

Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki

Programowanie Komputerów 2

Wisielec

autor	Tomasz Pelan
prowadzący	dr inż. Bożena Wieczorek
rok akademicki	2019/2020
kierunek	informatyka
rodzaj studiów	SSI
semestr	2
termin laboratorium	środa parzystą, 12:00 – 13:30
sekcja	72
termin oddania sprawozdania	2020-06-05

1 Treść zadania

Napisać grę w wisielca z różnymi kategoriami haseł. Kategorie oraz hasła mają być zawarte w pliku tekstowym. Wymagane jest stosowanie dynamicznych struktur danych.

2 Analiza zadania

Zagadnienie przedstawia problem napisania gry w wisielca.

2.1 Struktury danych

W programie wykorzystano listę jednokierunkową list jednokierunkowych. Główna lista zawiera kategorie haseł. Od każdej kategorii wychodzi lista jednokierunkowa przechowywująca hasła dla danej kategorii. Taka struktura danych pozwala na łatwe uzupełnianie programu o nowe kategorie i hasła, co sprawia że program jest bardzo elastyczny i łatwy w rozbudowie.

Operacje dodawania kategorii i haseł do list oraz usuwanie list odbywa się w sposób iteracyjny.

2.2 Algorytmy

Program dodaje z pliku kolejne kategorie i hasła, najpierw dodając kategorię do głównej listy, a następnie dodając po kolei hasła przydzielone do danej kategorii.

3 Specyfikacja zewnętrzna

Program jest uruchamiany jako plik wykonywalny (.exe). Program oparty jest na pliku zawierającym kategorie i hasła o nazwie `plik.txt`.

Gdyby jednak coś poszło nie tak i plik nie zostałby odnaleziony, wyświetli się komunikat

```
Nie udało sie poprawnie otworzyc pliku.
```

Program jest również odporny na wszelkie pomyłki użytkownika w wprowadzaniu danych, przykładowo jeśli użytkownik miał wybrać kategorię od 1-7 , a wpisanie zostana litera lub cyfra spoza tego przedziału, wyświetlony zostanie komunikat

```
Nieprawidlowe dane.Wybierz numer kategorii z listy.
```

Ponadto jeśli użytkownik chciałby zmienić kategorie haseł oraz hasła, w folderze gry umieszczony został folder o nazwie

`INSTRUKCJA_OBSŁUGI.txt`

w którym znajduje się szczegółowa instrukcja jak to zrobić.

4 Specyfikacja wewnętrzna

Program został zrealizowany zgodnie z paradygmatem strukturalnym. W programie wydzielono funkcje odpowiedzialne za interfejs, operacje związane ze strukturą danych oraz funkcjami odpowiadającymi za przeprowadzanie właściwej rozgrywki.

4.1 Ogólna struktura programu

W funkcji głównej wywołana jest funkcja `odczytaj_z_pliku`. Funkcja ta sprawdza czy plik dostał poprawnie odczytany a dane poprawnie wczytane do struktury danych. Gdy program napotkałby błąd na tym etapie, wyświetlony zostanie stosowny komunikat i program się zakończy.

Następnie wywoływana jest funkcja `menu_glowne`. Funkcja ta wyświetla interfejs użytkownikowi, który ma wybrać tryb gry.

Jeżeli użytkownik wybierze tryb gry z losową kategorią, wywołana zostanie funkcja `gra_losowa`, która inicjalizuje grę losową (wybiera losową kategorię oraz losowe hasło z tej kategorii).

Jeżeli użytkownik wybierze tryb gry z wyborem kategorii, wywołana zostanie funkcja `wypisz_kategorie`, która spowoduje wyświetlenie pełnej listy kategorii haseł. Po wybraniu jednej z kategorii wywołana zostanie funkcja `gra_z_wyborem_kategorii`, która w przeciwieństwie do funkcji `gra_losowa`, pomija losowanie kategorii (gdyż jest ona wybierana przez użytkownika).

Następnie w zależności od wybranego trybu funkcja `gra_losowa` lub funkcja `gra_z_wyborem_kategorii` wywołują funkcję `graj`, która w argumencie dostaje hasło i przeprowadza rzeczywistą grę w wisielca. Sama rozgrywka przebiega w całości w funkcji `graj`, jedynie po wybraniu litery wywoływana jest funkcja `dobierz_rysunek`, która wybiera odpowiedni rysunek wisielca do stanu gry.

