

Telefonkönyv

Generated by Doxygen 1.8.18

1 A program leírása:	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Class Documentation	9
5.1 Contact Class Reference	9
5.2 Menu Class Reference	9
5.2.1 Detailed Description	9
5.2.2 Constructor & Destructor Documentation	10
5.2.2.1 Menu()	10
5.2.3 Member Function Documentation	10
5.2.3.1 runItem()	10
5.3 Phonebook Class Reference	10
5.3.1 Member Function Documentation	11
5.3.1.1 getContactsWithId()	11
5.3.1.2 isEmpty()	12
5.3.1.3 listContactType()	12
5.3.1.4 searchByName()	12
5.3.1.5 searchByNumber()	13
5.3.1.6 searchContactsFor()	13
5.4 Private Class Reference	13
5.4.1 Detailed Description	14
5.4.2 Member Function Documentation	14
5.4.2.1 getAddress()	14
5.4.2.2 getBirthday()	15
5.4.2.3 getId()	15
5.4.2.4 getNickname()	15
5.4.2.5 readFromFile()	15
5.4.2.6 toString()	16
5.4.2.7 writeToFile()	16
5.5 String Class Reference	16
5.5.1 Detailed Description	17
5.5.2 Constructor & Destructor Documentation	17
5.5.2.1 String() [1/3]	17
5.5.2.2 String() [2/3]	17
5.5.2.3 String() [3/3]	17
5.5.3 Member Function Documentation	18

5.5.3.1 <code>c_str()</code>	18
5.5.3.2 <code>operator+()</code> [1/2]	18
5.5.3.3 <code>operator+()</code> [2/2]	18
5.5.3.4 <code>operator=()</code>	19
5.5.3.5 <code>operator[]()</code> [1/2]	19
5.5.3.6 <code>operator[]()</code> [2/2]	19
5.5.3.7 <code>printDbg()</code>	20
5.5.3.8 <code>size()</code>	20
5.6 <code>Vector< T ></code> Class Template Reference	20
5.6.1 Constructor & Destructor Documentation	21
5.6.1.1 <code>Vector()</code> [1/2]	21
5.6.1.2 <code>Vector()</code> [2/2]	21
5.6.2 Member Function Documentation	22
5.6.2.1 <code>deleteItem()</code>	22
5.6.2.2 <code>getSize()</code>	22
5.6.2.3 <code>operator=()</code>	22
5.6.2.4 <code>operator[]()</code>	23
5.6.2.5 <code>push_back()</code>	23
5.6.2.6 <code>swap()</code>	23
5.7 Work Class Reference	24
5.7.1 Detailed Description	24
5.7.2 Member Function Documentation	24
5.7.2.1 <code>getCompany()</code>	25
5.7.2.2 <code>getId()</code>	25
5.7.2.3 <code>getWebsite()</code>	25
5.7.2.4 <code>readFromFile()</code>	25
5.7.2.5 <code>toString()</code>	26
5.7.2.6 <code>writeToFile()</code>	26
6 File Documentation	27
6.1 <code>/Users/tomi/Desktop/Phonebook/contact.hpp</code> File Reference	27
6.1.1 Detailed Description	27
6.2 <code>/Users/tomi/Desktop/Phonebook/menu.cpp</code> File Reference	27
6.2.1 Detailed Description	27
6.3 <code>/Users/tomi/Desktop/Phonebook/menu.hpp</code> File Reference	28
6.3.1 Detailed Description	28
6.4 <code>/Users/tomi/Desktop/Phonebook/phone.cpp</code> File Reference	28
6.4.1 Detailed Description	28
6.5 <code>/Users/tomi/Desktop/Phonebook/phone.hpp</code> File Reference	28
6.5.1 Detailed Description	29
6.6 <code>/Users/tomi/Desktop/Phonebook/phonebook.cpp</code> File Reference	29
6.6.1 Detailed Description	29

