

## Telefonkönyv

Generated by Doxygen 1.8.18



<b>1 A program leírása:</b>	<b>1</b>
<b>2 Hierarchical Index</b>	<b>3</b>
2.1 Class Hierarchy . . . . .	3
<b>3 Class Index</b>	<b>5</b>
3.1 Class List . . . . .	5
<b>4 File Index</b>	<b>7</b>
4.1 File List . . . . .	7
<b>5 Class Documentation</b>	<b>9</b>
5.1 Contact Class Reference . . . . .	9
5.1.1 Detailed Description . . . . .	10
5.1.2 Member Function Documentation . . . . .	10
5.1.2.1 getEmail() . . . . .	10
5.1.2.2 getName() . . . . .	10
5.1.2.3 getNumber() . . . . .	10
5.2 Menu Class Reference . . . . .	11
5.2.1 Detailed Description . . . . .	11
5.2.2 Constructor & Destructor Documentation . . . . .	11
5.2.2.1 Menu() . . . . .	11
5.2.3 Member Function Documentation . . . . .	11
5.2.3.1 runItem() . . . . .	12
5.3 Phonebook Class Reference . . . . .	12
5.3.1 Member Function Documentation . . . . .	13
5.3.1.1 convertToContact() [1/2] . . . . .	13
5.3.1.2 convertToContact() [2/2] . . . . .	13
5.3.1.3 getPrivateContacts() . . . . .	14
5.3.1.4 getWorkContacts() . . . . .	14
5.3.1.5 isEmpty() . . . . .	14
5.3.1.6 listContactType() . . . . .	16
5.3.1.7 searchByName() . . . . .	16
5.3.1.8 searchByNumber() . . . . .	16
5.3.1.9 searchContactsFor() . . . . .	17
5.4 Private Class Reference . . . . .	17
5.4.1 Detailed Description . . . . .	18
5.4.2 Member Function Documentation . . . . .	18
5.4.2.1 getAddress() . . . . .	18
5.4.2.2 getBirthday() . . . . .	18
5.4.2.3 getNickname() . . . . .	19
5.4.2.4 toString() . . . . .	19
5.5 String Class Reference . . . . .	19
5.5.1 Detailed Description . . . . .	20

5.5.2 Constructor & Destructor Documentation	20
5.5.2.1 String() [1/3]	20
5.5.2.2 String() [2/3]	20
5.5.2.3 String() [3/3]	20
5.5.3 Member Function Documentation	21
5.5.3.1 c_str()	21
5.5.3.2 operator+() [1/2]	21
5.5.3.3 operator+() [2/2]	21
5.5.3.4 operator=()	22
5.5.3.5 operator[]() [1/2]	22
5.5.3.6 operator[]() [2/2]	22
5.5.3.7 printDbg()	23
5.5.3.8 size()	23
5.6 Vector< T > Class Template Reference	23
5.6.1 Constructor & Destructor Documentation	24
5.6.1.1 Vector() [1/2]	24
5.6.1.2 Vector() [2/2]	24
5.6.2 Member Function Documentation	25
5.6.2.1 deleteItem()	25
5.6.2.2 getSize()	25
5.6.2.3 operator=()	25
5.6.2.4 operator[]()	26
5.6.2.5 push_back()	26
5.6.2.6 swap()	26
5.7 Work Class Reference	27
5.7.1 Detailed Description	27
5.7.2 Member Function Documentation	27
5.7.2.1 getCompany()	27
5.7.2.2 getWebsite()	28
5.7.2.3 toString()	28
<b>6 File Documentation</b>	<b>29</b>
6.1 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/contact.hpp File Reference	29
6.1.1 Detailed Description	29
6.2 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/menu.cpp File Reference	29
6.2.1 Detailed Description	29
6.3 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/menu.hpp File Reference	30
6.3.1 Detailed Description	30
6.4 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/phone.cpp File Reference	30
6.4.1 Detailed Description	30
6.5 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/phone.hpp File Reference	30
6.5.1 Detailed Description	31