Po zakończeniu gry, użytkownik za pomocą funkcji `menu_po_grze` może wybrać czy chce zagrać ponownie, czy może chce już zakończyć.

Wybranie pierwszej opcji spowoduje powrót programu do momentu w którym użytkownik wybiera tryb gry.

Wybranie opcji drugiej spowoduje wywołanie funkcji `usun_liste` i zakończy działanie programu.

4.2 Szczegółowy opis typów i funkcji

Szczegółowy opis typów i funkcji zawarty na końcu sprawozdania.

5 Testowanie

Program został przetestowany na różnego rodzaju plikach tekstowych. Pliki niepoprawne powodują zakończenie działania programu oraz wyświetlenia komunikatu o błędzie. Maksymalna wielkość pliku z kategoriami i hasłami jest uzależniona od specyfikacji komputera na jakim egzekwowany jest program.

Program został sprawdzony pod kątem wycieków pamięci.

6 Wnioski

Program imitujący grę w wisielca jest programem prostym, ale wymaga samodzielnego i bardzo uważnego zarządzania pamięcią. Najbardziej wymagające wydawały mi się operacje związane ze strukturą danych. Wymyślenie odpowiedniej struktury zajęło mi dość dużo czasu. Samo dodawanie kategorii i haseł z plików do listy również nie należało do najprostszych.

Problematyczne okazało się również ponowne używanie funkcji `scanf`, która przy wpisywaniu liter zamiast cyfry powodowała pętle nieskończoną. Po kilku godzinach poszukiwań pomocna okazała się wtedy funkcja `rewind` rozwiązująca ten problem. Przydatne okazały się również zdalne laboratoria, które pozwoliły na oswojenie się z językiem C, który moim zdaniem nie należy do najprostszych.

Myślę jednak, że największym mankamentem podczas pisania całego projektu była kwarantanna, która uniemożliwiła odbycie się zajęć i konsultacji w warunkach szkolnych.

Dodatek

Szczegółowy opis typów i funkcji

Wisielec

Generated by Doxygen 1.8.16

1 Data Structure Index	1
1.1 Data Structures	1
2 File Index	3
2.1 File List	3
3 Data Structure Documentation	5
3.1 hasla Struct Reference	5
3.1.1 Detailed Description	5
3.2 kategorie Struct Reference	5
3.2.1 Detailed Description	6
4 File Documentation	7
4.1 C:/Users/tomas/Desktop/Wisielec_/Wisielec_/funkcje.h File Reference	7
4.1.1 Function Documentation	7
4.1.1.1 dobierz_rysunek()	8
4.1.1.2 dodaj_do_listy()	8
4.1.1.3 dodaj_haslo()	8
4.1.1.4 gra_losowa()	8
4.1.1.5 gra_z_wyborem_kategorii()	9
4.1.1.6 graj()	9
4.1.1.7 menu_glowne()	9
4.1.1.8 menu_po_grze()	10
4.1.1.9 odczytaj_z_pliku()	10
4.1.1.10 rysunek1()	10
4.1.1.11 rysunek10()	10
4.1.1.12 rysunek2()	10
4.1.1.13 rysunek3()	10
4.1.1.14 rysunek4()	11
4.1.1.15 rysunek5()	11
4.1.1.16 rysunek6()	11
4.1.1.17 rysunek7()	11
4.1.1.18 rysunek8()	11
4.1.1.19 rysunek9()	11
4.1.1.20 usun_liste()	11
4.1.1.21 wypisz_kategorie()	12
4.2 C:/Users/tomas/Desktop/Wisielec_/Wisielec_/struktury.h File Reference	12
Index	13

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

hasla	5
kategorie	5

Chapter 2

File Index

2.1 File List

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

C:/Users/tomas/Desktop/Wisielec_/Wisielec_/funkcje.h	7
C:/Users/tomas/Desktop/Wisielec_/Wisielec_/struktury.h	12

Chapter 3

Data Structure Documentation

3.1 hasla Struct Reference

```
#include <struktury.h>
```

Data Fields

- char * [haslo](#)
haslo do gry
- struct [hasla](#) * [pNext](#)
wskaznik na nastepny element listy

3.1.1 Detailed Description

Struktura listy jednokierunkowej z haslami dla danej kategorii

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

- C:/Users/tomas/Desktop/Wisielec_/Wisielec_/ [struktury.h](#)

