Lokal Gastronomiczny

Generated by Doxygen 1.9.4

1	Hierarchical Index	1
	1.1 Class Hierarchy	1
2	Class Index	3
	2.1 Class List	3
3	File Index	5
	3.1 File List	5
4	Class Documentation	7
	4.1 Danie Class Reference	7
	4.1.1 Detailed Description	7
	4.1.2 Constructor & Destructor Documentation	7
	4.1.2.1 Danie()	8
	4.1.3 Member Function Documentation	9
	4.1.3.1 operator<=>()	9
	4.1.3.2 pokaz_czas()	9
	4.1.4 Friends And Related Function Documentation	9
	4.1.4.1 operator <<	9
	4.1.4.2 Zamowienie	9
	4.1.4.3 Zrealizowane_zamowienie	10
	4.1.5 Member Data Documentation	10
	4.1.5.1 cena	10
	4.1.5.2 czas_przygotwania	10
	4.1.5.3 nazwa	10
	4.1.5.4 wielkosc_porcji	10
	4.2 Kelner Class Reference	11
	4.2.1 Detailed Description	11
	4.2.2 Constructor & Destructor Documentation	11
	4.2.2.1 Kelner()	11
	4.2.3 Member Function Documentation	12
	4.2.3.1 aktualizuj()	12
	4.2.3.2 odwolaj()	12
	4.2.3.3 wyswietl_pracownika()	12
	4.2.3.4 zajmij_wolnego_pracownika()	12
	4.2.3.5 znajdz_wolnego_pracownika()	13
	4.3 Kucharz Class Reference	13
	4.3.1 Detailed Description	13
	4.3.2 Constructor & Destructor Documentation	14
	4.3.2.1 Kucharz()	14
	4.3.3 Member Function Documentation	14
	4.3.3.1 aktualizuj()	14
	4.3.3.2 odwolaj()	14

4.3.3.3 wyswietl_pracownika()	 15
4.3.3.4 zajmij_wolnego_pracownika()	 15
4.3.3.5 znajdz_wolnego_pracownika()	 15
4.4 Kuchnia Class Reference	 15
4.4.1 Detailed Description	 16
4.4.2 Constructor & Destructor Documentation	 16
4.4.2.1 Kuchnia()	 16
4.4.2.2 ~ Kuchnia()	 16
4.4.3 Member Function Documentation	 16
4.4.3.1 pokaz_sklad_kuchni()	 16
4.4.3.2 zajmij_pracownika()	 16
4.4.3.3 znajdz_wolnego_kelnera()	 17
4.4.3.4 znajdz_wolnego_kucharza()	 17
4.4.4 Member Data Documentation	 17
4.4.4.1 pracownicy	 17
4.5 ListaStolikow Class Reference	 17
4.5.1 Detailed Description	 18
4.5.2 Constructor & Destructor Documentation	 18
4.5.2.1 ListaStolikow()	 18
4.5.3 Member Function Documentation	 18
4.5.3.1 dostep_stoliki()	 18
4.5.3.2 szukaj_stolika()	 19
4.5.3.3 wyswietl_stoliki()	 19
4.5.3.4 zwolnij_stolik()	 19
4.5.4 Member Data Documentation	 19
4.5.4.1 stoliki	 19
4.6 Menu Class Reference	 20
4.6.1 Detailed Description	 20
4.6.2 Constructor & Destructor Documentation	 20
4.6.2.1 Menu()	 20
4.6.3 Member Function Documentation	 20
4.6.3.1 wyswietl_menu()	 21
4.6.4 Friends And Related Function Documentation	 21
4.6.4.1 Zamowienie	 21
4.6.5 Member Data Documentation	 21
4.6.5.1 lista_dan	 21
4.7 Obserwator Class Reference	 21
4.7.1 Detailed Description	 22
4.7.2 Constructor & Destructor Documentation	 22
4.7.2.1 ∼Obserwator()	 22
4.7.3 Member Function Documentation	 22
4.7.3.1 aktualizuj()	 22

4.7.3.2 odwolaj()	22
4.8 Pracownik Class Reference	23
4.8.1 Detailed Description	24
4.8.2 Constructor & Destructor Documentation	24
4.8.2.1 Pracownik()	24
4.8.3 Member Function Documentation	24
4.8.3.1 aktualizuj()	24
4.8.3.2 dostep_aktywny()	24
4.8.3.3 dostep_imie()	25
4.8.3.4 dostep_nazwisko()	25
4.8.3.5 wyswietl_pracownika()	25
4.8.3.6 zajmij_wolnego_pracownika()	25
4.8.3.7 znajdz_wolnego_pracownika()	26
4.8.4 Member Data Documentation	26
4.8.4.1 aktywny	26
4.8.4.2 czas_na_ile_zajety	26
4.8.4.3 imie	26
4.8.4.4 nazwisko	26
4.9 Stolik Class Reference	26
4.9.1 Detailed Description	27
4.9.2 Constructor & Destructor Documentation	27
4.9.2.1 Stolik()	27
4.9.3 Member Function Documentation	27
4.9.3.1 dostep_do_maxa()	28
4.9.3.2 dostep_numer_stolika()	28
4.9.3.3 pokaz_stolik()	28
4.9.3.4 zajmij()	28
4.9.4 Friends And Related Function Documentation	28
4.9.4.1 ListaStolikow	28
4.9.4.2 wyznacz_id_wolny_stolik_min_os	28
4.9.5 Member Data Documentation	29
4.9.5.1 aktualni_uzytkownicy	29
4.9.5.2 max_ilosc_osob	29
4.9.5.3 numer	29
4.9.5.4 wolny	29
4.10 Zamowienie Class Reference	29
4.10.1 Detailed Description	30
4.10.2 Constructor & Destructor Documentation	30
4.10.2.1 Zamowienie()	30
4.10.3 Member Function Documentation	31
4.10.3.1 anuluj()	31
4.10.3.2 dodaj_danie_do_zamowienia()	31