6.7 /Users/tomi/Desktop/Phonebook/phonebook.hpp File Reference	29
6.7.1 Detailed Description	29
6.8 /Users/tomi/Desktop/Phonebook/string5.cpp File Reference	30
6.8.1 Function Documentation	30
6.8.1.1 operator<<()	30
6.8.1.2 operator>>()	30
6.9 /Users/tomi/Desktop/Phonebook/string5.h File Reference	31
6.9.1 Detailed Description	31
6.9.2 Function Documentation	31
6.9.2.1 operator+()	31
6.9.2.2 operator<<()	32
6.9.2.3 operator>>()	32
6.10 /Users/tomi/Desktop/Phonebook/vector.hpp File Reference	32
6.10.1 Detailed Description	32
Index	33

Chapter 1

A program leírása:

A program egy telefonkönyvként funkcionál. Két féle kontaktot lehet benne tárolni, munkahelyi számokat, és otthoni számokat. Hozzá lehet adni kontaktokat, és kitörölni, lehet listázni ezeket, aztán az adatokat lementeni egy fájlba, ahonnan ezek később visszatölthetők lesznek. Keresni is lehet név vagy telefonszám szerint. Pintér Tamás - JY4D5L

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Contact	9
Private	13
Work	24
Menu	9
Phonebook	10
String	16
Vector< T >	20
Vector< Contact * >	20

Chapter 3

Class Index

3.1 Class List

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

Contact	Contact class létrehozása	9
Menu	Menu class létrehozása	9
Phonebook	10
Private	Private class létrehozása	13
String	16
Vector< T >	20
Work	Work class létrehozása	24

Chapter 4

File Index

4.1 File List

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

/Users/tomi/Desktop/Phonebook/ contact.hpp	
Contact class	27
/Users/tomi/Desktop/Phonebook/ menu.cpp	
Menu class függvényei	27
/Users/tomi/Desktop/Phonebook/ menu.hpp	
Menu class	28
/Users/tomi/Desktop/Phonebook/ phone.cpp	
Phone class függvényei	28
/Users/tomi/Desktop/Phonebook/ phone.hpp	
Phone class függvényei	28
/Users/tomi/Desktop/Phonebook/ phonebook.cpp	
Phonebook class függvényei	29
/Users/tomi/Desktop/Phonebook/ phonebook.hpp	
Phonebook class	29
/Users/tomi/Desktop/Phonebook/ string5.cpp	30
/Users/tomi/Desktop/Phonebook/ string5.h	31
/Users/tomi/Desktop/Phonebook/ test.hpp	??
/Users/tomi/Desktop/Phonebook/ vector.hpp	
Vector class	32

Chapter 5

Class Documentation

5.1 Contact Class Reference

[Contact](#) class létrehozása.

```
#include <contact.hpp>
```

Inheritance diagram for Contact:

5.2 Menu Class Reference

[Menu](#) class létrehozása.

```
#include <menu.hpp>
```

Public Member Functions

- [Menu](#) ([Phonebook](#) &phonebook)
[Menu](#) CTOR Létrehoz egy Menüt egy telefonkönyvvel.
- void [run](#) ()
Futtatja a menüt.
- void [runItem](#) (int index, bool *isRunning)
Futtat egy menüpontot.

Static Public Member Functions

- static void [showMenu](#) ()
Kiírja a menüt a felhasználónak.

5.2.1 Detailed Description

[Menu](#) class létrehozása.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 Menu()

```
Menu::Menu (
    Phonebook & phonebook ) [inline], [explicit]
```

[Menu](#) CTOR Létrehoz egy Menüt egy telefonkönyvvel.

Parameters

<i>phonebook</i>	ezt kapja meg telefonkönyvnek
------------------	-------------------------------

5.2.3 Member Function Documentation

5.2.3.1 runItem()

```
void Menu::runItem (
    int index,
    bool * isRunning )
```

Futtat egy menüpontot.

Parameters

<i>index</i>	ezzel a sorszámmal
<i>isRunning</i>	megkapja, hogy a programnak kell-e még futnia

Here is the call graph for this function: Here is the caller graph for this function:

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

- [/Users/tomi/Desktop/Phonebook/menu.hpp](#)
- [/Users/tomi/Desktop/Phonebook/menu.cpp](#)

5.3 Phonebook Class Reference

Public Member Functions

- [Phonebook](#) ()
[Phonebook](#) CTOR.