3.2 kategorie Struct Reference

```
#include <struktury.h>
```

Data Fields

- char * [kategoria](#)
kategoria do gry
- int [licznik_hasel](#)
licznik wszystkich hasel
- struct [kategorie](#) * [pNext](#)
wskaznik na nastepny element listy
- struct [hasla](#) * [pHasla](#)
wskaznik na pierwszy element listy hasel dla danej kategorii

3.2.1 Detailed Description

Struktura listy jednokierunkowej z kategoriami

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

- C:/Users/tomas/Desktop/Wisielec_/Wisielec_/struktury.h

Chapter 4

File Documentation

4.1 C:/Users/tomas/Desktop/Wisielec_/Wisielec_/funkcje.h File Reference

```
#include "struktury.h"
#include <stdio.h>
#include <stdlib.h>
```

Functions

- int [odczytaj_z_pliku](#) (struct [kategorie](#) **pHead, int *licznik_kategorii)
- void [dodaj_do_listy](#) (struct [kategorie](#) **pHead, char *slowo, FILE *plik)
- void [dodaj_haslo](#) (struct [hasla](#) **pHead, char *bufor)
- void [wypisz_kategorie](#) (struct [kategorie](#) *pHead, int licznik_kategorii)
- void [usun_liste](#) (struct [kategorie](#) **pHead)
- void [gra_losowa](#) (struct [hasla](#) *pHead, int licznik_kategorii)
- void [gra_z_wyborem_kategorii](#) (struct [hasla](#) *pHead, int kategoria)
- void [graj](#) (char *haslo, char *kategoria)
- void [rysunek1](#) ()
- void [rysunek2](#) ()
- void [rysunek3](#) ()
- void [rysunek4](#) ()
- void [rysunek5](#) ()
- void [rysunek6](#) ()
- void [rysunek7](#) ()
- void [rysunek8](#) ()
- void [rysunek9](#) ()
- void [rysunek10](#) ()
- void [dobierz_rysunek](#) (int zycia)
- void [menu_glowne](#) ()
- void [menu_po_grze](#) ()

4.1.1 Function Documentation

4.1.1.1 dobierz_rysunek()

```
void dobierz_rysunek (
    int zycia )
```

Funkcja dobiera odpowiedni rysunek wisielca

Parameters

<i>zycia</i>	jest to ilosc zyc uzytkownika w danym momencie gry
--------------	--

4.1.1.2 dodaj_do_listy()

```
void dodaj_do_listy (
    struct kategorie ** pHead,
    char * slowo,
    FILE * plik )
```

Funkcja dodaje kategorie do listy kategorii

Parameters

<i>plik</i>	jest to wskaznik na strumien plikowy z haslami i kategoriami
<i>slowo</i>	nowo odczytany wyraz , nazwa nowej (dodawanej) kategorii
<i>[int,out]</i>	pHead wskaznik na pierwszy element listy kategorii

4.1.1.3 dodaj_haslo()

```
void dodaj_haslo (
    struct hasla ** pHead,
    char * bufor )
```

Funkcja dodaje haslo do listy hasel

Parameters

<i>bufor</i>	nazwa hasla dodawanego
<i>[int,out]</i>	pHead wskaznik na pierwszy element listy hasel

4.1.1.4 gra_losowa()

```
void gra_losowa (
```

```
struct hasla * pHead,  
int licznik_kategorii )
```

Funkcja inicjalizuje gre losowa

Parameters

<i>licznik_kategorii</i>	jest to licznik wszystkich kategorii
<i>[int,out]</i>	pHead wskaznik na pierwszy element listy kategorii

4.1.1.5 gra_z_wyborem_kategorii()

```
void gra_z_wyborem_kategorii (  
    struct hasla * pHead,  
    int kategoria )
```

Funkcja inicjalizuje gre z wyborem kategorii

Parameters

<i>kategoria</i>	numer kategorii ktory ma byc wybrany do gry
<i>[int,out]</i>	pHead wskaznik na pierwszy element listy kategorii

4.1.1.6 graj()

```
void graj (  
    char * haslo,  
    char * kategoria )
```

Funkcja odpowiada za przeprowadzenie samej gry

Parameters

<i>haslo</i>	jest to haslo ktore uzytkownik musi zgadnac
<i>kategoria</i>	jest to kategoria hasla

4.1.1.7 menu_glowne()

```
void menu_glowne ( )
```

Wysietla menu glowne

4.1.1.8 menu_po_grze()

```
void menu_po_grze ( )
```

Wyswietla menu po grze

4.1.1.9 odczytaj_z_pliku()

```
int odczytaj_z_pliku (
    struct kategorie ** pHead,
    int * licznik_kategorii )
```