4.10.3.3 dopisz()	31
4.10.3.4 dostep_stolik()	31
4.10.3.5 oblicz_czas_przygotowania()	32
4.10.3.6 oblicz_koszt()	32
4.10.3.7 pobierz_zamowione()	32
4.10.3.8 pokaz_zamowione_rzeczy()	32
4.10.3.9 powiadom()	32
4.10.3.10 prezentuj_aktualne_zakupy()	32
4.10.3.11 wypisz()	32
4.10.3.12 zloz_zamowienie()	33
4.10.4 Member Data Documentation	33
4.10.4.1 ctr	33
4.10.4.2 menu	33
4.10.4.3 obserwatorzy	33
4.10.4.4 stolik	33
4.10.4.5 zamowione_dania	34
4.11 Zrealizowane_zamowienie Class Reference	34
4.11.1 Detailed Description	34
4.11.2 Member Function Documentation	35
4.11.2.1 aktualizuj()	35
4.11.2.2 odwolaj()	36
4.11.2.3 pokaz_statystyki()	36
4.11.3 Friends And Related Function Documentation	36
4.11.3.1 Zamowienie	36
4.11.4 Member Data Documentation	36
4.11.4.1 ilosc	36
4.11.4.2 zamowienia	36
5 File Documentation	37
5.1 C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal_gastronomiczi	
Project1/dania.h File Reference	37
5.2 dania.h	37
5.3 C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal_gastronomiczi	•
Project1/funkcje.cpp File Reference	38
5.3.1 Function Documentation	38
5.3.1.1 operator<<()	39
5.3.1.2 pokaz_menu_dzialania_wstep()	39
5.3.1.3 pokaz_menu_dzialania_zamowienie()	39
5.3.1.4 wyznacz_id_wolny_stolik_min_os()	39
5.4 C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal_gastronomiczi	ny/← 39
5.4.1 Function Documentation	40
5.4.1.1 konwerter_ze_stringa()	

5.4.1.2 pokaz_menu_dzialania_wstep()	40
5.4.1.3 pokaz_menu_dzialania_zamowienie()	40
5.4.1.4 wyznacz_id_wolny_stolik_min_os()	40
5.5 funkcje.h	41
5.6 C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal_gastronomiczny/ <pre>Project1/main.cpp File Reference</pre>	
5.7 C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal_gastronomiczny/ <a href="mailto:Project1/pracownicy.h">Project1/pracownicy.h File Reference</a>	← 42
5.8 pracownicy.h	42
5.9 C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal_gastronomiczny/ <pre>Project1/stolik.h File Reference</pre>	← 45
5.10 stolik.h	45
5.11 C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal_gastronomiczny Project1/zamowienia.h File Reference	//← 47
5.12 zamowienia.h	47
5.13 C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal_gastronomiczny Project1/zrealizowane_zamowienie.h File Reference	//← 50
5.14 zrealizowane_zamowienie.h	50
Index	51

# **Chapter 1**

# **Hierarchical Index**

# 1.1 Class Hierarchy

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

anie	7
uchnia	Ę
istaStolikow	
lenu	
bserwator	1
Pracownik	3
Kelner	
Kucharz	3
Zrealizowane_zamowienie	4
tolik	e
amowienie 2	c

2 Hierarchical Index

# Chapter 2

# **Class Index**

# 2.1 Class List

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

Danie																											7
Kelner																											11
Kucharz																											13
Kuchnia																											15
ListaStolikow																											17
Menu																											20
Obserwator																											
Pracownik .																											
Stolik																											26
Zamowienie																											29
<b>7realizowane</b>	7	aı	mr	אור	ιie	'ni	ie																				34

4 Class Index

# **Chapter 3**

# File Index

# 3.1 File List

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

C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal_gastronomiczny/	
Project1/dania.h	37
C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal_gastronomiczny/	
Project1/funkcje.cpp	38
C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal_gastronomiczny/	
Project1/funkcje.h	39
C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal_gastronomiczny/	
Project1/main.cpp	41
C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal_gastronomiczny/	
Project1/pracownicy.h	42
C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal_gastronomiczny/	
Project1/stolik.h	45
C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal_gastronomiczny/	
Project1/zamowienia.h	47
C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal_gastronomiczny/	
Project1/zrealizowane zamowienie h	50

6 File Index

# **Chapter 4**

# **Class Documentation**

# 4.1 Danie Class Reference

```
#include <dania.h>
```

#### **Public Member Functions**

- Danie (std::string nazwa, int czas\_przygotowania, int wielkosc\_porcji, double cena)
- int pokaz czas ()
- auto operator<=> (const Danie &rhs) const =default

#### **Protected Attributes**

- std::string nazwa
- int czas\_przygotwania
- int wielkosc\_porcji
- double cena

#### **Friends**

- class Zamowienie
- · class Zrealizowane zamowienie
- std::ostream & operator<< (std::ostream &s, const Danie &danie)

### 4.1.1 Detailed Description

Klasa opisujaca parametry dania

#### 4.1.2 Constructor & Destructor Documentation

# 4.1.2.1 Danie()

Konstruktor klasy Danie inicjuje zmienne wartosciami podanymi do konstruktora

4.1 Danie Class Reference 9

#### **Parameters**

nazwa	nazwa dania
czas_przygotowania	czas przygotowania dania
wielkosc_porcji	wielkosc porcji dania
cena	cena dania