- `Vector< Contact * > searchByName (String &name)`
Kilistázza azokat a kontaktokat amik nevében szerepel a "name" szó
- `Vector< Contact * > searchByNumber (String &number)`
Kilistázza azokat a kontaktokat amik telefonszámában szerepel a "number" szám.
- `Vector< Contact * > getContactsWithId (int id) const`
Visszatér a megfelelő típusú kontaktokkal.
- `void listContacts (std::ostream &)`
Kilistázza a work és private Vectorok adatait a megadott ostreamre.
- `void addContact (Contact *const &)`
Hozzáad egy contactot a telefonkönyvhöz.
- `void removeContact ()`
Kitöröl egy kontaktot az indexe alapján (ezt a futása közben kéri be)
- `void loadFromFile ()`
Felveszi a database.txt-ben talált kontaktokat a telefonkönyvbe.
- `void saveToFile ()`
Elmenti az összes kontaktot a database.txt fileba.
- `~Phonebook ()`
Phonebook DTOR Kitörli az összes kontaktot a memóriából.

Static Public Member Functions

- static `Vector< Contact * > searchContactsFor (const Vector< Contact * > &contacts, bool(*searchCommand)(const Contact *, const String &), const String &pattern)`
Kereső függvény.
- static `void listContactType (const Vector< Contact * > &contacts, std::ostream &)`
Kilistázza a kontaktokat a megadott ostreamre.
- static `bool isEmpty (const char *filename)`
Megmondja hogy az adott file üres-e.

5.3.1 Member Function Documentation

5.3.1.1 getContactsWithId()

```
Vector<Contact *> Phonebook::getContactsWithId (
    int id ) const [inline]
```

Visszatér a megfelelő típusú kontaktokkal.

Parameters

<i>id</i>	Ilyen kontakt típusú Vektorral tér vissza
-----------	---

Here is the call graph for this function:

5.3.1.2 isFileEmpty()

```
bool Phonebook::isFileEmpty (
    const char * filename ) [static]
```

Megmondja hogy az adott file üres-e.

Parameters

<i>filename</i>	ezt a filet teszteli
-----------------	----------------------

Returns

üres-e a file

5.3.1.3 listContactType()

```
void Phonebook::listContactType (
    const Vector< Contact * > & contacts,
    std::ostream & os ) [static]
```

Kilistázza a kontaktokat a megadott ostreamre.

Parameters

<i>contacts</i>	Ezeket listázza
-----------------	-----------------

Here is the call graph for this function:

5.3.1.4 searchByName()

```
Vector< Contact * > Phonebook::searchByName (
    String & name )
```

Kilistázza azokat a kontaktokat amik nevében szerepel a "name" szó

Parameters

<i>name</i>	ez alapján a név alapján listáz
-------------	---------------------------------

Returns

a keresésnek megfelelt kontaktokat

Here is the call graph for this function:

5.3.1.5 searchByNumber()

```
Vector< Contact * > Phonebook::searchByNumber (
    String & number )
```

Kilistázza azokat a kontaktokat amik telefonszámában szerepel a "number" szám.

Parameters

<i>name</i>	ez alapján a név alapján listáz
-------------	---------------------------------

Returns

a keresésnek megfelelt kontaktokat

Here is the call graph for this function:

5.3.1.6 searchContactsFor()

```
Vector< Contact * > Phonebook::searchContactsFor (
    const Vector< Contact * > & contacts,
    bool(*) (const Contact *, const String &) searchCommand,
    const String & pattern ) [static]
```

Kereső függvény.

Parameters

<i>contacts</i>	Ebben a Vectorban keres
<i>searchCommand</i>	Ezzel a kereső függvénnyel
<i>pattern</i>	Ez alapján

Returns

Keresésnek megfelelt [Vector](#)

Here is the call graph for this function: Here is the caller graph for this function:

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

- /Users/tomi/Desktop/Phonebook/[phonebook.hpp](#)
- /Users/tomi/Desktop/Phonebook/[phonebook.cpp](#)

5.4 Private Class Reference

[Private](#) class létrehozása.

```
#include <phone.hpp>
```

Inheritance diagram for Private:

Collaboration diagram for Private:

