Telefonkönyv

Generated by Doxygen 1.8.18

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 visszatölthetőek 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:

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

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

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

phonebook	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

index	ezzel a sorszámmal
isRunning	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 isFileEmpty (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

id	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

filename	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

contacts	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

name	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

name	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

contacts	Ebben a Vectorban keres
searchCommand	Ezzel a kereső függvénnyel
pattern	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 létrehoöz 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

ch	- 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 deafault is!

Parameters

p	- pointer egy C sztringre

5.5.2.3 String() [3/3]

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

Másoló konstruktor

Parameters

s1	- String, amiből létrehozzuk 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

rhs←↩ _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

rhs←↩ _s	- jobboldali String

Returns

új String, ami tartalmazza a két stringet egmás után

5.5.3.4 operator=()

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

Értékadó operátor.

Parameters

rhs↩ _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

idx	- charakter 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

idx	- 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

txt	- 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)

  hozzárak egy elemet a tárolóhoz
- void swap (size_t index1, size_t index2)

  megcserél két tárolt elemet az indexeik alapján
- void deleteItem (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

size	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

vector	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

index	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

vector	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

index	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

t	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

index1	1. elem
index2	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]
```

Kiírja 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]
```

Kiírja 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 fügvényei.

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

Include dependency graph for menu.cpp:

6.2.1 Detailed Description

Menu class fügvényei.

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 fügvényei.

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

Include dependency graph for phone.cpp:

6.4.1 Detailed Description

Phone class fügvényei.

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

Phone class fügvényei.

```
#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 fügvényei.

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

Phonebook class fügvényei.

```
#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 fügvényei.

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: kiír az ostream-re

Parameters

os	- ostream típusú objektum
s0	- String, amit kiírunk

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

is	- istream típusú objektum
s0	- 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

ch	- karakter
str	- 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

os	- ostream típusú objektum
s0	- 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

is	- istream típusú objektum
s0	- 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