#### 4.1.3 Member Function Documentation

#### 4.1.3.1 operator<=>()

Domyslny operator spaceship dla wlasnego typu

#### 4.1.3.2 pokaz\_czas()

```
int Danie::pokaz_czas ( ) [inline]
```

Funkcja zwracajaca czas przygotowania

Returns

funkcja zwraca czas przygotowania dania

#### 4.1.4 Friends And Related Function Documentation

## 4.1.4.1 operator <<

```
std::ostream & operator<< (
          std::ostream & s,
          const Danie & danie ) [friend]</pre>
```

zakolegowanie funkcji przeciazajacej operator Dania z klasa Dania

# 4.1.4.2 Zamowienie

```
friend class Zamowienie [friend]
```

zakolegowanie klasy Zamowienie z klasa Danie

#### 4.1.4.3 Zrealizowane\_zamowienie

```
friend class Zrealizowane_zamowienie [friend]
```

zakolegowanie klasy Zrealizowane\_zamowienie z klasa Danie

#### 4.1.5 Member Data Documentation

#### 4.1.5.1 cena

```
double Danie::cena [protected]
```

cena dania w zlotowkach

#### 4.1.5.2 czas\_przygotwania

```
int Danie::czas_przygotwania [protected]
```

czas przygotowania dania w minutosekundach (normalnie to byly by minuty, ale czekac az zrobi sie zamowienie kilka minut, to troche bez sensu, dlatego 1min=1sec

#### 4.1.5.3 nazwa

```
std::string Danie::nazwa [protected]
```

nazwa dania

#### 4.1.5.4 wielkosc\_porcji

```
int Danie::wielkosc_porcji [protected]
```

wielkosc porcji w gramach

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

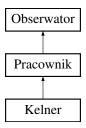
• C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal\_gastronomiczny/← Project1/dania.h

4.2 Kelner Class Reference

# 4.2 Kelner Class Reference

```
#include cownicy.h>
```

Inheritance diagram for Kelner:



#### **Public Member Functions**

- Kelner (std::string imie, std::string nazwisko)
- void aktualizuj (Zamowienie &h) override
- void odwolaj () override
- void wyswietl\_pracownika () override
- Pracownik \* znajdz wolnego pracownika () override
- void zajmij\_wolnego\_pracownika () override

#### **Additional Inherited Members**

### 4.2.1 Detailed Description

Klasa Kelner dziedziczaca publicznie po klasie Pracownik

### 4.2.2 Constructor & Destructor Documentation

#### 4.2.2.1 Kelner()

Konstruktor klasy Kelner inicjujacy zmienne wartosciami podanymi w konstruktorze, korzysta z konstruktora klasy bazowej Pracownik

#### **Parameters**

imie	imie pracownika
nazwisko	nazwisko pracownika

# 4.2.3 Member Function Documentation

#### 4.2.3.1 aktualizuj()

Funkcja aktualizuje kelnera, to znaczy przydziela mu zamowienie i karze mu je dostarczyc

**Parameters** 

```
zam zamowienie przekazane do realizacji
```

Reimplemented from Pracownik.

#### 4.2.3.2 odwolaj()

```
void Kelner::odwolaj ( ) [inline], [override], [virtual]
```

Funkcja dezaktywuje kelnera

Implements Obserwator.

#### 4.2.3.3 wyswietl\_pracownika()

```
void Kelner::wyswietl_pracownika ( ) [inline], [override], [virtual]
```

Funkcja wyswietla pracownika -> tutaj kelnera

Implements Pracownik.

#### 4.2.3.4 zajmij wolnego pracownika()

```
void Kelner::zajmij_wolnego_pracownika ( ) [inline], [override], [virtual]
```

Funkcja zajmuje wolnego pracownika w tym przypadku kelnera

Implements Pracownik.

#### 4.2.3.5 znajdz\_wolnego\_pracownika()

```
Pracownik * Kelner::znajdz_wolnego_pracownika ( ) [inline], [override], [virtual]
```

Funkcja szuka wolnego pracownika i ponadto znajduje takowego

#### Returns

wskaznik na wolnego pracownika, no w tym przypadku nie da sie ukryc ze to bedzie kelner. A jesli nie znajdzie wolnego to wtedy zwraca nullptr

Implements Pracownik.

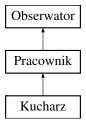
The documentation for this class was generated from the following file:

C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal\_gastronomiczny/
 — Project1/pracownicy.h

#### 4.3 Kucharz Class Reference

```
#include <pracownicy.h>
```

Inheritance diagram for Kucharz:



#### **Public Member Functions**

- Kucharz (std::string imie, std::string nazwisko)
- · void aktualizuj (Zamowienie &zam) override
- void odwolaj () override
- void wyswietl\_pracownika () override
- Pracownik \* znajdz\_wolnego\_pracownika () override
- void zajmij\_wolnego\_pracownika () override

#### **Additional Inherited Members**

## 4.3.1 Detailed Description

Klasa Kucharz dziedziczaca publicznie po klasie Pracownik

#### 4.3.2 Constructor & Destructor Documentation

#### 4.3.2.1 Kucharz()

Konstruktor klasy Kucharz inicjujacy zmienne wartosciami podanymi w konstruktorze, korzysta z konstruktora klasy bazowej Pracownik

#### **Parameters**

imie	imie pracownika
nazwisko	nazwisko pracownika

#### 4.3.3 Member Function Documentation

#### 4.3.3.1 aktualizuj()

Funkcja aktualizuje kucharza, to znaczy przydziela mu zamowienie i karze mu je zrobic

#### **Parameters**

zam	zamowienie przekazane do realizacji

Reimplemented from Pracownik.