Public Member Functions

- **Private** (const [String](#) &number="unknown", const [String](#) &name="unknown", const [String](#) &nickname="unknown", const [String](#) &email="unknown", const [String](#) &address="unknown", int birthday=18900101)
Private CTOR A megadott paraméterekkel létrehoóz egy private kontaktot.
- int const & [getBirthday](#) () const
Birthday getter.
- [String](#) const & [getNickname](#) () const
Nickname getter.
- [String](#) const & [getAddress](#) () const
Addressg getter.
- std::ostream & [toString](#) (std::ostream &) override
Kiírja a megadott kimenetre a private adatait formázva.
- void [writeToFile](#) (std::ostream &) override
Kiírja a megadott kimenetre a private adatait formázatlanul.
- void [readFromFile](#) (std::ifstream &) override
Beolvas a megadott kimenetről.

Static Public Member Functions

- static int [getId](#) ()
id getter

Additional Inherited Members

5.4.1 Detailed Description

[Private](#) class létrehozása.

5.4.2 Member Function Documentation

5.4.2.1 getAddress()

```
String const& Private::getAddress ( ) const [inline]
```

Addressg getter.

Returns

address

Here is the caller graph for this function:

5.4.2.2 getBirthday()

```
int const& Private::getBirthday ( ) const [inline]
```

Birthday getter.

Returns

birthday

Here is the caller graph for this function:

5.4.2.3 getId()

```
static int Private::getId ( ) [inline], [static]
```

id getter

Returns

id

Here is the caller graph for this function:

5.4.2.4 getNickname()

```
String const& Private::getNickname ( ) const [inline]
```

Nickname getter.

Returns

nickname

Here is the caller graph for this function:

5.4.2.5 readFromFile()

```
void Private::readFromFile (
    std::ifstream & os ) [override], [virtual]
```

Beolvas a megadott kimenetről.

Returns

Implements [Contact](#).

5.4.2.6 toString()

```
std::ostream & Private::toString (
    std::ostream & os ) [override], [virtual]
```

Kiírja a megadott kimenetre a private adatait formázva.

Returns

private adattagjai formázva

Implements [Contact](#).

Here is the call graph for this function:

5.4.2.7 writeToFile()

```
void Private::writeToFile (
    std::ostream & os ) [override], [virtual]
```

Kiírja a megadott kimenetre a private adatait formázatlanul.

Returns

private adattagjai formázatlanul

Implements [Contact](#).

Here is the call graph for this function:

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

- /Users/tomi/Desktop/Phonebook/[phone.hpp](#)
- /Users/tomi/Desktop/Phonebook/[phone.cpp](#)

5.5 String Class Reference

```
#include <string5.h>
```

Public Member Functions

- `size_t` [size](#) () const
- `const char *` [c_str](#) () const
- [String](#) (char ch)
Konstruktor: egy char karakterből (createStrFromChar)
- [String](#) (const char *p="")
- [String](#) (const [String](#) &s1)
- `virtual` [~String](#) ()
Destruktor.
- `void` [printDbg](#) (const char *txt="") const
- [String](#) & [operator=](#) (const [String](#) &rhs_s)
- [String](#) [operator+](#) (const [String](#) &rhs_s) const
- [String](#) [operator+](#) (char rhs_c) const
- `char` & [operator\[\]](#) (unsigned int idx)
- `const char` & [operator\[\]](#) (unsigned int idx) const

5.5.1 Detailed Description

[String](#) osztály. A pData-ban vannak a karakterek (a lezáró nullával együtt), len a hossz. A hosszba nem számít bele a lezáró nulla.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 String() [1/3]

```
String::String (  
    char ch )
```

Konstruktor: egy char karakterből (createStrFromChar)

Konstruktor egy char karakterből

Parameters

<i>ch</i>	- karakter
-----------	------------

Here is the caller graph for this function:

5.5.2.2 String() [2/3]

```
String::String (  
    const char * p = "" )
```

Konstruktor egy nullával lezárt char sorozatból Ez a default is!

