Obsługa Rezerwacji Sali Kinowej Projekt PK3 - Michał Wieczorek

Wygenerowano przez Doxygen 1.9.3

Rozdział 1

Indeks hierarchiczny

1.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

Rozdział 2

Indeks klas

2.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

Rozdział 3

Indeks plików

3.1 Lista plików

Tutaj znajduje się lista wszystkich udokumentowanych plików z ich krótkimi opisami:

Rozdział 4

Dokumentacja klas

4.1 Dokumentacja klasy Data

Glowna klasa Data przechowujaca i obslugujaca wszystkie dane w bazie.

`#include <Data.h>`

Metody publiczne

- void **display_cinema** ()

  Publiczna metoda wyswietlajaca animacje wizualizujaca sale kinowa.
- void **display_sits_info** ()

  Publiczna metoda wyswietlajaca informacje dodycznaca dostepnosci miejsc (wolne/zajete) w sali kinowej.
- void **make_reservation** ()

  Publiczna metoda umozliwiajaca dokonanie nowej rezerwacji.
- void **choose_sits** (int &verse, int &column)

  Publiczna metoda sprawdzajaca czy wprowadzone dane (wybor rzedu i miejsca w sali) sa dostepne do rezerwacji.
- void **search_reservation** ()

  Publiczna metoda umozliwiajaca wyszukanie danej rezerwacji.
- void **display_reserwation** ()

  Publiczna metoda umozliwiajaca wyswietlenie wszystkich rezerwacji znajdujacych sie w bazie.
- void **cancel_reserwation** ()

  Publiczna metoda umozliwiajaca anulowanie konkretnej rezerwacji.
- void **sort_reservation** ()

  Publiczna metoda umozliwiajaca wyswietlenie roznego rodzaju sortowania zgromadzonych danych (sortowanie wedlug ilosci rezeracji lub wyswietlenie tylko rezerwacji o konkretnym typie tj. Normalny, Studencki, Ulgowy, Rodzinny).
- void **display_reservation_ranking** ()

  Publiczna metoda sortujaca i wyswietlajaca rezerwacje pod katem ilosci dodokanych rezerwacji.
- void **add_to_his** (const std::string &name, const std::string &surname, const double &price, const int &verse, const int &column, const int &type, const int &mode)

  Publiczna metoda zapisujaca dane dzialanie w bazie danych (dodanie/anulowanie rezerwacji) w historii, ktora moze zostac wydrukowana w raporcie tj. zewnetrznym pliku tekstowym.
- void **print_raport** ()

  Publiczna metoda zapisujaca historie rezerwacji oraz aktualne zyski w zewnetrznym pliku tekstowym.

Metody prywatne

- void **add_new_reservation** (const std::string &name, const std::string &surname, const int &verse, const int &column, const int &type)

  Prywatna metoda dodajaca nowa rezerwacje - dokonuje zmian w strukturach danych przechowujacych dane o rezerwacjach.

Atrybuty prywatne

- std::vector< std::vector< std::unique_ptr< Ticket > > > **all_tickets**

  Struktura danych odpowiedzialna za wyswietlanie animacji wizualizujaca sale kinowa.
- std::list< std::unique_ptr< Ticket > > **reservation**

  Struktura danych odpowiedzialna za obluge aktualnych danych dotyczacych rezerwacji.
- std::vector< His_obj > **history**

  Struktura danych odpowiedzialna za historie rezerwacji zapisywana w raporcie.

4.1.1 Opis szczegółowy

Glowna klasa Data przechowujaca i obslugujaca wszystkie dane w bazie.

Dokumentacja dla tej klasy została wygenerowana z plików:

- C:/ProjektPK3 - Kopia/ProjektPK3/Data.h
- C:/ProjektPK3 - Kopia/ProjektPK3/Data.cpp

4.2 Dokumentacja struktury His_obj

Struktura pomocnicza przechowujaca dane rezerwacji, ktore zostana zapisane w raporcie.

`#include <Data.h>`

Atrybuty publiczne

- std::string **name**
- std::string **surname**
- double **price**
- int **verse**
- int **column**
- int **type**
- int **mode**

4.2.1 Opis szczegółowy

Struktura pomocnicza przechowujaca dane rezerwacji, ktore zostana zapisane w raporcie.

