Programozás alapjai 1 Nyivántartás

Göbhardter Ádám
Péntek December 13 2019

Adatszerkezet-mutató

Adatszerkezetek

Az összes adatszerkezet listája rövid leírásokkal:

Fájlmutató

Fájllista

Az összes dokumentált fájl listája rövid leírásokkal:

Adatszerkezetek dokumentációja

DebugmallocData struktúrareferencia

Adatmezõk

- char **logfile** [256]
- DebugmallocElem **head** [debugmalloc_tablesize]
- DebugmallocElem **tail** [debugmalloc_tablesize]

Ez a dokumentáció a struktúráról a következõ fájl alapján készült:

- debugmalloc-impl.h

DebugmallocElem struktúrareferencia

Adatmezõk

- void * **real_mem**
- void * **user_mem**
- size_t **size**
- char **file** [64]
- unsigned **line**
- char **func** [32]
- char **expr** [128]
- struct DebugmallocElem * **prev**
- struct DebugmallocElem * **next**

Ez a dokumentáció a struktúráról a következõ fájl alapján készült:

- debugmalloc-impl.h

ListaElem struktúrareferencia

Adatmezõk

- Student **tanulo**
- struct ListaElem * **kov**

Ez a dokumentáció a struktúráról a következõ fájl alapján készült:

- progress.h

Student struktúrareferencia

Adatmezõk

- char **vez_nev** [50+1]
- char **ker_nev** [50+1]
- char **neptun** [6+1]
- char **eloadas_kod** [3+1]
- char **gyakorlat_kod** [3+1]
- char **gyakvez_vez_nev** [50+1]
- char **gyakvez_ker_nev** [50+1]
- int **hianyzas**
- int **nZH**
- int **kZH**
- char **NHF** [4+1]

Ez a dokumentáció a struktúráról a következõ fájl alapján készült:

- progress.h

Fájlok dokumentációja

main.c fájlreferencia

```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#include "debugmalloc.h"
#include "progress.h"
#include "progress_functions.h"
```

Függvények

- void menu (void)
- int choice (void)
- int main (void)

Függvények dokumentációja

int choice (void)

Addig kér be a felhasználótól egy számot, amíg az bele nem esik az 1-9 intervallumba, a határokat is beleértve.

Paraméterek

Nincs

Visszatérési érték

Egy egész szám, ami a menübõl választható opciók közül reprezentál egyet.

int main (void)

Ciklusban alkalmazza a menu, choice és progress függvényeket egészen addig, amíg a felhasználó ki nem választja a 11-es menüpontot. Ezután felszabadítja a megmaradt teljes láncolt listát.

Paraméterek

Nincs

Visszatérési érték

0, ami a program hiba nélküli lefutását jelenti.

void menu (void)

Megjeleníti a képernyõn a menürendszert.

Paraméterek

Nincs

Visszatérési érték

Nincs

progress.h fájlreferencia
(progress.c fájlban definiálva és inkludálva, progress_functions.c és main.c fájlban inkludálva)

Adatszerkezetek

- struct Student
- struct ListaElem

Típusdefiníciók

- typedef struct Student **Student**
- typedef struct ListaElem **ListaElem**

Függvények

- ListaElem * progress (int valasztas, ListaElem *eleje)

Részletes leírás

A függvények által használt láncolt lista deklarálása (ListaElem), A tanulók adatait összefogó struktúra deklarálása (Student). A progress függvény deklarálása.

Függvények dokumentációja

ListaElem* progress (int *valasztas*, ListaElem * *eleje*)

A menü egy opcióját hajtja végre a felhasználó választása alapján.

Paraméterek

Egy választott menüpont (egész szám) és az adatokat tároló láncolt lista elejére mutató pointer.

Visszatérési érték

Láncolt lista.

progress_functions.h fájlreferencia
(progress_functions.c fájlban definiálva és inkludálva, progress.c és main.c fájlban inkludálva)

Függvények

- ListaElem * read_from_file (void)
- ListaElem * list_new (void)
- ListaElem * list_add (ListaElem *eleje)
- ListaElem * list_free (ListaElem *eleje)
- void list_save (ListaElem *eleje)
- void list_print (ListaElem *eleje)
- ListaElem * del_eleje (ListaElem *eleje)
- ListaElem * del_from_list (ListaElem *eleje, char *neptun)
- int no_of_nodes (ListaElem *eleje)
- ListaElem * list_node_swap (ListaElem *node1, ListaElem *node2)
- void list_sort_ABC_Nev (ListaElem **eleje, int hossz)
- void nZH_potlas (ListaElem *eleje)
- void NHF_potlas (ListaElem *eleje)
- void list_sort_decrease_nZH (ListaElem **eleje, int hossz)
- void nZH_eredmenyek (ListaElem *eleje)
- void list_sort_decrease_Hianyzas (ListaElem **eleje, int hossz)
- void hianyzas_rangsor (ListaElem *eleje)
- void database_content (void)

Részletes leírás

A menürendszer opcióit végrehajtó függvények deklarálása.

Függvények dokumentációja

void database_content (void)

Beolvassa a database.txt fájl teljes tartalmát.

Paraméterek

Nincs.

Visszatérési érték

Nincs.

ListaElem* del_eleje (ListaElem * *eleje*)

Kitörli a láncolt lista legelsõ elemét.

Paraméterek

A láncolt lista elejére mutató pointer.

Visszatérési érték

A megmaradt láncolt lista elejére mutató pointer.