---

6.6 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/phonebook.cpp File Reference . . . . .	31
6.6.1 Detailed Description . . . . .	31
6.7 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/phonebook.hpp File Reference . . . . .	31
6.7.1 Detailed Description . . . . .	31
6.8 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/string5.cpp File Reference . . . . .	32
6.8.1 Function Documentation . . . . .	32
6.8.1.1 operator<<() . . . . .	32
6.8.1.2 operator>>() . . . . .	32
6.9 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/string5.h File Reference . . . . .	33
6.9.1 Detailed Description . . . . .	33
6.9.2 Function Documentation . . . . .	33
6.9.2.1 operator+() . . . . .	33
6.9.2.2 operator<<() . . . . .	34
6.9.2.3 operator>>() . . . . .	34
6.10 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/vector.hpp File Reference . . . . .	34
6.10.1 Detailed Description . . . . .	35
<b>Index</b>	<b>37</b>



## 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 . . . . .	17
Work . . . . .	27
Menu . . . . .	11
Phonebook . . . . .	12
String . . . . .	19
Vector< T > . . . . .	23
Vector< Private * > . . . . .	23
Vector< Work * > . . . . .	23



## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">Contact</a>		
	<a href="#">Contact</a> class létrehozása . . . . .	9
<a href="#">Menu</a>		
	<a href="#">Menu</a> class létrehozása . . . . .	11
<a href="#">Phonebook</a>		
	<a href="#">Phonebook</a> . . . . .	12
<a href="#">Private</a>		
	<a href="#">Private</a> class létrehozása . . . . .	17
<a href="#">String</a>		
	<a href="#">String</a> . . . . .	19
<a href="#">Vector&lt; T &gt;</a>		
	<a href="#">Vector&lt; T &gt;</a> . . . . .	23
<a href="#">Work</a>		
	<a href="#">Work</a> class létrehozása . . . . .	27



## Chapter 4

# File Index

### 4.1 File List

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

/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/ <a href="#">contact.hpp</a>	
<a href="#">Contact</a> class . . . . .	29
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/ <a href="#">menu.cpp</a>	
<a href="#">Menu</a> class függvényei . . . . .	29
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/ <a href="#">menu.hpp</a>	
<a href="#">Menu</a> class . . . . .	30
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/ <a href="#">phone.cpp</a>	
Phone class függvényei . . . . .	30
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/ <a href="#">phone.hpp</a>	
Phone class függvényei . . . . .	30
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/ <a href="#">phonebook.cpp</a>	
<a href="#">Phonebook</a> class függvényei . . . . .	31
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/ <a href="#">phonebook.hpp</a>	
<a href="#">Phonebook</a> class . . . . .	31
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/ <a href="#">string5.cpp</a>	32
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/ <a href="#">string5.h</a>	33
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/ <a href="#">test.hpp</a>	??
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/ <a href="#">vector.hpp</a>	
<a href="#">Vector</a> class . . . . .	34



## Chapter 5

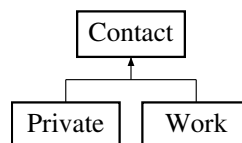
# Class Documentation

### 5.1 Contact Class Reference

`Contact` class létrehozása.

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

Inheritance diagram for `Contact`:



#### Public Member Functions

- `Contact` (const `String` &number, const `String` &name, const `String` &email)  
*`Contact` CTOR A megadott értékekkel létrehoz egy `Contactot`.*
- `String getNumber` () const  
*Number getter.*
- `String getName` () const  
*Name getter.*
- `String getEmail` () const  
*Email getter.*
- virtual std::ostream & `toString` (std::ostream &)=0  
*A `toString` tiszta virtuális függvénye.*
- virtual `~Contact` ()=default  
*`Contact` DTOR.*

#### Protected Attributes

- `String number`  
*A közös tulajdonságok number, name, email.*
- `String name`
- `String email`

### 5.1.1 Detailed Description

[Contact](#) class létrehozása.

### 5.1.2 Member Function Documentation

#### 5.1.2.1 getEmail()

```
String Contact::getEmail ( ) const [inline]
```

Email getter.