Dokumentacja dla tej struktury została wygenerowana z pliku:

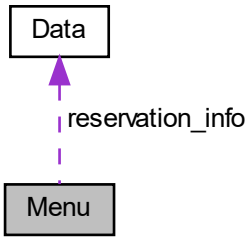
- C:/ProjektPK3 - Kopia/ProjektPK3/Data.h

4.3 Dokumentacja klasy Menu

Klasa glowna zarzadzajaca pozostalymi klasami programu (rdzen programu - zgodnie z wzorcem projektowym kompozyt).

`#include <Menu.h>`

Diagram współpracy dla Menu:



Metody publiczne

- void **display_opt** ()

  Publiczna metoda wyswietlajaca opcje w glownym menu programu.
- void **start_prog** ()

  Publiczna metoda wlaczajaca program - glowna petla programu.

Atrybuty prywatne

- Data **reservation_info**

4.3.1 Opis szczegółowy

Klasa glowna zarzadzajaca pozostalymi klasami programu (rdzen programu - zgodnie z wzorcem projektowym kompozyt).

Dokumentacja dla tej klasy została wygenerowana z plików:

- C:/ProjektPK3 - Kopia/ProjektPK3/Menu.h
- C:/ProjektPK3 - Kopia/ProjektPK3/Menu.cpp

4.4 Dokumentacja struktury Ranking

Struktura pomocnicza reprezentujaca liczbe danych rezerwacji na dane nazwisko - wykorzystanie w funkcjach sortujacych programu.

`#include <Data.h>`

Atrybuty publiczne

- std::string **name**
- std::string **surname**
- int **count**

4.4.1 Opis szczegółowy

Struktura pomocnicza reprezentujaca liczbe danych rezerwacji na dane nazwisko - wykorzystanie w funkcjach sortujacych programu.

Dokumentacja dla tej struktury została wygenerowana z pliku:

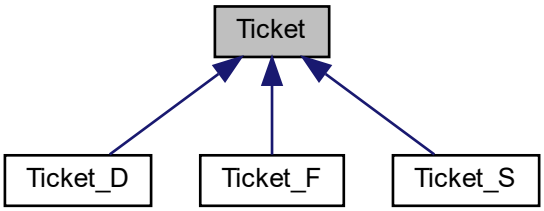
- C:/ProjektPK3 - Kopia/ProjektPK3/Data.h

4.5 Dokumentacja klasy Ticket

Klasa reprezentujaca rezerwacje normalne dziedziczaca po klasie Ticket.

`#include <Ticket.h>`

Diagram dziedziczenia dla Ticket



Metody publiczne

- **Ticket** (std::string name, std::string surname, double price, int verse, int column)
- **Ticket** (std::string name, std::string surname, int verse, int column)
- void **set_name** (const std::string &name)

  Publiczna metoda ustawiajaca informacje o imieniu na ktorym jest przypisana dana rezerwacja.
- void **set_surname** (const std::string &surname)

  Publiczna metoda ustawiajaca informacje o nazwisku na ktorym jest przypisana dana rezerwacja.
- void **set_price** (const double &price)

  Publiczna metoda ustawiajaca cene danego miejsca.
- std::string **get_name** ()

  Publiczna metoda zwracajaca informacje o imieniu na ktorym jest przypisana dana rezerwacja.

- std::string **get_surname** ()

  Publiczna metoda zwracajaca informacje o nazwisku na ktorym jest przypisana dana rezerwacja.
- double **get_price** ()

  Publiczna metoda zwracajaca informacje o cenie danej rezerwacji.
- bool **get_is_free** ()

  Publiczna metoda zwracajaca informacje czy dane miejsce jest wolne.
- int **get_verse** ()

  Publiczna metoda zwracajaca informacje o kolumnie w ktorej znajduje sie dane miejsce.
- int **get_column** ()

  Publiczna metoda zwracajaca informacje o kolumnie w ktorej znajduje sie dane miejsce.
- void **reserve** ()

  Publiczna metoda dokonujaca operacji rezerwacji miejsca - modyfikuje pole klasy is_free.
- void **unreserve** ()

  Publiczna metoda dokonujaca operacji anulowania rezerwacji miejsca - modyfikuje pole klasy is_free.
- virtual void display_sit ()

  Publiczna metoda wirtualna, przeslaniana przez klasy dziedziczne, wyswietlajaca odpowiednio pole w animacji wizualizujacej sale kinowa dla rezerwacji norlamnych.
- virtual void display_ticket_type ()

  Publiczna metoda wirtualna, przeslaniana przez klasy dziedziczne, wyswietlajaca informacje o typie rezerwacji.
- virtual int type ()

  Publiczna metoda wirtualna, przeslaniana przez klasy dziedziczne, zwracajaca informacje o typie obiektu (zwraca wartosc typu int).

