter a dokumentumfilmek plusz adatahoz.

Reimplemented from [Film](#).

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

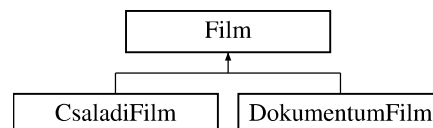
- filmtipusok.h

7.6 Film Class Reference

Ez a film osztály. Innen származnak le a különbozo filmtipusok, filmek adatait tarolja.

```
#include <film.h>
```

Inheritance diagram for Film:



Public Member Functions

- [Film](#) ()
- [Film](#) (std::string nev, int hossz, int kiadas, char jeloles)
- std::string [getNev](#) ()
- void [setNev](#) (std::string &ujnev)
- int [getHossz](#) ()
- void [setHossz](#) (int ujhossz)
- std::string [getKiirhossz](#) ()
- int [getKiadas](#) ()
- void [setKiadas](#) (int ujkiadas)
- char [getJel](#) ()
- void [setJel](#) (char ujjel)
- void [kiir](#) ()
- virtual std::string [getPlusData](#) ()
- virtual void [setPlusData](#) (std::string uj)
- void [filmModosit](#) (int answer=-1, std::string megadott="")
- virtual [~Film](#) ()

7.6.1 Detailed Description

Ez a film osztaly.Innen szarmaznak le a kulonbozo filmtipusok,filmek adatait tarolja.

7.6.2 Constructor & Destructor Documentation

7.6.2.1 Film() [1/2]

```
Film::Film ( ) [inline]
```

Default konstruktor.

7.6.2.2 Film() [2/2]

```
Film::Film (
    std::string nev,
    int hossz,
    int kiadas,
    char jeloles ) [inline]
```

Konstruktor amely a bekert adatokbol hoz létre egy uj filmet.

7.6.2.3 ~Film()

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

Virtualis destruktork a film szamara.

7.6.3 Member Function Documentation

7.6.3.1 filmModosit()

```
void Film::filmModosit (
    int answer = -1,
    std::string megadott = "" )
```

Fuggveny amely a modosito menut hozza elo,melynek segitsegevel lehet modositani a filmek adatait.

7.6.3.2 getHossz()

```
int Film::getHossz ( ) [inline]
```

Getter a hossz lekerdezesere.

7.6.3.3 getJel()

```
char Film::getJel ( ) [inline]
```

Getter a filmhez tartozo jelolesre.

7.6.3.4 getKiadas()

```
int Film::getKiadas ( ) [inline]
```

Getter a kiadas evere.

7.6.3.5 getKiirhossz()

```
std::string Film::getKiirhossz ( )
```

A hosszt stringkent adja vissza a megfelelo HH:MM:SS formatumban.

7.6.3.6 getNev()

```
std::string Film::getNev ( ) [inline]
```

Getter a nev lekerdezesere.

7.6.3.7 getPlusData()

```
virtual std::string Film::getPlusData ( ) [inline], [virtual]
```

Virtualis fuggveny amely a film plusz adatot adja vissza.

Reimplemented in [CsaladiFilm](#), and [DokumentumFilm](#).

7.6.3.8 kiir()

```
void Film::kiir ( )
```

Kiirja a film adatait.

7.6.3.9 setHossz()

```
void Film::setHossz (
    int ujhossz ) [inline]
```

Setter a hossz valtozatasara.

7.6.3.10 setJel()

```
void Film::setJel (
    char ujjel ) [inline]
```

Setter a filmhez tartozo jeles valtoztatasra.

7.6.3.11 setKiadas()

```
void Film::setKiadas (
    int ujkiadas ) [inline]
```

Setter a kiadas evenek valtoztatasara.

7.6.3.12 setNev()

```
void Film::setNev (
    std::string & ujnev ) [inline]
```

Setter a nev valtoztatasara.

7.6.3.13 setPlusData()

```
virtual void Film::setPlusData (
    std::string uj ) [inline], [virtual]
```

Virtualis fuggveny amely a film plusz adatot valtoztatja(mivel a filmnek nincsen ezert itt ures).

Reimplemented in [CsaladiFilm](#), and [DokumentumFilm](#).

