KonfReader

Fébert Tamás (UQ8MSF)
Version 1.1.0

KonfHash Class Reference

Public Member Functions

- **KonfHash()**
- **KonfHash** (const **KonfHash** &)
- void **post** (const pair< string, string > &element)
- template<typename T > T **get** (const string &key)
- void **put** (const pair< string, string > &newElement)
- void **del** (const string &key)
- void **readFile** (const string &fileLocation)
- void **saveToFile** (const string &name)
- **~KonfHash()**

Constructor & Destructor Documentation

KonfHash::KonfHash () [default]

KonfHash::KonfHash (const KonfHash &) [default]

KonfHash::~KonfHash () [default]

Member Function Documentation

void KonfHash::del (const string & *key*) [inline]

del | delete a key-value pair from the table, if it exists.

Parameters

key	- String key of the value to delete

template<typename T > T KonfHash::get (const string & *key*) [inline]

get | return the value of a key-value pair in the proper type

Parameters

Template	- the preferred type of the value
key	- the key in string of the sought value

Returns

type specific value of the key

void KonfHash::post (const pair< string, string > & *element*) [inline]

post | get the hash of the inputted key and post key-value pair to the table

Parameters

element	- a key-value pair

void KonfHash::put (const pair< string, string > & *newElement*) [inline]

put | update an existing key-value pairs value

Parameters

newElement	- key-value pair

void KonfHash::readFile (const string & *fileLocation*)[inline]

readFile | reads a configuration file and post the key-value pairs to the table

Parameters

fileLocation	- the exact location of the configuration file

void KonfHash::saveToFile (const string & *name*)[inline]

saveToFile | create a file, then write all the elements of the table in the file, in the right format

Parameters

name	- name of the new save file

Private Member Functions

- `string` **`getValue`** `(const string &key)`
- `int` **`hashFunction`** `(const string &keyInString)`
- `template<typename U> U` **`toType`** `(const string &value)`
- `int` **`getPos`** `(const string &key)`
- `bool` **`isMember`** `(const string &key)`

Private Member Function Documentation

`string KonfHash::getValue (const string & key)[inline]`

getValue

Parameters

key	the key that belongs to the wanted value

Returns

the value that belongs to the key, in string

`int KonfHash::hashFunction (const string & keyInString) [inline][static]`

hashFunction

Parameters

key	key from input

Returns

hashed key

`template<typename U > U KonfHash::toType (const string & value) [inline]`

toType | get a value, convert it from string to its right type and return it

Parameters

Template	the wanted type
value	the value we want to convert

Returns

the value converted to the specific type

Throws

runtime_error	if there is a type mismatch

`int KonfHash::getPos (const string & key)[inline]`

getPos | get a key and returns its position in the array

Parameters

key	a key of a key-value pair

Returns

the position in int

Throws

runtime_error	if the key isn't a member of the table

`bool KonfHash::isMember (const string & key)[inline]`

isMember

Parameters

key	the key of the inspected key-value pair

Returns

true, if the key is already member of the table

menu.cpp File Reference

Functions

- string **getFileLocation** ()
- void **menuSwitch** (**KonfHash** konfObj)
- string **lower** (string str)
- void **clearConsole** ()

Function Documentation

string getFileLocation ()

getFileLocation | requests the location of a file on standard input

Returns

the location of the configuration file

void menuSwitch (KonfHash *konfObj*)

menuSwitch | basically the menu of the program, this is the user interface

Parameters

konfObj	- a konfHash object

string lower () [static]

lower | source: https://stackoverflow.com/questions/313970/how-to-convert-an-instance-of-stdstring-to-lower-case

Returns

the string with all letters transformed to lowercase

void clearConsole ()

clearConsole | clears the console

KonfReader felhasználói dokumentáció

KonfReader – Fébert Tamás (UQ8MSF)

A program használata

A program konfigurációs fájlok beolvasására, az azok által tartalmazott adatok kezelésére, adatok hozzáadására, frissítésére, törlésére és az adatok új fájlba való kiírására lett készítve.

A program indulásakor bekéri egy konfigurációs fájl helyét, amit az általános DOS path formátumban kell megadni. Ezek után a program feldob egy menüt, amiben lehet választani, hogy a felhasználó meg akar e tekinteni egy adatot, hozzáadni szeretne, frissíteni szeretne, esetleg törölni szeretne-e egy adatot. Továbbá be lehet kérni a programba egy új fájlt, aminek az adatait beolvassa a már eltárolt adatok mellé. A program végső soron képes kiírni az általa az indításától kezdve eltárolt összes adatot egy új, vagy akár egy meglévő fájlba is.

A programból a menüben felkínált opcióval lehet kilépni. A program csak az angol abc betűit képes kezelni, így elvárt csak azon karakterek használata. A program hibás bemenet esetén lehetőséget biztosít a bemenet helyes bevitelére.

A menürendszer

1 – érték visszakérése

A program bekéri az értékpár kulcsát és az érték típusát, majd visszaadja az értéket típushelyesen.

2 – új értékpár bevitele

A program kér egy kulcsot, majd egy értéket, amit aztán mint egy értékpár eltárol.

3 – meglévő értékpár frissítése

A program kér egy kulcsot majd egy értéket és frissíti a kulcshoz tartozó értéket a bekért értékre.

4 – meglévő értékpár törlése

A program kér egy kulcsot, majd törli az ahhoz tartozó értékpárt.

5 – új fájl beolvasása

A program bekéri egy fájl helyét, amit aztán beolvas.

6 – értékek fájlba való mentése

A program kér egy nevet, amilyen néven létrehoz egy új fájlt és az összes értékpárt kiírja a fájlba.

7 – kilépés

A program kilép.

A program UML diagramja