#### 4.3.3.2 odwolaj()

```
void Kucharz::odwolaj ( ) [inline], [override], [virtual]
```

Funkcja dezaktywuje kucharza

Implements Obserwator.

#### 4.3.3.3 wyswietl\_pracownika()

```
void Kucharz::wyswietl_pracownika ( ) [inline], [override], [virtual]
```

Funkcja wyswietla pracownika -> tutaj kucharza

Implements Pracownik.

#### 4.3.3.4 zajmij\_wolnego\_pracownika()

```
void Kucharz::zajmij_wolnego_pracownika ( ) [inline], [override], [virtual]
```

Funkcja zajmuje wolnego pracownika w tym przypadku kucharza

Implements Pracownik.

#### 4.3.3.5 znajdz\_wolnego\_pracownika()

```
Pracownik * Kucharz::znajdz_wolnego_pracownika ( ) [inline], [override], [virtual]
```

Funkcja szuka wolnego pracownika i ponadto znajduje takowego

Returns

wskaznik na wolnego pracownika, no w tym przypadku nie da sie ukryc ze to bedzie kucharz. A jesli nie znajdzie wolnego to wtedy zwraca nullptr

Implements Pracownik.

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

C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal\_gastronomiczny/
 — Project1/pracownicy.h

#### 4.4 Kuchnia Class Reference

#include <pracownicy.h>

### **Public Member Functions**

- Kuchnia (std::string nazwa\_pliku)
- void pokaz\_sklad\_kuchni ()
- Pracownik \* znajdz\_wolnego\_kelnera ()
- Pracownik \* znajdz\_wolnego\_kucharza ()
- void zajmij\_pracownika (Pracownik \*pracownik)
- ∼Kuchnia ()

#### **Protected Attributes**

std::vector < Pracownik \* > pracownicy

# 4.4.1 Detailed Description

Klasa Kuchnia

## 4.4.2 Constructor & Destructor Documentation

#### 4.4.2.1 Kuchnia()

Konstruktor klasy Kuchnia inicjuje tablice pracownikami podanymi w pliku, ktorego nazwa jest podawana do konstruktora

#### **Parameters**

#### 4.4.2.2 ∼Kuchnia()

```
Kuchnia::~Kuchnia ( ) [inline]
```

Destruktor klasy Kuchnia czysci pamiec

#### 4.4.3 Member Function Documentation

#### 4.4.3.1 pokaz\_sklad\_kuchni()

```
void Kuchnia::pokaz_sklad_kuchni ( ) [inline]
```

Funkcja wyswietla sklad personelu kuchennego

#### 4.4.3.2 zajmij\_pracownika()

Funkcja zajmuje wolnego pracownika

#### **Parameters**

#### 4.4.3.3 znajdz\_wolnego\_kelnera()

```
Pracownik * Kuchnia::znajdz_wolnego_kelnera ( ) [inline]
```

Funkcja szuka wolnego kelnera

Returns

wskaznik na wolnego kelnera. A jesli nie znajdzie wolnego to wtedy zwraca nullptr

#### 4.4.3.4 znajdz\_wolnego\_kucharza()

```
Pracownik * Kuchnia::znajdz_wolnego_kucharza ( ) [inline]
```

Funkcja szuka wolnego kucharza

Returns

wskaznik na wolnego kucharza. A jesli nie znajdzie wolnego to wtedy zwraca nullptr

#### 4.4.4 Member Data Documentation

# 4.4.4.1 pracownicy

```
std::vector<Pracownik*> Kuchnia::pracownicy [protected]
```

Tablica typu vetor przetrzymujaca wskazniki na pracownikow

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

C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal\_gastronomiczny/
 — Project1/pracownicy.h

#### 4.5 ListaStolikow Class Reference

```
#include <stolik.h>
```

#### **Public Member Functions**

- std::vector< Stolik > dostep\_stoliki ()
- ListaStolikow (std::string nazwa\_pliku)
- · void wyswietl stoliki ()
- void zwolnij\_stolik (int poz)
- int szukaj\_stolika (int ilosc\_chetnych)

#### **Protected Attributes**

std::vector < Stolik > stoliki

# 4.5.1 Detailed Description

Klasa ListaStolikow

#### 4.5.2 Constructor & Destructor Documentation

#### 4.5.2.1 ListaStolikow()

```
ListaStolikow::ListaStolikow (
std::string nazwa_pliku) [inline]
```

Konstruktor dla klasy ListaStolikow, inicjuje zmienna Stolikami zawartymi w pliku

**Parameters** 

nazwa\_pliku | nazwa pliku ze stolikami

#### 4.5.3 Member Function Documentation

#### 4.5.3.1 dostep\_stoliki()

```
\verb|std::vector| < Stolik| > ListaStolikow::dostep_stoliki () [inline]|
```

Funkcja zwraca stoliki

Returns

stoliki zwraca stoliki

#### 4.5.3.2 szukaj\_stolika()

Funkcja szuka stolik i zwraca zajeta pozycje

**Parameters** 

**Returns** 

pozycja znalezionego stolika

#### 4.5.3.3 wyswietl\_stoliki()

```
void ListaStolikow::wyswietl_stoliki ( ) [inline]
```

Funkcja wyswietla stoliki

#### 4.5.3.4 zwolnij\_stolik()

Funkcja zwalnia stolik

#### 4.5.4 Member Data Documentation

#### 4.5.4.1 stoliki

```
std::vector<Stolik> ListaStolikow::stoliki [protected]
```

tablica wektorowa Stolikow

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

• C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal\_gastronomiczny/← Project1/stolik.h