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

- film.h
- filmcpp.cpp

7.7 Filmek Class Reference

[Filmek](#) osztaly a filmek tarolasara.Film pointereket tarol dinamikusan.

```
#include <film.h>
```

Public Member Functions

- [Filmek](#) ()
- int [getMeret](#) ()
- [Film](#) & [getFilm](#) (int i)
- [Film](#) * [getFilmPointer](#) (int i)
- void [setFilmPointer](#) (int i, [Film](#) *mivel)
- [Film](#) * [getFilmekPointer](#) ()
- void [setFilm](#) (int i, [Film](#) &uj)
- void [torol](#) (int i)
- void [torol](#) ([Film](#) &torlendo)
- void [add](#) ([Film](#) *addelement)
- void [add](#) (std::string nev, int hossz, int kiadas, char jeloles)
- void [kilistaz](#) ()
- void [keres](#) ([Filmek](#) &talalatok, std::string mit="")
- void [keresveTorol](#) ()
- void [keresveModosit](#) ()
- void [clear](#) ()
- void [hozzaadas](#) ()
- void [kereslistaz](#) ()
- [~Filmek](#) ()

7.7.1 Detailed Description

[Filmek](#) osztaly a filmek tarolasara.Film pointereket tarol dinamikusan.

7.7.2 Constructor & Destructor Documentation

7.7.2.1 Filmek()

```
Filmek::Filmek ( ) [inline]
```

Default konstruktor.

7.7.2.2 ~Filmek()

```
Filmek::~~Filmek ( ) [inline]
```

Destruktor a filmekhez.

7.7.3 Member Function Documentation

7.7.3.1 add() [1/2]

```
void Filmek::add (
    Film * addelement )
```

Hozzaadja a listához egy filmet amelyre a megadott pointer mutat.

7.7.3.2 add() [2/2]

```
void Filmek::add (
    std::string nev,
    int hossz,
    int kiadas,
    char jeloles )
```

A megadott adatok alapján létrehoz egy filmet amit hozzáad a filmek listájához.

7.7.3.3 clear()

```
void Filmek::clear ( )
```

Törli a tárolt filmeket, a méretet nullazza.

7.7.3.4 getFilm()

```
Film& Filmek::getFilm (
    int i ) [inline]
```

Getter amely az i. filmet adja vissza.

7.7.3.5 getFilmekPointer()

```
Film* Filmek::getFilmekPointer ( ) [inline]
```

Getter amely visszater a filmek pointerevel.

7.7.3.6 getFilmPointer()

```
Film* Filmek::getFilmPointer (
    int i ) [inline]
```

Getter amely az i. film pointerevel ter vissza.

7.7.3.7 getMeret()

```
int Filmek::getMeret ( ) [inline]
```

Getter a filmek szamara.

7.7.3.8 hozzaadas()

```
void Filmek::hozzaadas ( )
```

Uj film adatainak bekerese utan uj filmet hoz létre amit a tarolt filmek kozé helyez.

7.7.3.9 keres()

```
void Filmek::keres (
    Filmek & talalatok,
    std::string mit = "" )
```

Kereses funkciót valósítja meg, a talalatokat a megadott **Filmek** objektumba helyezi el. Lehetőség van a keresett szót is megadni.

7.7.3.10 kereslistaz()

```
void Filmek::kereslistaz ( )
```

Keresést valósít meg a tarolt filmek kozott es kiirja az eredmenyeket.

7.7.3.11 keresveModosit()

```
void Filmek::keresveModosit ( )
```

Kereses utan a talalatok kozul lehetoseg ad a fuggveny a film adatainak modositására.

7.7.3.12 keresveTorol()

```
void Filmek::keresveTorol ( )
```

Kereses funkciót valósítja meg, a vegen a találatok közül torlessel.

7.7.3.13 kilistaz()

```
void Filmek::kilistaz ( )
```

Kiírja a tarolt filmeket.

7.7.3.14 setFilm()

```
void Filmek::setFilm (
    int i,
    Film & uj )
```

Setter amely az i. filmet törli és hozzáadja helyette az új megadott filmet.

7.7.3.15 setFilmPointer()

```
void Filmek::setFilmPointer (
    int i,
    Film * mivel ) [inline]
```

Setter amely az i. film pointeret beállítja a megadott film pointerre.