Parameters

<i>p</i>	- pointer egy C sztringre
----------	---------------------------

5.5.2.3 String() [3/3]

```
String::String (  
    const String & s1 )
```

Másoló konstruktor

Parameters

<i>s1</i>	- String , amiből létrehozuk az új String-et
-----------	--

5.5.3 Member Function Documentation

5.5.3.1 `c_str()`

```
const char* String::c_str ( ) const [inline]
```

Default konstruktor [String\(\)](#) :pData(0), len(0) {} helyett ""-val inicializáljuk a const char*-osban C-sztringet ad vissza

Returns

pinter egy '\0'-val lezárt (C) sztringre

Here is the caller graph for this function:

5.5.3.2 `operator+()` [1/2]

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

Sztrinhez karaktert összefűz

Parameters

<i>rhs</i> ↔ _c	- jobboldali karakter
--------------------	-----------------------

Returns

új [String](#), ami tartalmazza a sztringet és a karaktert egymás után

Here is the call graph for this function:

5.5.3.3 `operator+()` [2/2]

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

Két Stringet összefűz

Parameters

<i>rhs</i> ↔ _s	- jobboldali String
--------------------	-------------------------------------

Returns

új [String](#), ami tartalmazza a két stringet egymás után

5.5.3.4 operator=()

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

Értékadó operátor.

Parameters

<i>rhs</i> ↔ _s	- jobboldali String
--------------------	-------------------------------------

Returns

baoldali (módosított) string (referenciája)

5.5.3.5 operator[]() [1/2]

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

A string egy megadott indexű elemének REFERENCIÁJÁVAL tér vissza.

Parameters

<i>idx</i>	- karakter indexe
------------	-------------------

Returns

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

5.5.3.6 operator[]() [2/2]

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

A string egy megadott indexű elemének REFERENCIÁJÁVAL tér vissza.

Parameters

<i>idx</i>	- karakter indexe
------------	-------------------

Returns

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

5.5.3.7 printDbg()

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

Kiírunk egy Stringet (debug célokra) Előtte kiírunk egy tetszőleges szöveget.

Parameters

<i>txt</i>	- nullával lezárt szövegre mutató pointer
------------	---

5.5.3.8 size()

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

Hossz lekérdezése.

Returns

Sztring hossza

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

- [/Users/tomi/Desktop/Phonebook/string5.h](#)
- [/Users/tomi/Desktop/Phonebook/string5.cpp](#)

5.6 Vector< T > Class Template Reference**Public Member Functions**

- [Vector](#) (size_t size=0)
ennyi elemet tárolunk benne éppen
- [~Vector](#) ()
Vector DTOR.

- [Vector](#) (const [Vector](#) &vector)
Copy CTOR.
- [Vector](#) & [operator=](#) (const [Vector](#) &vector)
Ertekado operator.
- T & [operator\[\]](#) (size_t index) const
Indexelő operátor.
- size_t [getSize](#) () const
méret getter
- void [increaseSize](#) ()
Átméretezi a vektort egyel nagyobbra.
- void [push_back](#) (const T &item)
hozzárag egy elemet a tárolóhoz
- void [swap](#) (size_t index1, size_t index2)
megcserél két tárolt elemet az indexeik alapján
- void [deleteltem](#) (size_t index)
kitöröl egy elemet a tárolóbol

5.6.1 Constructor & Destructor Documentation

5.6.1.1 Vector() [1/2]

```
template<typename T >
Vector< T >::Vector (
    size_t size = 0 ) [inline]
```

ennyi elemet tárolunk benne éppen

[Vector](#) CTOR

Parameters

<i>size</i>	size of the vector
-------------	--------------------

5.6.1.2 Vector() [2/2]

```
template<typename T >
Vector< T >::Vector (
    const Vector< T > & vector ) [inline]
```

Copy CTOR.

Parameters

<i>vector</i>	ezt a vektort másoljuk le
---------------	---------------------------

5.6.2 Member Function Documentation

5.6.2.1 deleteItem()

```
template<typename T >
void Vector< T >::deleteItem (
    size_t index ) [inline]
```

kitöröl egy elemet a tárolóbol

Parameters

<i>index</i>	ezt az indexű tagot törli
--------------	---------------------------

Here is the caller graph for this function:

5.6.2.2 getSize()

```
template<typename T >
size_t Vector< T >::getSize ( ) const [inline]
```

méret getter

Returns

méret

Here is the caller graph for this function:

5.6.2.3 operator=()

```
template<typename T >
Vector& Vector< T >::operator= (
    const Vector< T > & vector ) [inline]
```

Ertekado operator.

