

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 convertToContact() [1/2]	11
5.3.1.2 convertToContact() [2/2]	12
5.3.1.3 getPrivateContacts()	12
5.3.1.4 getWorkContacts()	12
5.3.1.5 isEmpty()	14
5.3.1.6 listContactType()	14
5.3.1.7 searchByName()	14
5.3.1.8 searchByNumber()	15
5.3.1.9 searchContactsFor()	15
5.4 Private Class Reference	16
5.4.1 Detailed Description	16
5.4.2 Member Function Documentation	16
5.4.2.1 getAddress()	16
5.4.2.2 getBirthday()	17
5.4.2.3 getNickname()	17
5.4.2.4 toString()	17
5.5 Vector< T > Class Template Reference	18
5.5.1 Constructor & Destructor Documentation	18
5.5.1.1 Vector() [1/2]	18
5.5.1.2 Vector() [2/2]	18
5.5.2 Member Function Documentation	19
5.5.2.1 deleteItem()	19
5.5.2.2 getSize()	19

5.5.2.3 operator=()	19
5.5.2.4 operator[]()	20
5.5.2.5 push_back()	20
5.5.2.6 swap()	20
5.6 Work Class Reference	21
5.6.1 Detailed Description	21
5.6.2 Member Function Documentation	21
5.6.2.1 getCompany()	22
5.6.2.2 getWebsite()	22
5.6.2.3 toString()	22
6 File Documentation	23
6.1 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/contact.hpp File Reference	23
6.1.1 Detailed Description	23
6.2 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/menu.cpp File Reference	23
6.2.1 Detailed Description	23
6.3 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/menu.hpp File Reference	24
6.3.1 Detailed Description	24
6.4 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/phone.cpp File Reference	24
6.4.1 Detailed Description	24
6.5 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/phone.hpp File Reference	24
6.5.1 Detailed Description	25
6.6 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/phonebook.cpp File Reference	25
6.6.1 Detailed Description	25
6.7 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/phonebook.hpp File Reference	25
6.7.1 Detailed Description	25
6.8 /Users/tomi/Documents/CPP/phonebook-cpp/DOXY/vector.hpp File Reference	26
6.8.1 Detailed Description	26
Index	27

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	16
Work	21
Menu	9
Phonebook	10
Vector< T >	18
Vector< Private * >	18
Vector< Work * >	18

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	16
Vector< T >		18
Work		
	Work class létrehozása	21

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/ contact.hpp	
Contact class	23
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/ menu.cpp	
Menu class függvényei	23
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/ menu.hpp	
Menu class	24
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/ phone.cpp	
Phone class függvényei	24
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/ phone.hpp	
Phone class függvényei	24
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/ phonebook.cpp	
Phonebook class függvényei	25
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/ phonebook.hpp	
Phonebook class	25
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/ test.hpp	??
/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/ vector.hpp	
Vector class	26

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/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 (std::string &name)`
Kilistázza azokat a kontaktokat amik nevében szerepel a "name" szó
- `Vector< Contact * > searchByNumber (std::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 loadFromEmpty ()`
Ha nem talált a loadFromFile adatokat a database.txt-ben, akkor feltölti üres adatokkal a telefonkönyvet.
- `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 std::string &), const std::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

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

Returns

átkonvertált [Vector](#)

Here is the call graph for this function:

5.3.1.2 convertToContact() [2/2]

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

Konvertál [Vector<Work*>](#)-ról [Vector<Contact*>](#)-ra.

Parameters

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

Returns

átkonvertált [Vector](#)

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

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]

Here is the caller graph for this function:

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]

Here is the caller graph for this function:

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 filet 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
-----------------	-----------------

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

5.3.1.7 searchByName()

```
Vector< Contact * > Phonebook::searchByName (
    std::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:

5.3.1.8 searchByNumber()

```
Vector< Contact * > Phonebook::searchByNumber (
    std::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:

5.3.1.9 searchContactsFor()

```
Vector< Contact * > Phonebook::searchContactsFor (
    const Vector< Contact * > & contacts,
    bool(*) (const Contact *, const std::string &) searchCommand,
    const std::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/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:

Collaboration diagram for Private:

Public Member Functions

- [Private](#) (const std::string &[number](#)="unknown", const std::string &name="unknown", std::string nickname="unknown", const std::string &email="unknown", std::string address="unknown", int birthday=18900101)
[Private](#) CTOR A megadott paraméterekkel létrehoóz egy private kontaktot.
- int [getBirthday](#) () const
Birthday getter.
- std::string [getNickname](#) () const
Nickname getter.
- std::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()

```
std::string Private::getAddress ( ) const [inline]
```

Addressg getter.

Returns

address

Here is the caller graph for this function:

5.4.2.2 getBirthday()

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

Birthday getter.

Returns

birthday

Here is the caller graph for this function:

5.4.2.3 getNickname()