##### Returns

email

#### 5.1.2.2 getName()

```
String Contact::getName ( ) const [inline]
```

Name getter.

##### Returns

name

#### 5.1.2.3 getNumber()

```
String Contact::getNumber ( ) const [inline]
```

Number getter.

##### Returns

number

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

- [/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/contact.hpp](#)



## 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

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

- [/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/menu.hpp](#)
- [/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/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.*
- [Work \\* getWorkContacts](#) (size\_t index) const  
*Visszatér a workContacts egy elemével.*
- [Private \\* getPrivateContacts](#) (size\_t index) const  
*Visszatér a workContacts egy elemével.*
- void [listContacts](#) (std::ostream &)  
*Kilistázza a work és private Vectorok adatait a megadott ostreamre.*
- void [addContact](#) (Work \*const &)  
*hozzáad egy work Vectort a telefonkönyvhöz*
- void [addContact](#) (Private \*const &)  
*Hozzáad egy private Vectort 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 * > convertToContact (Vector< Work * > const &workContacts)`  
*Konvertál `Vector<Work*>`-ről `Vector<Contact*>`-ra.*
- static `Vector< Contact * > convertToContact (Vector< Private * > const &privateContacts)`  
*Konvertál `Vector<Private*>`-ről `Vector<Contact*>`-ra.*
- static `Vector< Contact * > searchContactsFor (const Vector< Contact * > &contacts, bool(*search← Command)(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.*
- static void `saveContactsToDB (std::ostream &, Work *)`  
*Elmenti a database.txt fileba a work kontaktokat.*
- static void `saveContactsToDB (std::ostream &, Private *)`  
*Elmenti a database.txt fileba a private kontaktokat.*

## 5.3.1 Member Function Documentation

### 5.3.1.1 convertToContact() [1/2]

```
Vector< Contact * > Phonebook::convertToContact (
    Vector< Private * > const & privateContacts ) [static]
```

Konvertál `Vector<Private*>`-ről `Vector<Contact*>`-ra.

#### Parameters

<code>privateContacts</code>	ezt a Vectort konvertálja
------------------------------	---------------------------

#### Returns

átkonvertált `Vector`

### 5.3.1.2 convertToContact() [2/2]

```
Vector< Contact * > Phonebook::convertToContact (
    Vector< Work * > const & workContacts ) [static]
```

Konvertál `Vector<Work*>`-ről `Vector<Contact*>`-ra.

**Parameters**

<i>workContacts</i>	ezt a Vectort konvertálja
---------------------	---------------------------

**Returns**

átkonvertált [Vector](#)

**5.3.1.3 getPrivateContacts()**

```
Private* Phonebook::getPrivateContacts (
    size_t index ) const [inline]
```

Visszatér a workContacts egy elemével.

**Parameters**

<i>index</i>	Ezzel az elemével tér vissza
--------------	------------------------------

**Returns**

privateContacts[i]

**5.3.1.4 getWorkContacts()**

```
Work* Phonebook::getWorkContacts (
    size_t index ) const [inline]
```

Visszatér a workContacts egy elemével.

**Parameters**

<i>index</i>	Ezzel az elemével tér vissza
--------------	------------------------------

**Returns**

workContacts[index]

**5.3.1.5 isFileEmpty()**

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

Megmondja hogy az adott file üres-e.

## Parameters

<i>filename</i>	ezt a file-t teszteli
-----------------	-----------------------

## Returns

üres-e a file

### 5.3.1.6 listContactType()

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

Kilistázza a kontaktokat a megadott ostreamre.

## Parameters

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

### 5.3.1.7 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

### 5.3.1.8 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

## 5.3.1.9 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 függvénnnyel
<i>pattern</i>	Ez alapján

## Returns

Keresésnek megfelelt [Vector](#)

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

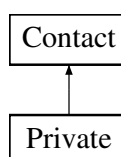
- /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/[phonebook.hpp](#)
- /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/[phonebook.cpp](#)

## 5.4 Private Class Reference

[Private](#) class létrehozása.