ListaElem* del_from_list (ListaElem * *eleje*, char * *neptun*)

A felhasználó által beírt Neptun kód alapján megkeres egy tanulót a láncolt listában, majd kitörli belõle minden, az adott tanulóhoz tartozó adattal együtt.

Paraméterek

A láncolt lista elejére mutató pointer és egy char típusú Neptun kód.

Visszatérési érték

A láncolt lista elejére mutató ponter vagy NULL pointer, ha a paraméterként megadott pointer NULL pointer.

void hianyzas_rangsor (ListaElem * *eleje*)

Megjeleníti a képernyõn a, felhasználó által bevitt, tanulók neveit, illetve hiányzásainak számát.

Paraméterek

A láncolt lista elejére mutató pointer.

Visszatérési érték

Nincs.

ListaElem* list_add (ListaElem * *eleje*)

Az eddigi láncolt lista végéhez hozzáfûz egy új elemet.

Paraméterek

A láncolt lista elejére mutató pointer.

Visszatérési érték

A létrejövõ, új láncolt lista elejére mutató pointer.

ListaElem* list_free (ListaElem * *eleje*)

Felszabadítja a láncolt lista által foglalt memóriaterületet.

Paraméterek

A láncolt lista elejére mutató pointer.

Visszatérési érték

NULL pointer.

ListaElem* list_new (void)

Beolvassa egy tanuló adatait, és egy új láncolt lista-elemet hoz létre belõlük.

Paraméterek

Nincs

Visszatérési érték

A létrehozott új láncolt lista-elemre mutató pointer vagy NULL pointer, ha a dinamikus memóriafoglalás sikertelen volt.

ListaElem* list_node_swap (ListaElem * *node1*, ListaElem * *node2*)

Megcseréli két láncolt lista-elem teljes tartalmát.

Paraméterek

A két megcserélendõ láncolt lista-elem.

Visszatérési érték

A két megcserélt elem közül a csere után elsõ helyre kerülõ elemre mutató ponter.

void list_print (ListaElem * *eleje*)

Kiírja a képernyõre a teljes láncolt lista tartalmát.

Paraméterek

A láncolt lista elejére mutató pointer.

Visszatérési érték

Nincs.

void list_save (ListaElem * *eleje*)

Létrehoz egy database.txt nevû fájlt, majd abba kiírja a teljes láncolt lista tartalmát.

Paraméterek

A láncolt lista elejére mutató pointer.

Visszatérési érték

Nincs.

void list_sort_ABC_Nev (ListaElem ** *eleje*, int *hossz*)

A láncolt lista elemeit a felhasználó által bevitt vezetéknevek alapján abc-sorrendbe rendezi. Forrás: StackOverflow.

Paraméterek

A láncolt lista elejére mutató pointerre mutató pointer.

Visszatérési érték

Nincs.

void list_sort_decrease_Hianyzas (ListaElem ** *eleje*, int *hossz*)

A láncolt lista elemeit a felhasználó által bevitt hiányzások száma alapján csökkenõ sorrendbe rendezi. Forrás: StackOverflow.

Paraméterek

A láncolt lista elejére muató pointerre mutató pointer.

Visszatérési érték

Nincs.

void list_sort_decrease_nZH (ListaElem ** *eleje*, int *hossz*)

A láncolt lista elemeit a felhasználó által bevitt nZH eredmények alapján alapján csökkenõ sorrendbe rendezi. Forrás: StackOverflow.

Paraméterek

A láncolt lista elejére mutató pointerre mutató pointer és egy egész szám, ami a láncolt lista hosszát reprezentálja.

Visszatérési érték

Nincs.

void NHF_potlas (ListaElem * *eleje*)

Megjeleníti a képernyõn azoknak a, felhasználó által bevitt, tanulóknak a nevét, akiknek nem fogadták el a NHF-ukat (ez is egy, a felhasználó által elõzetesen bevitt adat).

Paraméterek

A láncolt lista elejére mutató pointer.

Visszatérési érték

Nincs.

int no_of_nodes (ListaElem * *eleje*)

Megszámolja, hogy hány elemből áll a láncolt lista.

Paraméterek

A láncolt lista elejére mutató pointer.

Visszatérési érték

Egy egész szám, ami a láncolt lista hosszát reprezentálja.

void nZH_eredmenyek (ListaElem * *eleje*)

Megjeleníti a képernyõn a, felhasználó által bevitt, tanulók neveit, illetve nZH eredményét.

Paraméterek

A láncolt lista elejére mutató pointer.

Visszatérési érték

Nincs.

void nZH_potlas (ListaElem * *eleje*)

Megjeleníti a képernyõn azoknak a, felhasználó által bevitt, tanulóknak a nevét, akik egy bizonyos értéknél kevesebbet értek el a nZH-k során (ez is egy, a felhasználó által elõzetesen bevitt adat).

Paraméterek

A láncolt lista elejére mutató pointer.

Visszatérési érték

Nincs.

ListaElem* read_from_file (void)

Beolvassa a database.txt fájl tartalmát egy láncolt listába.

Paraméterek

Nincs

Visszatérési érték

A létrehozott új láncolt lista-elemre mutató pointer vagy NULL pointer, ha a fájl megnyitása / a dinamikus memóriafoglalás sikertelen volt.

- A program futtatásához tetszőleges fejlesztői környezet használható, amely képes kezelni a szabványos C programozási nyelvet (CodeBlocks IDE-ben írva és tesztelve).

- A program, futási időben és azon kívül, külső könyvtárat nem használ.

- A forráskódból a ,,Debugmalloc” eszköz opcionálisan eltávolítható, a program viselkedését nem fogja befolyásolni , csupán a tesztelést könnyíti meg.