Metody chronione

- **Ticket** (double price)

Atrybuty chronione

- std::string **name**
- std::string **surname**
- double **price**
- bool **is_free**
- int **verse**
- int **column**

4.5.1 Opis szczegółowy

Klasa reprezentujaca rezerwacje normalne dziedziczaca po klasie Ticket.

4.5.2 Dokumentacja funkcji składowych

4.5.2.1 display_sit()

```
void Ticket::display_sit ( )  [virtual]
```

Publiczna metoda wirtualna, przeslaniana przez klasy dziedziczne, wyswietlajaca odpowiednio pole w animacji wizualizujacej sale kinowa dla rezerwacji norlamnych.

Reimplementowana w Ticket_D, Ticket_F i Ticket_S.

4.5.2.2 display_ticket_type()

```
void Ticket::display_ticket_type ( )  [virtual]
```

Publiczna metoda wirtualna, przeslaniana przez klasy dziedziczne, wyswietlajaca informacje o typie rezerwacji.

Reimplementowana w Ticket_D, Ticket_F i Ticket_S.

4.5.2.3 type()

```
int Ticket::type ( )  [virtual]
```

Publiczna metoda wirtualna, przeslaniana przez klasy dziedziczne, zwracajaca informacje o typie obiektu (zwraca wartosc typu int).

Reimplementowana w Ticket_D, Ticket_F i Ticket_S.

Dokumentacja dla tej klasy została wygenerowana z plików:

- C:/ProjektPK3 - Kopia/ProjektPK3/Ticket.h
- C:/ProjektPK3 - Kopia/ProjektPK3/Ticket.cpp

4.6 Dokumentacja klasy Ticket_D

Klasa reprezentujaca rezerwacje ulgowe dziedziczaca po klasie Ticket.

```
#include <Ticket_D.h>
```

Diagram dziedziczenia dla Ticket_D

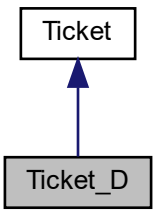
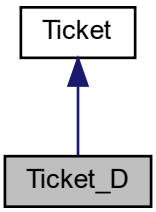


Diagram współpracy dla Ticket_D:



Metody publiczne

- **Ticket_D** (std::string name, std::string surname, double price, int verse, int column)
- virtual void display_sit ()

  Publiczna metoda wirtualna wyswietlajaca odpowiednio pole w animacji wizualizujacej sale kinowa dla rezerwacji ulgowych.
- virtual void display_ticket_type ()

  Publiczna metoda wirtualna wyswietlajaca informacje o typie rezerwacji.
- virtual int type ()

  Publiczna metoda wirtualna zwracajaca informacje o typie obiektu (zwraca wartosc typu int).

Dodatkowe Dziedziczone Składowe

4.6.1 Opis szczegółowy

Klasa reprezentujaca rezerwacje ulgowe dziedziczaca po klasie Ticket.

Copyright

Copyright 2023 Michal Wieczorek.

4.6.2 Dokumentacja funkcji składowych

4.6.2.1 display_sit()

```
void Ticket_D::display_sit ( )  [virtual]
```

Publiczna metoda wirtualna wyswietlajaca odpowiednio pole w animacji wizualizujacej sale kinowa dla rezerwacji ulgowych.

Reimplementowana z Ticket.

4.6.2.2 display_ticket_type()

```
void Ticket_D::display_ticket_type ( )  [virtual]
```

Publiczna metoda wirtualna wyswietlajaca informacje o typie rezerwacji.

Reimplementowana z Ticket.

4.6.2.3 type()

```
int Ticket_D::type ( )  [virtual]
```

Publiczna metoda wirtualna zwracajaca informacje o typie obiektu (zwraca wartosc typu int).

Reimplementowana z Ticket.