# 4.6 Menu Class Reference

```
#include <dania.h>
```

#### **Public Member Functions**

- Menu (std::string nazwa\_pliku)
- void wyswietl\_menu ()

# **Protected Attributes**

• std::vector< Danie > lista\_dan

#### **Friends**

• class Zamowienie

# 4.6.1 Detailed Description

Klasa opisujaca parametry Menu

#### 4.6.2 Constructor & Destructor Documentation

## 4.6.2.1 Menu()

Konstruktor klasy Menu inicjuje zmienne wartosciami podanymi w pliku, ktorego nazwa jest podawana do konstruktora

#### **Parameters**

```
nazwa_pliku | nazwa pliku z daniami
```

#### 4.6.3 Member Function Documentation

#### 4.6.3.1 wyswietl\_menu()

```
void Menu::wyswietl_menu ( ) [inline]
```

Funkcja wyswietlajaca menu

#### 4.6.4 Friends And Related Function Documentation

#### 4.6.4.1 Zamowienie

```
friend class Zamowienie [friend]
```

zakolegowanie klasy Zamowienie z klasa Menu

#### 4.6.5 Member Data Documentation

# 4.6.5.1 lista\_dan

```
std::vector<Danie> Menu::lista_dan [protected]
```

vector Dan prezentujacy liste dan

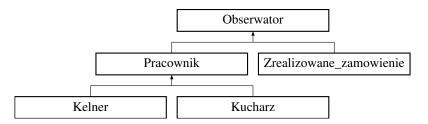
The documentation for this class was generated from the following file:

C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal\_gastronomiczny/
 — Project1/dania.h

# 4.7 Obserwator Class Reference

```
#include <zamowienia.h>
```

Inheritance diagram for Obserwator:



#### **Public Member Functions**

- virtual ∼Obserwator ()
- virtual void aktualizuj (Zamowienie &zamowienie)=0
- virtual void odwolaj ()=0

# 4.7.1 Detailed Description

Klasa Obserwator

#### 4.7.2 Constructor & Destructor Documentation

#### 4.7.2.1 ∼Obserwator()

```
\label{local_problem} \mbox{virtual Obserwator::$$\sim$Obserwator ( ) [inline], [virtual]$} $$ \mbox{destruktor witualny} $$
```

# 4.7.3 Member Function Documentation

#### 4.7.3.1 aktualizuj()

Metoda czysto wirtualna aktualizujaca obserwatorow

**Parameters** 

zamowienie	zamowienie przeslane do oberwatorow

Implemented in Pracownik, Kelner, Kucharz, and Zrealizowane\_zamowienie.

## 4.7.3.2 odwolaj()

```
virtual void Obserwator::odwolaj ( ) [pure virtual]
```

Metoda czysto wirtualna odwolujaca obserwatorow

#### **Parameters**

zamowienie zamowienie przeslane do oberwatorow	)W
--	----

Implemented in Kucharz, Kelner, and Zrealizowane\_zamowienie.

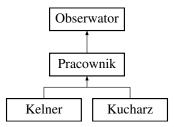
The documentation for this class was generated from the following file:

C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal\_gastronomiczny/
 — Project1/zamowienia.h

#### 4.8 Pracownik Class Reference

#include pracownicy.h>

Inheritance diagram for Pracownik:



#### **Public Member Functions**

- Pracownik (std::string imie, std::string nazwisko)
- std::string dostep\_imie ()
- std::string dostep\_nazwisko ()
- bool dostep\_aktywny ()
- virtual void wyswietl\_pracownika ()=0
- virtual Pracownik \* znajdz\_wolnego\_pracownika ()=0
- virtual void zajmij wolnego pracownika ()=0

#### **Protected Member Functions**

· virtual void aktualizuj (Zamowienie &h)

#### **Protected Attributes**

- std::string imie
- std::string nazwisko
- · bool aktywny
- int czas\_na\_ile\_zajety

# 4.8.1 Detailed Description

Klasa Pracownik dziedziczaca publicznie po klasie Obserwator

#### 4.8.2 Constructor & Destructor Documentation

#### 4.8.2.1 Pracownik()

Konstruktor klasy Pracownik inicjujacy zmienne wartosciami podanymi w konstruktorze, ponadto ustawiajacy status aktywny na false i czas\_na\_ile\_zajety na 0

#### **Parameters**

imie	imie pracownika
nazwisko	nazwisko pracownika

#### 4.8.3 Member Function Documentation

## 4.8.3.1 aktualizuj()

Metoda czysto wirtualna aktualizujaca obserwatorow

#### **Parameters**

zamowienie	zamowienie przeslane do oberwatorow
------------	-------------------------------------

Implements Obserwator.

Reimplemented in Kelner, and Kucharz.

#### 4.8.3.2 dostep\_aktywny()

```
bool Pracownik::dostep_aktywny ( ) [inline]
```

Metoda zwracajaca czy pracownik jest aktywny

Returns

aktywny (true / false)

#### 4.8.3.3 dostep\_imie()

```
std::string Pracownik::dostep_imie ( ) [inline]
```

Metoda zwracajaca imie pracownika

**Returns** 

imie

#### 4.8.3.4 dostep\_nazwisko()

```
std::string Pracownik::dostep_nazwisko ( ) [inline]
```

Metoda zwracajaca nazwisko pracownika

Returns

nazwisko

# 4.8.3.5 wyswietl\_pracownika()

```
virtual void Pracownik::wyswietl_pracownika ( ) [pure virtual]
```

Metoda wirtualna wyswietlajaca pracownika

Implemented in Kucharz, and Kelner.

