

Politechnika Śląska
Wydział Informatyki, Elektroniki i Informatyki

Programowanie Komputerów

Rezerwacje stolików

autor Jakub Polczyk

prowadzący mgr inż. Grzegorz Kwiatkowski

rok akademicki 2019/2020

kierunek informatyka

rodzaj studiów SSI

semestr 2

termin laboratorium piątek, 12:00 – 13:30

sekcja 2.2

termin oddania sprawozdania 2020-09-08

Link GitHub: <https://github.com/Kubson900/polsl-aei-pk2-projekt-rezerwacje>

1. Analiza zadania

Zagadnienie przedstawia problem operacji na liście oraz operacjach na plikach tekstowych obejmujących odczyt z pliku tekstowego oraz zapis do pliku tekstowego.

1.1 Struktury danych

W programie wykorzystano listę w liście. W pierwszej liście przechowywane są stoliki. Każdy stół ma swoją listę klientów. Taka struktura danych umożliwia łatwe uporządkowanie danych z pliku wejściowego.

1.2 Kluczowe funkcje

1. AddTable/AddClient, są to najważniejsze funkcje, która są odpowiedzialne za dodawanie odpowiednio stolika/klienta do posortowanej listy.
2. Check, funkcja ta pobiera dane od użytkownika i sprawdza czy rezerwacja może zostać dokonana.

2. Specyfikacja zewnętrzna

Program uruchomiony jest przez lokalny Debugger w programie Visual Studio. Po uruchomieniu program prosi użytkownika o wybranie odpowiedniej funkcji programu.

3. Specyfikacja wewnętrzna

Program został zrealizowany zgodnie z paradygmatem strukturalnym. Program opiera się głównie na komunikacji z użytkownikiem. Osobno działającym działem jest fragment kodu odpowiedzialny za logikę aplikacji (działania na plikach tekstowych).

4. Testowanie

Program został przetestowany na różnego rodzaju plikach. Plik pusty nie powoduje zgłoszenia błędu oraz wyświetlenia odpowiedniego komunikatu. Program został sprawdzony pod kątem wycieków pamięci.

5. Instrukcja obsługi

1. Wybór funkcji programu liczbami od 1 do 3

```
WITAMY W NASZEJ RESTAURACJI!  
Godziny otwarcia 6:00 - 22:00  
Stoliki do 8 osob  
  
[1] Zarezerwuj stolik  
[2] Wswietl wszystkich gosci  
[3] Zakoncz  
Wybierz numer:
```

2. Po wybraniu numeru 1 należy odpowiednio podać kolejno ilość osób, godzinę, nazwisko oraz zniżkę, jaką należy naliczyć. Pomyślne wprowadzenie danych powróci do menu.

```
Na ile osob: 2  
Na ktora godzine: 6  
Na jakie nazwisko: Max  
Jaka znizke naliczyc: 10
```

3. Restauracja posiada stoliki do 8 osób stąd nie ma możliwości rezerwacji na większą ilość osób

```
Na ile osob: 9  
Nie mozna zarezerwowac na tyle osob
```

4. Jeżeli stolik na daną godzinę jest już zajęty bądź nie mieści się w godzinach otwarcia restauracji pojawi się komunikat

```
Na ile osob: 2  
Na ktora godzine: 8  
Nie mozna zarezerwowac na ta godzine
```

5. Wybranie numeru 2 wyświetli listę wszystkich rezerwacji w restauracji

```
Table number: 2  
  
Surname: Smith  
Hour: 8  
Discount: 10%  
Number of people: 2  
  
Surname: Kowalski  
Hour: 10  
Discount: 15%  
Number of people: 1
```

6. Wybranie opcji numer 3 kończy działanie programu

Zakończono działanie programu

6. Wnioski

Program **Rezerwacje** nie jest skomplikowanym programem, chociaż wymaga samodzielnego zarządzania pamięcią. Najbardziej wymagające okazało się działanie na wskaźnikach oraz łańcuchach znakowych. Realizacja projektu nauczyła mnie jak dobrze posługiwać się wskaźnikami i jak operować na listach Jednokierunkowych.

Literatura

[1] Krzysztof Simiński. Wykłady z podstaw programowania komputerów.

My Project

Generated by Doxygen 1.8.16

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Client	??
Table	??

Chapter 2

File Index

2.1 File List

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

C:/Users/polcz/OneDrive/Pulpit/PK_Sprawozdanie/kod/ functions.h	??
C:/Users/polcz/OneDrive/Pulpit/PK_Sprawozdanie/kod/ structures.h	??

Chapter 3

Data Structure Documentation

3.1 Client Struct Reference

```
#include <structures.h>
```

Data Fields

- char * [surname](#)
- int [hour](#)
- int [number](#)
- int [discount](#)
- struct [Client](#) * [pNextClient](#)

3.1.1 Detailed Description

Struktura opisujaca klienta

3.1.2 Field Documentation

3.1.2.1 discount

```
int discount
```

Znizka dla klienta

3.1.2.2 hour

```
int hour
```

Godzina rezerwacji

3.1.2.3 number

```
int number
```

Ilosc osob

3.1.2.4 pNextClient

```
struct Client* pNextClient
```

Wskaznik na kolejny element listy

3.1.2.5 surname

```
char* surname
```

Nazwisko klienta

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

- C:/Users/polcz/OneDrive/Pulpit/PK_Sprawozdanie/kod/[structures.h](#)

3.2 Table Struct Reference

```
#include <structures.h>
```

Data Fields

- int [number](#)
- struct [Client](#) * [pHeadClient](#)
- struct [Table](#) * [pNextTable](#)