```
std::string Private::getNickname ( ) const [inline]
```

Nickname getter.

Returns

nickname

Here is the caller graph for this function:

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](#).

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

5.5 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)
hozzáarak 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

5.5.1 Constructor & Destructor Documentation

5.5.1.1 Vector() [1/2]

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

ennyi elemet tárolunk benne éppen

[Vector](#) CTOR

Parameters

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

5.5.1.2 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
---------------	---------------------------

5.5.2 Member Function Documentation

5.5.2.1 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:

5.5.2.2 getSize()

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

méret getter

Returns

méret

Here is the caller graph for this function:

5.5.2.3 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

5.5.2.4 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]

5.5.2.5 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:

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

5.6 Work Class Reference

[Work](#) class létrehozása.

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

Inheritance diagram for Work:

Collaboration diagram for Work:

Public Member Functions

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

Additional Inherited Members

5.6.1 Detailed Description

[Work](#) class létrehozása.

5.6.2 Member Function Documentation

5.6.2.1 getCompany()

```
std::string Work::getCompany ( ) const [inline]
```

Company getter.

Returns

company

Here is the caller graph for this function:

5.6.2.2 getWebsite()

```
std::string Work::getWebsite ( ) const [inline]
```

Website getter.

Returns

website

Here is the caller graph for this function:

5.6.2.3 toString()

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

Kíírja a megadott kimenetre a work adatait formázva.

Returns

work adattagjai formázva

Implements [Contact](#).

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/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 <string>
#include <utility>
```

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

Phone class függvényei.

```
#include <iostream>
```

```
#include "contact.hpp"
```

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

[Phonebook](#) class függvényei.

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

Functions

- bool **equalFromHere** (std::string const &str, std::string const &pattern)
- bool **containsName** (const [Contact](#) *contact, const std::string &pattern)
- bool **containsNumber** (const [Contact](#) *contact, const std::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 "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/Documents/CPP/phonebook-cpp/DOXY/vector.hpp File Reference

[Vector](#) class.

```
#include <cstdlib>
```

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

Classes

- class [Vector< T >](#)

6.8.1 Detailed Description

[Vector](#) class.

Index

[/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/contact.cpp](#), 23
[/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/menu.cpp](#), 23
[/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/menu.hpp](#), 24
[/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/phone.cpp](#), 24
[/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/phone.hpp](#), 24
[/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/phonebook.cpp](#), 25
[/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/phonebook.hpp](#), 25
[/Users/tomi/Documents/CPP/phonebook-cpp/DOXY/vector.hpp](#), 26

[Contact](#), 9
[convertToContact](#)
 [Phonebook](#), 11, 12

[deleteItem](#)
 [Vector< T >](#), 19

[getAddress](#)
 [Private](#), 16
[getBirthday](#)
 [Private](#), 16
[getCompany](#)
 [Work](#), 21
[getNickname](#)
 [Private](#), 17
[getPrivateContacts](#)
 [Phonebook](#), 12
[getSize](#)
 [Vector< T >](#), 19
[getWebsite](#)
 [Work](#), 22
[getWorkContacts](#)
 [Phonebook](#), 12

[isEmpty](#)
 [Phonebook](#), 14

[listContactType](#)
 [Phonebook](#), 14

[Menu](#), 9
 [Menu](#), 10
 [runItem](#), 10

[operator=](#)
 [Vector< T >](#), 19
[operator\[\]](#)
 [Vector< T >](#), 20

[Phonebook](#), 10
[convertToContact](#), 11, 12
[getPrivateContacts](#), 12
[getWorkContacts](#), 12
[isEmpty](#), 14
[listContactType](#), 14
[searchByName](#), 14
[searchByNumber](#), 15
[searchContactsFor](#), 15

[Private](#), 16
 [getAddress](#), 16
 [getBirthday](#), 16
 [getNickname](#), 17
 [toString](#), 17

[push_back](#)
 [Vector< T >](#), 20

[runItem](#)
 [Menu](#), 10

[searchByName](#)
 [Phonebook](#), 14
[searchByNumber](#)
 [Phonebook](#), 15
[searchContactsFor](#)
 [Phonebook](#), 15

[swap](#)
 [Vector< T >](#), 20

[toString](#)
 [Private](#), 17
 [Work](#), 22

[Vector](#)
 [Vector< T >](#), 18
[Vector< T >](#), 18
 [deleteItem](#), 19
 [getSize](#), 19
 [operator=](#), 19
 [operator\[\]](#), 20
 [push_back](#), 20
 [swap](#), 20
 [Vector](#), 18

[Work](#), 21
 [getCompany](#), 21

`getWebsite`, [22](#)
`toString`, [22](#)