Dokumentacja dla tej klasy została wygenerowana z plików:

- C:/ProjektPK3 - Kopia/ProjektPK3/Ticket_D.h
- C:/ProjektPK3 - Kopia/ProjektPK3/Ticket_D.cpp

4.7 Dokumentacja klasy Ticket_F

Klasa reprezentujaca rezerwacje rodzinne dziedziczaca po klasie Ticket.

```
#include <Ticket_F.h>
```

Diagram dziedziczenia dla Ticket_F

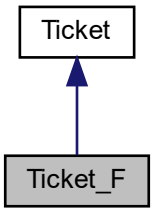
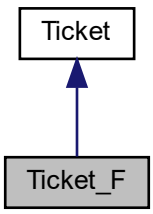


Diagram współpracy dla Ticket_F:



Metody publiczne

- **Ticket_F** (std::string name, std::string surname, double price, int verse, int column)
- virtual void display_sit ()

  Publiczna metoda wirtualna wyswietlajaca odpowiednio pole w animacji wizualizujacej sale kinowa dla rezerwacji rodzinnych.
- virtual void display_ticket_type ()

  Publiczna metoda wirtualna wyswietlajaca informacje o typie rezerwacji.
- virtual int type ()

  Publiczna metoda wirtualna zwracajaca informacje o typie obiektu (zwraca wartosc typu int).

Dodatkowe Dziedziczone Składowe

4.7.1 Opis szczegółowy

Klasa reprezentujaca rezerwacje rodzinne dziedziczaca po klasie Ticket.

Copyright

Copyright 2023 Michal Wieczorek.

4.7.2 Dokumentacja funkcji składowych

4.7.2.1 display_sit()

```
void Ticket_F::display_sit ( )  [virtual]
```

Publiczna metoda wirtualna wyswietlajaca odpowiednio pole w animacji wizualizujacej sale kinowa dla rezerwacji rodzinnych.

Reimplementowana z Ticket.

4.7.2.2 display_ticket_type()

```
void Ticket_F::display_ticket_type ( )  [virtual]
```

Publiczna metoda wirtualna wyswietlajaca informacje o typie rezerwacji.

Reimplementowana z Ticket.

4.7.2.3 type()

```
int Ticket_F::type ( )  [virtual]
```

Publiczna metoda wirtualna zwracajaca informacje o typie obiektu (zwraca wartosc typu int).

Reimplementowana z Ticket.

Dokumentacja dla tej klasy została wygenerowana z plików:

- C:/ProjektPK3 - Kopia/ProjektPK3/Ticket_F.h
- C:/ProjektPK3 - Kopia/ProjektPK3/Ticket_F.cpp

4.8 Dokumentacja klasy Ticket_S

Klasa reprezentujaca rezerwacje studenckie dziedziczaca po klasie Ticket.

```
#include <Ticket_S.h>
```

Diagram dziedziczenia dla Ticket_S

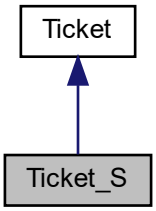
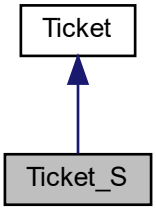


Diagram współpracy dla Ticket_S:



Metody publiczne

- **Ticket_S** (std::string name, std::string surname, double price, int verse, int column)
- virtual void display_sit ()

  Publiczna metoda wirtualna wyswietlajaca odpowiednio pole w animacji wizualizujacej sale kinowa dla rezerwacji studenckich.
- virtual void display_ticket_type ()

  Publiczna metoda wirtualna wyswietlajaca informacje o typie rezerwacji.
- virtual int type ()

  Publiczna metoda wirtualna zwracajaca informacje o typie obiektu (zwraca wartosc typu int).

Dodatkowe Dziedziczone Składowe

4.8.1 Opis szczegółowy

Klasa reprezentujaca rezerwacje studenckie dziedziczaca po klasie Ticket.

Copyright

Copyright 2023 Michal Wieczorek.

4.8.2 Dokumentacja funkcji składowych

4.8.2.1 display_sit()

```
void Ticket_S::display_sit ( )  [virtual]
```

Publiczna metoda wirtualna wyswietlajaca odpowiednio pole w animacji wizualizujacej sale kinowa dla rezerwacji studenckich.

Reimplementowana z Ticket.

4.8.2.2 display_ticket_type()

```
void Ticket_S::display_ticket_type ( )  [virtual]
```

Publiczna metoda wirtualna wyswietlajaca informacje o typie rezerwacji.

Reimplementowana z Ticket.

4.8.2.3 type()

```
int Ticket_S::type ( )  [virtual]
```

Publiczna metoda wirtualna zwracajaca informacje o typie obiektu (zwraca wartosc typu int).

Reimplementowana z Ticket.