7.7.3.16 torol() [1/2]

```
void Filmek::torol (
    Film & torlendo )
```

Törli a megadott filmet a listából.

7.7.3.17 torol() [2/2]

```
void Filmek::torol (
    int i )
```

Törli az i. filmet a listából.

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

- film.h
- filmcpp.cpp

7.8 gtest_lite::Test Struct Reference

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

Public Member Functions

- 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.

Static Public Member Functions

- static **Test** & **getTest** ()

Public Attributes

- 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

7.8.1 Detailed Description

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

7.8.2 Member Function Documentation

7.8.2.1 `getTest()`

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

< egyedüli (singleton) példány

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

- [gtest_lite.h](#)

Chapter 8

File Documentation

8.1 gtest_lite.h File Reference

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

Classes

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

Namespaces

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

Macros

- `#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 *) eltéréset 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 *) eltéréset 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(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 EXPECTTHROW(statement, exp, act)`
EXPECTTHROW: kivételkezelés.
- `#define ASSERTTHROW(statement, exp, act)`
- `#define ASSERT_(expected, actual, fn, op)`
- `#define GTINIT(IS)`
- `#define GTEND(os)`

Functions

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

8.1.1 Detailed Description

(v3/2019)

Google gtest keretrendszerhez hasonló rendszer. Sz.I. 2015., 2016., 2017. (_Has_X) Sz.I. 2018 (template), ENDM, ENDMsg, nullptr Sz.I. 2019 singleton Sz.I. 2021 ASSERT..., STRCASE... Sz.I. 2021 EXPECT_REGEX

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-
SSERT... 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.

8.1.2 Macro Definition Documentation

8.1.2.1 ASSERT_

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

Value:

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

8.1.2.2 ASSERT_EQ

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

Azonosságot elváró makró

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

8.1.2.3 ASSERT_NO_THROW [1/2]

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

Value:

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

Nem várunk kivételt.

8.1.2.4 ASSERT_NO_THROW [2/2]

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

Value:

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

Nem várunk kivételt.

8.1.2.5 ASSERTTHROW

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

Value:

```
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; }
```

8.1.2.6 CREATE_Has_

```
#define CREATE_Has_  
    X )
```

Value:

```
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)...`

8.1.2.7 ENDM

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

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

8.1.2.8 ENDMsg

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

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

8.1.2.9 EXPECT_ANY_THROW

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

Value:

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

Kivételt várunk.

8.1.2.10 EXPECT_NO_THROW

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

Value:

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

Nem várunk kivételt.

8.1.2.11 EXPECT_THROW

```
#define EXPECT_THROW(  
    statement,  
    exception_type )
```

Value:

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

Kivételt várunk.

8.1.2.12 EXPECT_THROW_THROW

```
#define EXPECT_THROW_THROW(  
    statement,  
    exception_type )
```

Value:

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

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

8.1.2.13 EXPECTTHROW

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

Value:

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

EXPECTTHROW: kivételkezelés.

Belső megvalósításhoz tartozó makrók, és osztályok.

8.1.2.14 Nem célszerű közvetlenül használni, vagy módosítani

8.1.2.15 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.

Parameters

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

8.1.3 Function Documentation

8.1.3.1 hasMember()

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

Segédfüggvény egy publikus adattag, vagy tagfüggvény létezésének tesztelésére fordítási időben

8.2 main.cpp File Reference

```
#include "film.h"
#include "segedfgvek.h"
#include "memtrace.h"
#include "gtest_lite.h"
#include <iostream>
#include <fstream>
```

Functions

- void [test_0](#) (Filmek &lista)
- void [test_1](#) (Filmek &lista)
- void [test_2](#) (Filmek &lista)
- void [test_3](#) (Filmek &lista)
- void [test_4](#) (Filmek &lista)
- void [test_5](#) (Filmek &lista)
- int [main](#) ()

8.2.1 Detailed Description

A program futása vezénylo cpp file. Az ellenorzesek es a funckiok meghivasat kezeli.

8.2.2 Function Documentation

8.2.2.1 test_0()

```
void test_0 (
    Filmek & lista )
```

Ellenorzes a beolvasara.

8.2.2.2 test_1()

```
void test_1 (
    Filmek & lista )
```

Ellenorzes a filmek hozzaadasara.

8.2.2.3 test_2()