#### 4.8.3.6 zajmij\_wolnego\_pracownika()

```
virtual void Pracownik::zajmij_wolnego_pracownika ( ) [pure virtual]
```

Metoda wirtualna zajmujaca wolnego pracownika

Implemented in Kucharz, and Kelner.

#### 4.8.3.7 znajdz\_wolnego\_pracownika()

```
virtual Pracownik * Pracownik::znajdz_wolnego_pracownika ( ) [pure virtual]
```

Metoda wirtualna szukajaca wolnego pracownika

Implemented in Kucharz, and Kelner.

#### 4.8.4 Member Data Documentation

#### 4.8.4.1 aktywny

```
bool Pracownik::aktywny [protected]
czy aktywny
```

#### 4.8.4.2 czas\_na\_ile\_zajety

```
int Pracownik::czas_na_ile_zajety [protected]
czas na ile zajety
```

#### 4.8.4.3 imie

```
std::string Pracownik::imie [protected]
```

imie pracownika

#### 4.8.4.4 nazwisko

```
std::string Pracownik::nazwisko [protected]
```

nazwisko pracownika

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

• C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal\_gastronomiczny/← Project1/pracownicy.h

# 4.9 Stolik Class Reference

#include <stolik.h>

4.9 Stolik Class Reference 27

#### **Public Member Functions**

- Stolik (int numer, int max\_ilosc\_osob)
- int dostep\_do\_maxa ()
- int dostep\_numer\_stolika ()
- void pokaz\_stolik ()
- void zajmij ()

#### **Protected Attributes**

- int numer
- int max\_ilosc\_osob
- int aktualni\_uzytkownicy
- bool wolny

#### **Friends**

- · class ListaStolikow
- int wyznacz\_id\_wolny\_stolik\_min\_os (std::vector< Stolik >stoliki, std::vector< int > pozycje)

# 4.9.1 Detailed Description

Klasa Stolik

#### 4.9.2 Constructor & Destructor Documentation

#### 4.9.2.1 Stolik()

```
Stolik::Stolik (
                int numer,
                int max_ilosc_osob ) [inline]
```

Konstruktor klasy Stolik inicjuje zmienne wartosciami podanymi do konstruktora, ponadto ustawia stolik na wolny i aktualnych uzytkownikow na 0

#### **Parameters**

numer	numer stolika
max_ilosc_osob	maksymalna ilosc osob przy danym stoliku

#### 4.9.3 Member Function Documentation

#### 4.9.3.1 dostep\_do\_maxa()

```
int Stolik::dostep_do_maxa ( ) [inline]
```

Funkcja zwraca wartosc maksymalnej liczby osob przy stole

Returns

max\_ilosc\_osob maksymalna ilosc osob przy danym stoliku

#### 4.9.3.2 dostep\_numer\_stolika()

```
int Stolik::dostep_numer_stolika ( ) [inline]
```

Funkcja zwraca numer stolika

Returns

numer numer stolika

#### 4.9.3.3 pokaz\_stolik()

```
void Stolik::pokaz_stolik ( ) [inline]
```

Funkcja pokazuje stolik

#### 4.9.3.4 zajmij()

```
void Stolik::zajmij ( ) [inline]
```

Funkcja zajmuje stolik

### 4.9.4 Friends And Related Function Documentation

#### 4.9.4.1 ListaStolikow

```
friend class ListaStolikow [friend]
```

zakolegowanie klasy Stolik z klasa ListaStolikow

#### 4.9.4.2 wyznacz\_id\_wolny\_stolik\_min\_os

zakolegowanie klasy Stolik z funkcja wyznacz\_id\_wolny\_stolik\_min\_os

Funkcja wyznaczajaca id pierwszego wolnego stolika jesli liczba chetnyc

#### **Parameters**

stoliki	vector slolikow
pozycje	vector pozycji

#### Returns

pozycja wolnego stolika na liscie stolikow

#### 4.9.5 Member Data Documentation

#### 4.9.5.1 aktualni\_uzytkownicy

```
int Stolik::aktualni_uzytkownicy [protected]
```

liczba aktualnych uzytkownikow stolika

#### 4.9.5.2 max\_ilosc\_osob

```
int Stolik::max_ilosc_osob [protected]
```

maksymalna ilosc osob przy danym stoliku

#### 4.9.5.3 numer

```
int Stolik::numer [protected]
```

numer stolika

#### 4.9.5.4 wolny

```
bool Stolik::wolny [protected]
```

czy stolik wolny

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

C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal\_gastronomiczny/
 — Project1/stolik.h

#### 4.10 Zamowienie Class Reference

#include <zamowienia.h>

30 Class Documentation

#### **Public Member Functions**

- Zamowienie (Menu \*menu, Stolik stolik)
- std::vector< Danie > pobierz\_zamowione () const
- Stolik dostep\_stolik ()
- void dopisz (Obserwator \*obserwator)
- void wypisz (Obserwator \*obserwator)
- bool zloz\_zamowienie ()
- void dodaj\_danie\_do\_zamowienia (int poz)
- double oblicz koszt ()
- double oblicz czas przygotowania ()
- void pokaz\_zamowione\_rzeczy ()
- void prezentuj\_aktualne\_zakupy ()

#### **Protected Member Functions**

- · void powiadom ()
- void anuluj ()