Dokumentacja dla tej klasy została wygenerowana z plików:

- C:/ProjektPK3 - Kopia/ProjektPK3/Ticket_S.h
- C:/ProjektPK3 - Kopia/ProjektPK3/Ticket_S.cpp

Rozdział 5

Dokumentacja plików

5.1 Data.h

```
1 #pragma once
2 #include "Ticket.h"
3 #include "Ticket_S.h"
4 #include "Ticket_D.h"
5 #include "Ticket_F.h"
6 #include "Ticket.h"
7 #include <vector>
8 #include <memory>
9 #include <list>
10 #include <algorithm>
11 #include <fstream>
12
18 enum {
19     Normal = 1,
20     Student = 2,
21     Discount = 3,
22     Family = 4
23 };
24
25 enum {
26     Add = 1,
27     Cancel = 2
28 };
29
33 struct Ranking {
34     std::string name;
35     std::string surname;
36     int count;
37 };
38
42 struct His_obj {
43     std::string name;
44     std::string surname;
45     double price;
46     int verse, column, type, mode;
47 };
48
52 bool comp_ranking_count(Ranking obj1, Ranking obj2);
53 std::string give_type(His_obj obj);
54 std::string give_mode(His_obj obj);
55
59 class Data {
60 private:
64     std::vector<std::vector<std::unique_ptr<Ticket>> all_tickets;
68     std::list<std::unique_ptr<Ticket> reservation;
72     std::vector<His_obj> history;
76     void add_new_reservation(const std::string& name, const std::string& surname, const int& verse, const
      int& column, const int& type);
77 public:
78     Data();
79     ~Data();
83     void display_cinema();
87     void display_sits_info();
91     void make_reservation();
95     void choose_sits(int& verse, int& column);
99     void search_reservation();
103     void display_reserwation();
107     void cancel_reserwation();
```

```
111     void sort_reservation();
115     void display_reservation_ranking();
119     void add_to_his(const std::string& name, const std::string& surname, const double& price, const int&
       verse, const int& column, const int& type, const int& mode);
123     void print_raport();
124 };
```

5.2 Menu.h

```
1 #pragma once
2 #include "Data.h"
3
9 enum Options{
10     Option_One,
11     Option_Two,
12     Option_Three,
13     Option_Four,
14     Option_Five,
15 };
16
17
21 class Menu {
22 private:
23     Data reservation_info;
24 public:
25     Menu();
26     ~Menu();
30     void display_opt();
34     void start_prog();
35 };
```

5.3 Ticket.h

```
1 #pragma once
2 #include <string>
3
9 const double Normal_Price = 25;
10 const double Student_Price = 12.5;
11 const double Discount_Price = 15;
12 const double Family_Price = 17.5;
13
17 class Ticket {
18 protected:
19     std::string name;
20     std::string surname;
21     double price;
22     bool is_free;
23     int verse, column;
24     Ticket(double price);
25 public:
26     Ticket();
27     Ticket(std::string name, std::string surname, double price, int verse, int column);
28     Ticket(std::string name, std::string surname, int verse, int column);
29     ~Ticket();
33     void set_name(const std::string& name);
37     void set_surname(const std::string& surname);
41     void set_price(const double& price);
45     std::string get_name();
49     std::string get_surname();
53     double get_price();
57     bool get_is_free();
61     int get_verse();
65     int get_column();
69     void reserve();
73     void unreserve();
77     virtual void display_sit();
81     virtual void display_ticket_type();
85     virtual int type();
86 };
```

5.4 Ticket_D.h

```
1 #pragma once
2 #include "Ticket.h"
3
```

```
13 class Ticket_D : public Ticket {
14 public:
15     Ticket_D();
16     Ticket_D(std::string name, std::string surname, double price, int verse, int column);
20     virtual void display_sit();
24     virtual void display_ticket_type();
28     virtual int type();
29 };
```

5.5 Ticket_F.h

```
1 #pragma once
2 #include "Ticket.h"
3
12 class Ticket_F : public Ticket {
13 public:
14     Ticket_F();
15     Ticket_F(std::string name, std::string surname, double price, int verse, int column);
19     virtual void display_sit();
23     virtual void display_ticket_type();
27     virtual int type();
28 };
```

5.6 Ticket_S.h

```
1 #pragma once
2 #include "Ticket.h"
3
12 class Ticket_S : public Ticket {
13 public:
14     Ticket_S();
15     Ticket_S(std::string name, std::string surname, double price, int verse, int column);
19     virtual void display_sit();
23     virtual void display_ticket_type();
27     virtual int type();
28 };
```

Indeks