3.2.1 Detailed Description

Struktura opisujaca stolik

3.2.2 Field Documentation

3.2.2.1 number

```
int number
```

Numer stolika

3.2.2.2 pHeadClient

```
struct Client* pHeadClient
```

Wskaźnik na liście klientów

3.2.2.3 pNextTable

```
struct Table* pNextTable
```

Wskaźnik na kolejny element listy

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

- C:/Users/polcz/OneDrive/Pulpit/PK_Sprawozdanie/kod/[structures.h](#)

Chapter 4

File Documentation

4.1 C:/Users/polcz/OneDrive/Pulpit/PK_Sprawozdanie/kod/functions.h File Reference

```
#include "structures.h"
```

Functions

- `Table * AddTable (Table **pHead, int number)`
- `void AddClient (Client **pHead, const char *surname, int hour, int number, int discount)`
- `void ShowClient (Client *pHead)`
- `void ShowTable (Table *pHead)`
- `Table * CheckNumber (Table *pHead, int number)`
- `int CheckHour (Client *pHead, int hour)`
- `int ExtractNumber (char *nazwa)`
- `void Upload (Table **pHead, char *files[], int num)`
- `void Check (Table *pHead)`
- `void SaveToFile (Table *pHead, char *files[], int num)`
- `void DeleteClient (Client **pHead)`
- `void DeleteTable (Table **pHead)`

4.1.1 Function Documentation

4.1.1.1 AddClient()

```
void AddClient (  
    Client ** pHead,  
    const char * surname,  
    int hour,  
    int number,  
    int discount )
```

Funkcja dodaje klienta do konkretnego stolika

Parameters

<i>pHead</i>	Reprezentant listy klientow
<i>surname</i>	Nazwisko klienta
<i>hour</i>	Godzina rezerwacji
<i>number</i>	Ilosc osob
<i>discount</i>	Znizka dla klienta

4.1.1.2 AddTable()

```
Table* AddTable (
    Table ** pHead,
    int number )
```

Funkcja dodaje stoliki do posortowanej listy

Parameters

<i>pHead</i>	Reprezentant listy stolikow
<i>number</i>	Numer stolika i max ilosc osob

Returns

Wskaznik na stolik

4.1.1.3 Check()

```
void Check (
    Table * pHead )
```

Funkcja pobiera dane od uzytkownika i sprawdza czy moze utworzyc rezerwacje

Parameters

<i>pHead</i>	Reprezentant listy stolikow
--------------	-----------------------------

4.1.1.4 CheckHour()

```
int CheckHour (
    Client * pHead,
    int hour )
```

Funkcja sprawdza czy jest dostepna wskazana godzina

Parameters

<i>pHead</i>	Reprezentant listy klientow
<i>hour</i>	Godzina rezerwacji

Returns

Prawda lub falsz

4.1.1.5 CheckNumber()

```
Table* CheckNumber (
    Table * pHead,
    int number )
```

Funkcja sprawdza przy którym stoliku może usiąść klient

Parameters

<i>pHead</i>	Reprezentant listy stolikow
<i>number</i>	Ilość osób

Returns

Wskaźnik na stolik

4.1.1.6 DeleteClient()

```
void DeleteClient (
    Client ** pHead )
```

Funkcja usuwa listę klientów ze stolika

Parameters

<i>pHead</i>	Wskaźnik na reprezentanta listy klientów
--------------	--

4.1.1.7 DeleteTable()

```
void DeleteTable (
    Table ** pHead )
```

Funkcja usuwa liste stolikow

Parameters

<i>pHead</i>	Wskaznik na reprezentanta listy stolikow
--------------	--

4.1.1.8 ExtractNumber()

```
int ExtractNumber (  
    char * nazwa )
```

Funkcja odczytuje numer stolika z nazwy pliku

Parameters

<i>nazwa</i>	Nazwa pliku
--------------	-------------

Returns

Numer stolika

4.1.1.9 SaveToFile()

```
void SaveToFile (  
    Table * pHead,  
    char * files[],  
    int num )
```

Funkcja zapisuje do plikow tekstowych

Parameters

<i>pHead</i>	Reprezentant listy stolikow
<i>files</i>	Tablica z nazwami plikow
<i>num</i>	Ilosc plikow do wczytania

4.1.1.10 ShowClient()

```
void ShowClient (  
    Client * pHead )
```

Funkcja wyswietla liste klientow

Parameters

<i>pHead</i>	Reprezentant listy klientow
--------------	-----------------------------

4.1.1.11 ShowTable()

```
void ShowTable (
    Table * pHead )
```

Funkcja wyswietla wszystkie stoliki

Parameters

<i>pHead</i>	Reprezentant listy stolikow
--------------	-----------------------------

4.1.1.12 Upload()

```
void Upload (
    Table ** pHead,
    char * files[],
    int num )
```

Funkcja odczytuje z pliku i tworzy strukture dynamiczna (lista w liscie)

Parameters

<i>pHead</i>	Wskaźnik na reprezentanta listy przystankow
<i>files</i>	Tablica z nazwami plikow
<i>num</i>	Ilość plikow do wczytania

4.2 C:/Users/polcz/OneDrive/Pulpit/PK_Sprawozdanie/kod/structures.h

File Reference

Data Structures

- struct [Client](#)
- struct [Table](#)

Typedefs

- typedef struct [Client](#) [Client](#)
- typedef struct [Table](#) [Table](#)

4.2.1 Typedef Documentation

4.2.1.1 Client

```
typedef struct Client Client
```

Struktura opisująca klienta

4.2.1.2 Table

```
typedef struct Table Table
```

Struktura opisująca stolik

