

KonfReader

Fébert Tamás (UQ8MSF)
Version 1.0.1

Hierarchical Index

Class Hierarchy

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

 KonfHash5

Class Index

Class List

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

KonfHash5

File Index

File List

Here is a list of all files with brief descriptions:

konf_hash.hpp	7
menu.cpp	12
menu.h	13

KonfHash Class Reference

```
#include <konf_hash.hpp>
```

Public Member Functions

- **KonfHash** ()=default
- **KonfHash** (const **KonfHash** &)=default
- 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** ()
- template<> string **get** (const string &key)

Constructor & Destructor Documentation

KonfHash::KonfHash () [default]

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

KonfHash::~~KonfHash () [inline], [default]

Member Function Documentation

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

del | deletes a key-value pair from the table

Parameters

<i>key</i>	- String key of the value to delete
------------	-------------------------------------

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

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

Parameters

<i>key</i>	- the key in string of the sought value
------------	---

Returns

type specific value of the key

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

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

<i>element</i>	- a key-value pair
----------------	--------------------

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

put | update an existing key-value pairs value

Parameters

<i>newElement</i>	- 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

<i>fileLocation</i>	- 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

<i>name</i>	- name of the new save file
-------------	-----------------------------

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

konf_hash.hpp

File Documentation

konf_hash.hpp File Reference

```
#include <string>
#include <iostream>
#include <vector>
#include <valarray>
#include <sstream>
#include <array>
#include <fstream>
#include <ostream>
```

Classes

class **KonfHash**

konf_hash.hpp

Go to the documentation of this file.¹

```
2 #ifndef KONFREADER_KONF_HASH_H
3 #define KONFREADER_KONF_HASH_H
4
5 #include <string>
6 #include <iostream>
7 #include <vector>
8 // #include <bits/stdc++.h>
9 #include <valarray>
10 #include <sstream>
11 #include <array>
12 #include <fstream>
13 #include <ostream>
14
15 #include "exceptions.h"
16
17 // #include "memtrace.h"
18
19 using namespace std;
20
21 /*TODO
22  * - hibakezeles
23  * - kivetelkezeles
24  * - kiirro fun irasa
25  */
26
27 class KonfHash
28 {
29     array<vector<pair<string, string>>, 256> table;
30
31     string getValue(const string &key);
32
33     static inline int hashFunction(const string &keyInString)
34     {
35         size_t sizeS = keyInString.size();
36         return (keyInString[0] + keyInString[1] + keyInString[sizeS - 1] +
37 keyInString[sizeS - 2]) % 256;
38     }
39
40     template<typename U>
41     U toType(const string &value);
42
43     int getPos(const string &key);
44
45     bool isMember(const string &key);
46
47 public:
48     KonfHash() = default;
49     KonfHash(const KonfHash &) = default;
50
51     void post(const pair<string, string>& element);
52
53     template<typename T>
54     T get(const string &key);
55
56     void put(const pair<string, string>& newElement);
57
58     void del(const string &key);
59
60     void readFile(const string &fileLocation);
61
62     void saveToFile(const string &name);
63
64     ~KonfHash();
65 };
66
67 // *****
68 *****
134
```



```

135
136 inline void KonfHash::post(const pair<string, string>& element)
137 {
138     if (!isMember(element.first))
139     {
140         int hashValue = hashFunction(element.first);
141         table[hashValue].push_back(element);
142     }
143     else
144     {
145         cerr << "The key is already used for another value!" << endl;
146     }
147
148 }
149
150
151
152 inline void KonfHash::put(const pair<string, string>& newElement)
153 {
154     int hashValue = hashFunction(newElement.first);
155
156     try
157     {
158         auto index = getPos(newElement.first);
159         table[hashValue][index] = newElement;
160     }
161     catch (runtime_error& except)
162     {
163         cerr << "There is no such element!" << endl;
164         return;
165     }
166 }
167
168
169 inline void KonfHash::del(const string &key)
170 {
171     int hashValue = hashFunction(key);
172
173     try
174     {
175         auto index = getPos(key);
176         table[hashValue].erase(table[hashValue].begin() + index);
177     }
178     catch (runtime_error& except)
179     {
180         cerr << "There is no such element!" << endl;
181         return;
182     }
183 }
184
185
186 inline string KonfHash::getValue(const string &key)
187 {
188     int hashedKey = hashFunction(key);
189
190     try
191     {
192         auto index = getPos(key);
193         return table[hashedKey][index].second;
194     }
195     catch (runtime_error& except)
196     {
197         return "";
198     }
199 }
200
201 template<typename U>
202 inline U KonfHash::toType(const string &value)
203 {
204     U convertedValue;
205
206     stringstream stream(value);
207     stream >> convertedValue;
208
209     if (stream.fail())
210     {
211         runtime_error e(value);