Funkcja odczytuje z pliku hasla i kategorie

Parameters

<i>licznik_kategorii</i>	jest to licznik wszystkich kategorii
[int,out]	pHead wskaznik na pierwszy element listy kategorii

Returns

informacja czy plik zostal poprawnie odczytany

4.1.1.10 rysunek1()

```
void rysunek1 ( )
```

Rysunek wisielca nr 1

4.1.1.11 rysunek10()

```
void rysunek10 ( )
```

Rysunek wisielca nr 10

4.1.1.12 rysunek2()

```
void rysunek2 ( )
```

Rysunek wisielca nr 2

4.1.1.13 rysunek3()

```
void rysunek3 ( )
```

Rysunek wisielca nr 3

4.1.1.14 rysunek4()

```
void rysunek4 ( )
```

Rysunek wisielca nr 4

4.1.1.15 rysunek5()

```
void rysunek5 ( )
```

Rysunek wisielca nr 5

4.1.1.16 rysunek6()

```
void rysunek6 ( )
```

Rysunek wisielca nr 6

4.1.1.17 rysunek7()

```
void rysunek7 ( )
```

Rysunek wisielca nr 7

4.1.1.18 rysunek8()

```
void rysunek8 ( )
```

Rysunek wisielca nr 8

4.1.1.19 rysunek9()

```
void rysunek9 ( )
```

Rysunek wisielca nr 9

4.1.1.20 usun_liste()

```
void usun_liste (
    struct katégorie ** pHead )
```

Funkcja inicjalizuje gre losowa

Parameters

<i>[int,out]</i>	pHead wskaźnik na pierwszy element listy kategorii
------------------	--

4.1.1.21 wypisz_kategorie()

```
void wypisz_kategorie (
    struct kategorie * pHead,
    int licznik_kategorii )
```

Funkcja inicjalizuje gre losowa

Parameters

<i>licznik_kategorii</i>	jest to licznik wszystkich kategorii
<i>[int,out]</i>	pHead wskaznik na pierwszy element listy kategorii

4.2 C:/Users/tomas/Desktop/Wisielec_/Wisielec_/struktury.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
```

Data Structures

- struct [hasla](#)
- struct [kategorie](#)

Index

C:/Users/tomas/Desktop/Wisielec_/Wisielec_/funkcje.h, [7](#)
C:/Users/tomas/Desktop/Wisielec_/Wisielec_/struktury.h, [12](#)

dobierz_rysunek
 funkcje.h, [7](#)
dodaj_do_listy
 funkcje.h, [8](#)
dodaj_haslo
 funkcje.h, [8](#)

funkcje.h
 dobierz_rysunek, [7](#)
 dodaj_do_listy, [8](#)
 dodaj_haslo, [8](#)
 gra_losowa, [8](#)
 gra_z_wyborem_kategorii, [9](#)
 graj, [9](#)
 menu_glowne, [9](#)
 menu_po_grze, [9](#)
 odczytaj_z_pliku, [10](#)
 rysunek1, [10](#)
 rysunek10, [10](#)
 rysunek2, [10](#)
 rysunek3, [10](#)
 rysunek4, [10](#)
 rysunek5, [11](#)
 rysunek6, [11](#)
 rysunek7, [11](#)
 rysunek8, [11](#)
 rysunek9, [11](#)
 usun_liste, [11](#)
 wypisz_kategorie, [12](#)

gra_losowa
 funkcje.h, [8](#)
gra_z_wyborem_kategorii
 funkcje.h, [9](#)
graj
 funkcje.h, [9](#)

hasla, [5](#)

kategorie, [5](#)

menu_glowne
 funkcje.h, [9](#)
menu_po_grze
 funkcje.h, [9](#)

odczytaj_z_pliku
 funkcje.h, [10](#)

rysunek1
 funkcje.h, [10](#)
rysunek10
 funkcje.h, [10](#)
rysunek2
 funkcje.h, [10](#)
rysunek3
 funkcje.h, [10](#)
rysunek4
 funkcje.h, [10](#)
rysunek5
 funkcje.h, [11](#)
rysunek6
 funkcje.h, [11](#)
rysunek7
 funkcje.h, [11](#)
rysunek8
 funkcje.h, [11](#)
rysunek9
 funkcje.h, [11](#)

usun_liste
 funkcje.h, [11](#)

wypisz_kategorie
 funkcje.h, [12](#)