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

Inheritance 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 `getBirthday` () const  
*Birthday getter.*
- `String` `getNickname` () const  
*Nickname getter.*
- `String` `getAddress` () const  
*Addressg getter.*
- `std::ostream & toString` (std::ostream &) override  
*Kiírja a megadott kimenetre a private adatait formázva.*

## Additional Inherited Members

### 5.4.1 Detailed Description

`Private` class létrehozása.

### 5.4.2 Member Function Documentation

#### 5.4.2.1 getAddress()

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

Addressg getter.

#### Returns

address

#### 5.4.2.2 getBirthday()

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

Birthday getter.

#### Returns

birthday



### 5.4.2.3 getNickname()

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

Nickname getter.

#### Returns

nickname

### 5.4.2.4 toString()

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

Kiírja a megadott kimenetre a private adatait formázva.

#### Returns

private adattagjai formázva

Implements [Contact](#).

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

- /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/[phone.hpp](#)
- /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/[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
-----------	------------

#### 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>	- <a href="#">String</a> , 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

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

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

Sztrinhez karaktert összefűz

##### Parameters

<code>rhs_c</code>	- jobboldali karakter
--------------------	-----------------------

##### Returns

új `String`, ami tartalmazza a sztringet és a karaktert egymás után

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

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

Két Stringet összefűz

##### Parameters

<code>rhs_s</code>	- jobboldali <code>String</code>
--------------------	----------------------------------

**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 <a href="#">String</a>
--------------------	-------------------------------------

**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/Documents/CPP/phonebook-cpp/DOXY/string5.h](#)
- [/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/string5.cpp](#)

## 5.6 Vector&lt; T &gt; 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)  
*hozzárak egy elemet a tárolóhoz*
- void [swap](#) (size\_t index1, size\_t index2)  
*megcserél két tárolt elemet az indexeik alapján*
- void [deletItem](#) (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
--------------	---------------------------

### 5.6.2.2 getSize()

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

méret getter

#### Returns

méret

### 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.
----------	-----------------------------

#### 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

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

- /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/[vector.hpp](#)

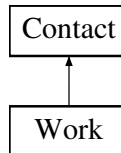


## 5.7 Work Class Reference

[Work](#) class létrehozása.

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

Inheritance 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](#) [getCompany](#) () const  
*Company getter.*
- [String](#) [getWebsite](#) () const  
*Website getter.*
- std::ostream & [toString](#) (std::ostream &) override  
*Kiírja a megadott kimenetre a work adatait formázva.*

### Additional Inherited Members

#### 5.7.1 Detailed Description

[Work](#) class létrehozása.

#### 5.7.2 Member Function Documentation

##### 5.7.2.1 [getCompany\(\)](#)

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

Company getter.

Returns

company

### 5.7.2.2 `getWebsite()`

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

Website getter.

#### Returns

website

### 5.7.2.3 `toString()`

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

Kíírja a megadott kimenetre a work adatait formázva.

#### Returns

work adattagjai formázva

Implements [Contact](#).

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

- /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/[phone.hpp](#)
- /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/[phone.cpp](#)

## Chapter 6

# File Documentation

### 6.1 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/contact.hpp File Reference

[Contact](#) class.

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

#### Classes

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

#### 6.1.1 Detailed Description

[Contact](#) class.

### 6.2 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/menu.cpp File Reference

[Menu](#) class függvényei.

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

#### 6.2.1 Detailed Description

[Menu](#) class függvényei.

## 6.3 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/menu.hpp File Reference

[Menu](#) class.

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

### Classes

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

#### 6.3.1 Detailed Description

[Menu](#) class.

## 6.4 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/phone.cpp File Reference

Phone class függvényei.

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

#### 6.4.1 Detailed Description

Phone class függvényei.

## 6.5 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/phone.hpp File Reference

Phone class függvényei.

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

### Classes

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

### 6.5.1 Detailed Description

Phone class függvényei.

## 6.6 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/phonebook.cpp File Reference

Phonebook class függvényei.

```
#include "phonebook.hpp"  
#include <fstream>  
#include "memtrace.h"
```

### 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/Documents/CPP/phonebook-cpp/DOXY/phonebook.hpp File Reference

Phonebook class.