Parameters

<i>vector</i>	ezt az értéket adjuk át
---------------	-------------------------

Returns

*this

5.6.2.4 operator[]()

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

Indexelő operátor.

Parameters

<i>index</i>	ennél az indexnél lévő adatot adjuk vissza
--------------	--

Returns

array[index]

5.6.2.5 push_back()

```
template<typename T >
void Vector< T >::push_back (
    const T & item ) [inline]
```

hozzárak egy elemet a tárolóhoz

Parameters

<i>t</i>	A kapott elem referenciája.
----------	-----------------------------

Here is the caller graph for this function:

5.6.2.6 swap()

```
template<typename T >
void Vector< T >::swap (
    size_t index1,
    size_t index2 ) [inline]
```

megcserél két tárolt elemet az indexeik alapján

Parameters

<i>index1</i>	1. elem
<i>index2</i>	2. elem

Here is the caller graph for this function:

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

- /Users/tomi/Desktop/Phonebook/[vector.hpp](#)

5.7 Work Class Reference

[Work](#) class létrehozása.

```
#include <phone.hpp>
```

Inheritance diagram for Work:

Collaboration diagram for Work:

Public Member Functions

- [Work](#) (const [String](#) &number="unknown", const [String](#) &name="unknown", const [String](#) &email="unknown", const [String](#) &company="unknown", const [String](#) &website="unknown")
Work CTOR A megadott adatokkal létrehoz egy work kontaktot.
- [String](#) const & [getCompany](#) () const
Company getter.
- [String](#) const & [getWebsite](#) () const
Website getter.
- std::ostream & [toString](#) (std::ostream &) override
Kiírja a megadott kimenetre a work adatait formázva.
- void [writeToFile](#) (std::ostream &) override
Kiírja a megadott kimenetre a work adatait formázatlanul.
- void [readFromFile](#) (std::ifstream &) override
Beolvas a megadott kimenetről.

Static Public Member Functions

- static int [getId](#) ()
id getter

Additional Inherited Members

5.7.1 Detailed Description

[Work](#) class létrehozása.

5.7.2 Member Function Documentation

5.7.2.1 getCompany()

```
String const& Work::getCompany ( ) const [inline]
```

Company getter.

Returns

company

Here is the caller graph for this function:

5.7.2.2 getId()

```
static int Work::getId ( ) [inline], [static]
```

id getter

Returns

id

Here is the caller graph for this function:

5.7.2.3 getWebsite()

```
String const& Work::getWebsite ( ) const [inline]
```

Website getter.

Returns

website

Here is the caller graph for this function:

5.7.2.4 readFromFile()

```
void Work::readFromFile (
    std::ifstream & os ) [override], [virtual]
```

Beolvas a megadott kimenetről.

Returns

Implements [Contact](#).

5.7.2.5 toString()

```
std::ostream & Work::toString (
    std::ostream & os ) [override], [virtual]
```

Kíírja a megadott kimenetre a work adatait formázva.

Returns

work adattagjai formázva

Implements [Contact](#).

Here is the call graph for this function:

5.7.2.6 writeToFile()

```
void Work::writeToFile (
    std::ostream & os ) [override], [virtual]
```

Kíírja a megadott kimenetre a work adatait formázatlanul.

Returns

work adattagjai formázatlanul

Implements [Contact](#).

Here is the call graph for this function:

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

- /Users/tomi/Desktop/Phonebook/[phone.hpp](#)
- /Users/tomi/Desktop/Phonebook/[phone.cpp](#)

Chapter 6

File Documentation

6.1 /Users/tomi/Desktop/Phonebook/contact.hpp File Reference

[Contact](#) class.

```
#include "string5.h"
```

Include dependency graph for contact.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class [Contact](#)
[Contact](#) class létrehozása.

6.1.1 Detailed Description

[Contact](#) class.