#### **Protected Attributes**

- Stolik stolik
- std::vector< Danie > zamowione\_dania
- Menu \* menu
- std::set< Obserwator \* > obserwatorzy

#### **Static Protected Attributes**

· static int ctr

#### 4.10.1 Detailed Description

Klasa Zamowienie

#### 4.10.2 Constructor & Destructor Documentation

#### 4.10.2.1 Zamowienie()

Konstruktor klasy Zamowienie inicjujacy zmienne obiektami podanymi w wejsciu i inkrementujacy ilosc zamowien (ctr)

#### **Parameters**

menu	wskaznik na menu
stolik	stolik dla ktorego jest zamowienie

#### 4.10.3 Member Function Documentation

#### 4.10.3.1 anuluj()

```
void Zamowienie::anuluj ( ) [inline], [protected]
```

Funkcja anulujaca dzialanie obserwatorow

#### 4.10.3.2 dodaj\_danie\_do\_zamowienia()

Funkcja dodaje danie do zamowienia param poz pozycja dania z menu

#### 4.10.3.3 dopisz()

```
void Zamowienie::dopisz (
          Obserwator * obserwator ) [inline]
```

Funkcja pozwalajaca dopisac obserwatora do zbioru obserwatorow

#### **Parameters**

obserwator	obserwator do dodania
0.000	obootinato. do doddina

#### 4.10.3.4 dostep\_stolik()

```
Stolik Zamowienie::dostep_stolik ( ) [inline]
```

Funkcja zwraca stolik

Returns

stolik

32 Class Documentation

```
4.10.3.5 oblicz_czas_przygotowania()
```

```
double Zamowienie::oblicz_czas_przygotowania ( ) [inline]
```

Funkcja obliczajaca czas przygotowania zamowienia

Returns

calkowity czas przygotowania zamowienia

#### 4.10.3.6 oblicz\_koszt()

```
double Zamowienie::oblicz_koszt ( ) [inline]
```

Funkcja obliczajaca koszt zamowienia

Returns

calkowity koszt zamowienia

#### 4.10.3.7 pobierz zamowione()

```
std::vector< Danie > Zamowienie::pobierz_zamowione ( ) const [inline]
```

Funkcja zwraca zamowione dania

Returns

zamowione\_dania

#### 4.10.3.8 pokaz\_zamowione\_rzeczy()

```
void Zamowienie::pokaz_zamowione_rzeczy ( ) [inline]
```

Funkcja pokazujaca zamowione rzeczy

#### 4.10.3.9 powiadom()

```
void Zamowienie::powiadom ( ) [inline], [protected]
```

Funkcja powiadamiajaca obserwatorow

#### 4.10.3.10 prezentuj\_aktualne\_zakupy()

```
void Zamowienie::prezentuj_aktualne_zakupy ( ) [inline]
```

Funkcja prezentujaca aktualne zakupy

#### 4.10.3.11 wypisz()

```
void Zamowienie::wypisz (
                Obserwator * obserwator ) [inline]
```

Funkcja pozwalajaca usunac obserwatora do zbioru obserwatorow

#### **Parameters**

obserwator	obserwator do usunieca z listy
------------	--------------------------------

#### 4.10.3.12 zloz\_zamowienie()

```
bool Zamowienie::zloz_zamowienie ( ) [inline]
```

Funkcja do skladania zamowienia

Returns

true jesli sie udalo zamowic, false jesli nie

#### 4.10.4 Member Data Documentation

#### 4.10.4.1 ctr

```
int Zamowienie::ctr [static], [protected]
```

licznik zamowien

#### 4.10.4.2 menu

```
Menu* Zamowienie::menu [protected]
```

wskaznik na klase Menu

#### 4.10.4.3 obserwatorzy

```
std::set<Obserwator*> Zamowienie::obserwatorzy [protected]
```

kontener typu set przetrzymujacy wskazniki na obserwatorow

#### 4.10.4.4 stolik

```
Stolik Zamowienie::stolik [protected]
```

stolik do ktorego ma byc zrealizowane zamowienie

34 Class Documentation

#### 4.10.4.5 zamowione\_dania

```
std::vector<Danie> Zamowienie::zamowione_dania [protected]
```

tablica wektorowa Dan

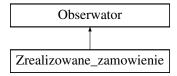
The documentation for this class was generated from the following files:

- C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal\_gastronomiczny/
   — Project1/zamowienia.h
- C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal\_gastronomiczny/
   — Project1/funkcje.cpp

#### 4.11 Zrealizowane zamowienie Class Reference

```
#include <zrealizowane_zamowienie.h>
```

Inheritance diagram for Zrealizowane\_zamowienie:



#### **Public Member Functions**

- void aktualizuj (Zamowienie &zam) override
- void odwolaj () override
- void pokaz\_statystyki ()

#### **Protected Attributes**

std::map< Danie, int > ilosc

#### **Static Protected Attributes**

static std::vector < Zamowienie > zamowienia

#### **Friends**

class Zamowienie

#### 4.11.1 Detailed Description

Klasa Zrealizowane\_zamowienie dziedziczaca publicznie po klasie Obserwator

#### 4.11.2 Member Function Documentation

#### 4.11.2.1 aktualizuj()

Funkcja dodajaca nowe dania do "magazynu", ktore uprzednio zostaly zrealizowane

36 Class Documentation

#### **Parameters**

```
zam zamowienie, ktore zostalo zamowione
```

Implements Obserwator.

#### 4.11.2.2 odwolaj()

```
void Zrealizowane_zamowienie::odwolaj ( ) [inline], [override], [virtual]
```

Funkcja nic nie robi, ale musi byc jako ze jest pochodna od virtualnej

Implements Obserwator.