```
void test_2 (
    Filmek & lista )
```

Ellenorzes a keresesre.

8.2.2.4 test_3()

```
void test_3 (
    Filmek & lista )
```

Ellenorzes a torlesre.

8.2.2.5 test_4()

```
void test_4 (
    Filmek & lista )
```

Ellenorzes a modositasra.

8.2.2.6 test_5()

```
void test_5 (
    Filmek & lista )
```

Ellenorzes a visszairasra.

8.3 segedfgvek.h File Reference

```
#include "film.h"
#include "filmtipusok.h"
```

Functions

- void [menukiir](#) ()
- void [clrscreen](#) ()
- void [txtTorol](#) ()
- [Film](#) * [customconst](#) (std::string, int, int, char, std::string)
- int [getszam](#) ()
- char [getjel](#) ()
- void [beolvas](#) ([Filmek](#) &lista)
- void [visszair](#) ([Filmek](#) &lista)

8.3.1 Detailed Description

This header contains the functions wich is to help the other functions.

8.3.2 Function Documentation

8.3.2.1 beolvas()

```
void beolvas (
    Filmek & lista )
```

This function is to read all the movies and store them in the given `Filmek` object.

8.3.2.2 clrscreen()

```
void clrscreen ( )
```

Function to clear the screen.

8.3.2.3 customconst()

```
Film* customconst (
    std::string nev,
    int hossz,
    int kiadas,
    char jel,
    std::string sor )
```

This function is to decide witch constructor to call.

8.3.2.4 getjel()

```
char getjel ( )
```

This function is to read in a char,if the input is invalid then another entering is called.

8.3.2.5 getszam()

```
int getszam ( )
```

This is a safe input for numbers,if the input is invalid then another entering is called.

8.3.2.6 menukiir()

```
void menukiir ( )
```

Function to display the menu.

8.3.2.7 txtTorol()

```
void txtTorol ( )
```

This function is to delete the txt file if it exists.

8.3.2.8 visszair()