```

```

212         throw e;
213     }
214     return convertedValue;
215 }
216
217 template<typename T>
218 inline T KonfHash::get(const string &key)
219 {
220     return getValue(key);
221 }
222
223 template<>
224 inline string KonfHash::get(const string &key)
225 {
226     return getValue(key);
227 }
228
229 template<>
230 inline bool KonfHash::get(const string &key)
231 {
232     return getValue(key) == "true";
233 }
234
235 template<>
236 inline int KonfHash::get(const string &key)
237 {
238     try
239     {
240         auto value = toType<int>(getValue(key));
241         return value;
242     }
243     catch (runtime_error& except)
244     {
245         cerr << "Type mismatch or missing value!" << endl;
246         return -1;
247     }
248 }
249
250
251 template<>
252 inline double KonfHash::get(const string &key)
253 {
254     try
255     {
256         auto value = toType<double>(getValue(key));
257         return value;
258     }
259     catch (runtime_error& except)
260     {
261         cerr << "Type mismatch" << endl;
262         return -1.0;
263     }
264 }
265
266 inline void KonfHash::readFile(const string &fileLocation)
267 {
268     fstream file;
269     string line;
270
271     file.open(fileLocation, ios::in);
272
273     pair<string, string> tempP;
274     string t1, t2;
275
276     while (getline(file, line))
277     {
278         if (line[0] != '#')
279         {
280             t1 = line.substr(0, line.find('='));
281             t2 = line.substr(line.find('=') + 1);
282             tempP = make_pair(t1, t2);
283             post(tempP);
284         }
285     }
286
287     file.close();
288 }

```

```

289
290
291 inline int KonfHash::getPos(const string &key)
292 {
293     int hashedKey = hashFunction(key);
294     int pos = 0;
295
296     for (auto &i : table[hashedKey])
297     {
298         if (i.first == key)
299         {
300             return pos;
301         }
302         pos++;
303     }
304
305     runtime_error except(key);
306     throw except;
307 }
308
309 inline bool KonfHash::isMember(const string &key)
310 {
311     try
312     {
313         getPos(key);
314         return true;
315     }
316     catch (runtime_error& except)
317     {
318         return false;
319     }
320 }
321
322 inline void KonfHash::saveToFile(const string &name)
323 {
324     string fileName = "../" + name + ".txt";
325
326     ofstream saveFile;
327
328     saveFile.open(fileName);
329
330     for (auto &i : table)
331     {
332         for (auto &j : i)
333         {
334             saveFile << j.first << "=" << j.second << endl;
335         }
336     }
337 }
338
339 inline KonfHash::~KonfHash() = default;
340
341 #endif //KONFREADER_KONF_HASH_H

```

menu.cpp File Reference

```
#include <algorithm>
#include "menu.h"
```

Functions

- string **getFileLocation** ()
 - void **menuSwitch** (KonfHash konfObj)
-

Function Documentation

string getFileLocation ()

getFileLocation

Returns

- the location of the configuration file

void menuSwitch (KonfHash *konfObj*)

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

Parameters

<i>konfObj</i>	- a konfHash object
----------------	---------------------

menu.h File Reference

```
#include <iostream>
#include <fstream>
#include "konf_hash.hpp"
```

Functions

- string **getFileLocation** ()
 - void **menuSwitch** (KonfHash konfObj)
-

Function Documentation

string getFileLocation ()

getFileLocation

Returns

- the location of the configuration file

void menuSwitch (KonfHash *konfObj*)

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

Parameters

<i>konfObj</i>	- a konfHash object
----------------	---------------------

menu.h

```
Go to the documentation of this file.1 //
2 // Created by Tomi on 2023. 05. 28..
3 //
4
5 #ifndef KONFREADER_MENU_H
6 #define KONFREADER_MENU_H
7
8 #include <iostream>
9 #include <fstream>
10
11 #include "konf_hash.hpp"
12
13 using namespace std;
14
19 string getFileLocation();
20
26 void menuSwitch(KonfHash konfObj);
27
28 #endif //KONFREADER_MEN
```

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 felkinált 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.

KonfHash
-table[]: vector<pair<string, string>>
-getValue(key: const std::string &): std::string -<instance>:hashFunction(keyInString: const std::string &): int -toType(value: const std::string &)<type U>: U -getPos(key: const std::string&): int -isMember(key: const std::string &): int +KonfHash() +KonfHash(const KonfHash &) +post(element: const std::pair<std::string, std::string> &): void +get(key: const std::string &)<type T>: T +put(element: const std::pair<std::string, std::string> &): void +del(key: const std::string &): void +readFile(fileLocation: const std::string &): void +saveToFile(name: const std::string &): void +~KonfHash()