6.2 /Users/tomi/Desktop/Phonebook/menu.cpp File Reference

[Menu](#) class függvényei.

```
#include <iostream>
#include "menu.hpp"
#include "memtrace.h"
```

Include dependency graph for menu.cpp:

6.2.1 Detailed Description

[Menu](#) class függvényei.

6.3 /Users/tomi/Desktop/Phonebook/menu.hpp File Reference

[Menu](#) class.

```
#include "phonebook.hpp"
```

Include dependency graph for menu.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class [Menu](#)
[Menu](#) class létrehozása.

6.3.1 Detailed Description

[Menu](#) class.

6.4 /Users/tomi/Desktop/Phonebook/phone.cpp File Reference

Phone class függvényei.

```
#include "phone.hpp"  
#include "memtrace.h"
```

Include dependency graph for phone.cpp:

6.4.1 Detailed Description

Phone class függvényei.

6.5 /Users/tomi/Desktop/Phonebook/phone.hpp File Reference

Phone class függvényei.

```
#include <iostream>  
#include "contact.hpp"  
#include "string5.h"
```

Include dependency graph for phone.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class [Work](#)
[Work](#) class létrehozása.
- class [Private](#)
[Private](#) class létrehozása.

6.5.1 Detailed Description

Phone class függvényei.

6.6 /Users/tomi/Desktop/Phonebook/phonebook.cpp File Reference

[Phonebook](#) class függvényei.

```
#include "phonebook.hpp"
#include <fstream>
#include "memtrace.h"
Include dependency graph for phonebook.cpp:
```

Functions

- bool **equalFromHere** ([String](#) const &str, [String](#) const &pattern)
- bool **containsName** (const [Contact](#) *contact, const [String](#) &pattern)
- bool **containsNumber** (const [Contact](#) *contact, const [String](#) &pattern)

6.6.1 Detailed Description

[Phonebook](#) class függvényei.

6.7 /Users/tomi/Desktop/Phonebook/phonebook.hpp File Reference

[Phonebook](#) class.

```
#include "contact.hpp"
#include "phone.hpp"
#include "string5.h"
#include "vector.hpp"
```

Include dependency graph for phonebook.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class [Phonebook](#)

6.7.1 Detailed Description

[Phonebook](#) class.

6.8 /Users/tomi/Desktop/Phonebook/string5.cpp File Reference

```
#include <iostream>
#include <cstring>
#include "string5.h"
#include "memtrace.h"
Include dependency graph for string5.cpp:
```

Functions

- `std::ostream & operator<< (std::ostream &os, const String &s0)`
- `std::istream & operator>> (std::istream &is, String &s0)`

6.8.1 Function Documentation

6.8.1.1 operator<<()

```
std::ostream& operator<< (
    std::ostream & os,
    const String & s0 )
```

Globális függvények: kír az ostream-re

Parameters

<i>os</i>	- ostream típusú objektum
<i>s0</i>	- String, amit kírunk

Returns

os

Here is the call graph for this function:

6.8.1.2 operator>>()

```
std::istream& operator>> (
    std::istream & is,
    String & s0 )
```

Beolvas az istream-ről egy szót egy string-be.

Parameters

<i>is</i>	- istream típusú objektum
<i>s0</i>	- String, amibe beolvas

Returns

is

6.9 /Users/tomi/Desktop/Phonebook/string5.h File Reference

```
#include <iostream>
```

Include dependency graph for string5.h: This graph shows which files directly or indirectly include this file:

Classes

- class [String](#)

Functions

- `std::ostream & operator<<` (`std::ostream &os`, `const String &s0`)
- `std::istream & operator>>` (`std::istream &is`, `String &s0`)
- `String operator+` (`char ch`, `const String &str`)

6.9.1 Detailed Description

Ez a fájl tartalmazza a [String](#) osztály deklarációját és inline függvényeit.