```
void visszair (
    Filmek & lista )
```

This function is to overwrite the "filmlista.txt",and write all the stored movies into it.

Index

[_Is_Types< F, T >, 13](#)

[_registry_item, 14](#)

[~Film](#)

[Film, 18](#)

[~Filmek](#)

[Filmek, 22](#)

[add](#)

[Filmek, 22](#)

[almostEQ](#)

[gtest_lite.h, 12](#)

[ASSERT_](#)

[gtest_lite.h, 30](#)

[ASSERT_EQ](#)

[gtest_lite.h, 30](#)

[ASSERT_NO_THROW](#)

[gtest_lite.h, 30, 31](#)

[ASSERTTHROW](#)

[gtest_lite.h, 31](#)

[beolvas](#)

[segedfgvek.h, 34](#)

[call_t, 14](#)

[clear](#)

[Filmek, 22](#)

[clrscreen](#)

[segedfgvek.h, 35](#)

[CREATE_Has_](#)

[gtest_lite.h, 31](#)

[CsaladiFilm, 14](#)

[CsaladiFilm, 15](#)

[getPlusData, 15](#)

[setPlusData, 15](#)

[customconst](#)

[segedfgvek.h, 35](#)

[DokumentumFilm, 16](#)

[DokumentumFilm, 16](#)

[getPlusData, 17](#)

[setPlusData, 17](#)

[ENDM](#)

[gtest_lite.h, 31](#)

[ENDMsg](#)

[gtest_lite.h, 31](#)

[eq](#)

[gtest_lite, 12](#)

[EXPECT_ANY_THROW](#)

[gtest_lite.h, 31](#)

[EXPECT_NO_THROW](#)

[gtest_lite.h, 32](#)

[EXPECT_THROW](#)

[gtest_lite.h, 32](#)

[EXPECT_THROW_THROW](#)

[gtest_lite.h, 32](#)

[EXPECTSTR](#)

[gtest_lite, 12](#)

[EXPECTTHROW](#)

[gtest_lite.h, 32](#)

[Film, 17](#)

[~Film, 18](#)

[Film, 18](#)

[filmModosit, 19](#)

[getHossz, 19](#)

[getJel, 19](#)

[getKiadás, 19](#)

[getKiirhossz, 19](#)

[getNev, 19](#)

[getPlusData, 19](#)

[kiir, 20](#)

[setHossz, 20](#)

[setJel, 20](#)

[setKiadás, 20](#)

[setNev, 20](#)

[setPlusData, 20](#)

[Filmek, 21](#)

[~Filmek, 22](#)

[add, 22](#)

[clear, 22](#)

[Filmek, 21](#)

[getFilm, 22](#)

[getFilmekPointer, 22](#)

[getFilmPointer, 23](#)

[getMeret, 23](#)

[hozzaadás, 23](#)

[keres, 23](#)

[kereslistaz, 23](#)

[keresveModosit, 23](#)

[keresveTorol, 23](#)

[kilstaz, 24](#)

[setFilm, 24](#)

[setFilmPointer, 24](#)

[torol, 24](#)

[filmModosit](#)

[Film, 19](#)

[getFilm](#)

[Filmek, 22](#)

[getFilmekPointer](#)

- Filmek, [22](#)
- getFilmPointer
 - Filmek, [23](#)
- getHossz
 - Film, [19](#)
- getJel
 - Film, [19](#)
- getjel
 - segedfgvek.h, [35](#)
- getKiadas
 - Film, [19](#)
- getKiirhossz
 - Film, [19](#)
- getMeret
 - Filmek, [23](#)
- getNev
 - Film, [19](#)
- getPlusData
 - CsaladiFilm, [15](#)
 - DokumentumFilm, [17](#)
 - Film, [19](#)
- getszam
 - segedfgvek.h, [35](#)
- getTest
 - gtest_lite::Test, [25](#)
- gtest_lite, [11](#)
 - almostEQ, [12](#)
 - eq, [12](#)
 - EXPECTSTR, [12](#)
- gtest_lite.h, [27](#)
 - ASSERT_, [30](#)
 - ASSERT_EQ, [30](#)
 - ASSERT_NO_THROW, [30](#), [31](#)
 - ASSERTTHROW, [31](#)
 - CREATE_Has_, [31](#)
 - ENDM, [31](#)
 - ENDMsg, [31](#)
 - EXPECT_ANY_THROW, [31](#)
 - EXPECT_NO_THROW, [32](#)
 - EXPECT_THROW, [32](#)
 - EXPECT_THROW_THROW, [32](#)
 - EXPECTTHROW, [32](#)
 - hasMember, [33](#)
 - TEST, [32](#)
- gtest_lite::Test, [25](#)
 - getTest, [25](#)
- hasMember
 - gtest_lite.h, [33](#)
- hozzaadas
 - Filmek, [23](#)
- keres
 - Filmek, [23](#)
- kereslistaz
 - Filmek, [23](#)
- keresveModosit
 - Filmek, [23](#)
- keresveTorol
- Filmek, [23](#)
- kiir
 - Film, [20](#)
- kilistaz
 - Filmek, [24](#)
- main.cpp, [33](#)
 - test_0, [33](#)
 - test_1, [33](#)
 - test_2, [34](#)
 - test_3, [34](#)
 - test_4, [34](#)
 - test_5, [34](#)
- menukiir
 - segedfgvek.h, [35](#)
- segedfgvek.h, [34](#)
 - beolvas, [34](#)
 - clrscreen, [35](#)
 - customconst, [35](#)
 - getjel, [35](#)
 - getszam, [35](#)
 - menukiir, [35](#)
 - txtTorol, [35](#)
 - visszair, [35](#)
- setFilm
 - Filmek, [24](#)
- setFilmPointer
 - Filmek, [24](#)
- setHossz
 - Film, [20](#)
- setJel
 - Film, [20](#)
- setKiadas
 - Film, [20](#)
- setNev
 - Film, [20](#)
- setPlusData
 - CsaladiFilm, [15](#)
 - DokumentumFilm, [17](#)
 - Film, [20](#)
- TEST
 - gtest_lite.h, [32](#)
- test_0
 - main.cpp, [33](#)
- test_1
 - main.cpp, [33](#)
- test_2
 - main.cpp, [34](#)
- test_3
 - main.cpp, [34](#)
- test_4
 - main.cpp, [34](#)
- test_5
 - main.cpp, [34](#)
- torol
 - Filmek, [24](#)
- txtTorol

segedfgvek.h, [35](#)

visszair

segedfgvek.h, [35](#)