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

### Classes

- class [Phonebook](#)

### 6.7.1 Detailed Description

Phonebook class.

## 6.8 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/string5.cpp File Reference

```
#include <iostream>
#include <cstring>
#include "memtrace.h"
#include "string5.h"
```

### 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: kiír az ostream-re

#### Parameters

<i>os</i>	- ostream típusú objektum
<i>s0</i>	- <a href="#">String</a> , amit kiírunk

#### Returns

os

#### 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>	- <a href="#">String</a> , amibe beolvas

## Returns

is

## 6.9 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/string5.h File Reference

```
#include <iostream>
```

## 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>	- <a href="#">String</a>

**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>	- <a href="#">String</a> , amit kiírnak

**Returns**

os

**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>	- <a href="#">String</a> , amibe beolvas

**Returns**

is

## 6.10 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/vector.hpp File Reference

[Vector](#) class.

```
#include <cstdlib>
```



## Classes

- class [Vector< T >](#)

### 6.10.1 Detailed Description

[Vector](#) class.



# Index

/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/contact.hpp, 29  
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/menu.cpp, 27  
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/menu.hpp, 14  
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/phonebook.cpp, 30  
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/phonebook.h, 30  
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/phonebook.hpp, 31  
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/phonebook.cpp, 31  
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/string5.cpp, 32  
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/string5.h, 33  
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/vector.hpp, 34

c\_str  
String, 21  
Contact, 9  
getEmail, 10  
getName, 10  
getNumber, 10  
convertToContact  
Phonebook, 13  
deleteItem  
Vector< T >, 25  
getAddress  
Private, 18  
getBirthday  
Private, 18  
getCompany  
Work, 27  
getEmail  
Contact, 10  
getName  
Contact, 10  
getNickname  
Private, 18  
getNumber  
Contact, 10  
getPrivateContacts  
Phonebook, 14  
getSize  
Vector< T >, 25  
getWebsite  
Work, 27  
getWorkContacts  
Phonebook, 14  
isEmpty  
Phonebook, 14  
listContactType  
Phonebook, 16  
Menu, 11  
Menu, 11  
runItem, 11  
operator<<  
string5.cpp, 32  
string5.h, 34  
operator>>  
string5.cpp, 32  
string5.h, 34  
operator+  
String, 21  
string5.h, 33  
operator=  
String, 22  
Vector< T >, 25  
operator[]  
String, 22  
Vector< T >, 25  
Phonebook, 12  
convertToContact, 13  
getPrivateContacts, 14  
getWorkContacts, 14  
isEmpty, 14  
listContactType, 16  
searchByName, 16  
searchByNumber, 16  
searchContactsFor, 17  
printDbg  
String, 23  
Private, 17  
getAddress, 18  
getBirthday, 18  
getNickname, 18  
toString, 19  
push\_back  
Vector< T >, 26

- runItem
  - Menu, [11](#)
- searchByName
  - Phonebook, [16](#)
- searchByNumber
  - Phonebook, [16](#)
- searchContactsFor
  - Phonebook, [17](#)
- size
  - String, [23](#)
- String, [19](#)
  - c\_str, [21](#)
  - operator+, [21](#)
  - operator=, [22](#)
  - operator[], [22](#)
  - printDbg, [23](#)
  - size, [23](#)
  - String, [20](#)
- string5.cpp
  - operator<<, [32](#)
  - operator>>, [32](#)
- string5.h
  - operator<<, [34](#)
  - operator>>, [34](#)
  - operator+, [33](#)
- swap
  - Vector< T >, [26](#)
- toString
  - Private, [19](#)
  - Work, [28](#)
- Vector
  - Vector< T >, [24](#)
- Vector< T >, [23](#)
  - deleteItem, [25](#)
  - getSize, [25](#)
  - operator=, [25](#)
  - operator[], [25](#)
  - push\_back, [26](#)
  - swap, [26](#)
  - Vector, [24](#)
- Work, [27](#)
  - getCompany, [27](#)
  - getWebsite, [27](#)
  - toString, [28](#)