6.9.2 Function Documentation

6.9.2.1 operator+()

```
String operator+ (  
    char ch,  
    const String & str ) [inline]
```

Karakterhez sztringet fűz

Parameters

<i>ch</i>	- karakter
<i>str</i>	- String

Returns

új [String](#), ami tartalmazza a karaktert és a sztringet egymás után

6.9.2.2 operator<<()

```
std::ostream& operator<< (
    std::ostream & os,
    const String & s0 )
```

Globális függvények: kiír az ostream-re

Parameters

<i>os</i>	- ostream típusú objektum
<i>s0</i>	- String , amit kiírunk

Returns

os

Here is the call graph for this function:

6.9.2.3 operator>>()

```
std::istream& operator>> (
    std::istream & is,
    String & s0 )
```

Beolvas az istream-ről egy szót egy string-be.

Parameters

<i>is</i>	- istream típusú objektum
<i>s0</i>	- String , amibe beolvas

Returns

is

6.10 /Users/tomi/Desktop/Phonebook/vector.hpp File Reference

[Vector](#) class.

```
#include <cstdlib>
#include "memtrace.h"
```

Include dependency graph for vector.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class [Vector< T >](#)

6.10.1 Detailed Description

[Vector](#) class.

Index

/Users/tomi/Desktop/Phonebook/contact.hpp, 27
/Users/tomi/Desktop/Phonebook/menu.cpp, 27
/Users/tomi/Desktop/Phonebook/menu.hpp, 28
/Users/tomi/Desktop/Phonebook/phone.cpp, 28
/Users/tomi/Desktop/Phonebook/phone.hpp, 28
/Users/tomi/Desktop/Phonebook/phonebook.cpp, 29
/Users/tomi/Desktop/Phonebook/phonebook.hpp, 29
/Users/tomi/Desktop/Phonebook/string5.cpp, 30
/Users/tomi/Desktop/Phonebook/string5.h, 31
/Users/tomi/Desktop/Phonebook/vector.hpp, 32

c_str
 String, 18
Contact, 9

deleteItem
 Vector< T >, 22

getAddress
 Private, 14
getBirthday
 Private, 14
getCompany
 Work, 24
getContactsWithId
 Phonebook, 11
getId
 Private, 15
 Work, 25
getNickname
 Private, 15
getSize
 Vector< T >, 22
getWebsite
 Work, 25

isEmpty
 Phonebook, 11

listContactType
 Phonebook, 12

Menu, 9
 Menu, 10
 runItem, 10

operator<<
 string5.cpp, 30
 string5.h, 31
operator>>
 string5.cpp, 30

 string5.h, 32
operator+
 String, 18
 string5.h, 31
operator=
 String, 19
 Vector< T >, 22
operator[]
 String, 19
 Vector< T >, 22

Phonebook, 10
 getContactsWithId, 11
 isEmpty, 11
 listContactType, 12
 searchByName, 12
 searchByNumber, 12
 searchContactsFor, 13
printDbg
 String, 20
Private, 13
 getAddress, 14
 getBirthday, 14
 getId, 15
 getNickname, 15
 readFromFile, 15
 toString, 15
 writeToFile, 16
push_back
 Vector< T >, 23

readFromFile
 Private, 15
 Work, 25
runItem
 Menu, 10

searchByName
 Phonebook, 12
searchByNumber
 Phonebook, 12
searchContactsFor
 Phonebook, 13
size
 String, 20
String, 16
 c_str, 18
 operator+, 18
 operator=, 19
 operator[], 19

- printDbg, [20](#)
 - size, [20](#)
 - String, [17](#)
- string5.cpp
 - operator<<, [30](#)
 - operator>>, [30](#)
- string5.h
 - operator<<, [31](#)
 - operator>>, [32](#)
 - operator+, [31](#)
- swap
 - Vector< T >, [23](#)
- toString
 - Private, [15](#)
 - Work, [25](#)
- Vector
 - Vector< T >, [21](#)
- Vector< T >, [20](#)
 - deleteItem, [22](#)
 - getSize, [22](#)
 - operator=, [22](#)
 - operator[], [22](#)
 - push_back, [23](#)
 - swap, [23](#)
 - Vector, [21](#)
- Work, [24](#)
 - getCompany, [24](#)
 - getId, [25](#)
 - getWebsite, [25](#)
 - readFromFile, [25](#)
 - toString, [25](#)
 - writeToFile, [26](#)
- writeToFile
 - Private, [16](#)
 - Work, [26](#)