#### 4.11.2.3 pokaz statystyki()

```
void Zrealizowane_zamowienie::pokaz_statystyki ( ) [inline]
```

Funkcja pokazujaca statystyki, to znaczy zawartosc magazynu

#### 4.11.3 Friends And Related Function Documentation

#### 4.11.3.1 Zamowienie

```
friend class Zamowienie [friend]
```

zakolegowanie klasy Zamowienie z klasa Zrealizowane zamowienie

#### 4.11.4 Member Data Documentation

#### 4.11.4.1 ilosc

```
std::map<Danie, int> Zrealizowane_zamowienie::ilosc [protected]
```

kontener typu mapa, bedacy magazynem zrealizowanych dan

#### 4.11.4.2 zamowienia

```
std::vector<Zamowienie> Zrealizowane_zamowienie::zamowienia [static], [protected]
```

vector Zamowien

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

C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal\_gastronomiczny/
 — Project1/zrealizowane zamowienie.h

### **Chapter 5**

### **File Documentation**

5.1 C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal\_gastronomiczny/Project1/dania.h File Reference

```
#include <iostream>
#include <vector>
#include <string>
#include <fstream>
#include <stdio.h>
#include <sstream>
#include "funkcje.h"
#include <cstdlib>
#include <conio.h>
```

#### Classes

- · class Danie
- class Menu

#### 5.2 dania.h

```
23 public:
       Danie(std::string nazwa, int czas_przygotowania, int wielkosc_porcji, double cena) : nazwa(nazwa),
       czas_przygotwania(czas_przygotowania), wielkosc_porcji(wielkosc_porcji), cena(cena) {}
31
3.5
       int pokaz_czas() { return czas_przygotwania; }
36
38
       auto operator<=>(const Danie& rhs) const = default;
39
40
       friend std::ostream& operator« (std::ostream& s, const Danie& danie);
41
       friend class Zamowienie;
42
       friend class Zrealizowane_zamowienie;
44 };
45
47 class Menu
48 {
49 protected:
50
       std::vector<Danie> lista_dan;
52 public:
      Menu(std::string nazwa_pliku)
           std::string linia;
59
           std::ifstream plikwej(nazwa_pliku);
60
           if (!plikwej)
61
               std::cerr « "Error: nie mozna otworzyc pliku :( " « nazwa_pliku « "\n";
           while (getline(plikwej, linia))
65
66
               std::istringstream dane(linia);
67
               std::string nazwal, cenal, czasl, porcjal;
               dane » nazwal » czasl » porcjal » cenal;
68
69
               Danie danie (nazwal, konwerter_ze_stringa<int>(czasl), konwerter_ze_stringa<int>(porcjal),
       konwerter_ze_stringa<double>(cenal));
70
               lista_dan.push_back(danie);
71
72
       };
73
75
       void wyswietl_menu()
           system("cls");
78
           for (int i = 0; i < lista_dan.size(); i++)</pre>
79
               std::cout « i + 1 « ". ";
80
               std::cout « lista_dan[i];
           std::cout « "\n 0 - zatwierdz operacje";
84
8.5
       friend class Zamowienie;
86 };
```

## 5.3 C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal\_gastronomiczny/Project1/funkcje.cpp File Reference

```
#include "funkcje.h"
#include "stolik.h"
#include "zamowienia.h"
#include <iostream>
```

#### **Functions**

- int wyznacz\_id\_wolny\_stolik\_min\_os (std::vector< Stolik >stoliki, std::vector< int > pozycje)
- void pokaz menu dzialania zamowienie ()
- void pokaz\_menu\_dzialania\_wstep ()
- std::ostream & operator<< (std::ostream &s, const Danie &danie)

#### 5.3.1 Function Documentation

#### 5.3.1.1 operator<<()

```
std::ostream & operator<< (
          std::ostream & s,
          const Danie & danie )</pre>
```

zakolegowanie funkcji przeciazajacej operator Dania z klasa Dania

#### 5.3.1.2 pokaz\_menu\_dzialania\_wstep()

```
void pokaz_menu_dzialania_wstep ( )
```

Funkcja prezentujaca menu dzialania na poczatku programu

#### 5.3.1.3 pokaz\_menu\_dzialania\_zamowienie()

```
void pokaz_menu_dzialania_zamowienie ( )
```

Funkcja prezentujaca menu dzialania zwiazane ze skladaniem zamowienia

#### 5.3.1.4 wyznacz\_id\_wolny\_stolik\_min\_os()

zakolegowanie klasy Stolik z funkcja wyznacz\_id\_wolny\_stolik\_min\_os

## 5.4 C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal\_gastronomiczny/Project1/funkcje.h File Reference

```
#include <vector>
#include <string>
#include "stolik.h"
#include "windows.h"
```

#### **Functions**

- int wyznacz\_id\_wolny\_stolik\_min\_os (std::vector< Stolik >stoliki, std::vector< int > pozycje)
- void pokaz\_menu\_dzialania\_zamowienie ()
- void pokaz\_menu\_dzialania\_wstep ()
- $\bullet \ \ template {<} typename \ T >$

T konwerter\_ze\_stringa (std::string wyraz)

#### 5.4.1 Function Documentation

#### 5.4.1.1 konwerter\_ze\_stringa()

```
template<typename T > T konwerter_ze_stringa ( std::string wyraz )
```

Funkcja szablonowa konwertujaca na podany typ ze stringa

#### **Parameters**

wyraz	wyraz do przekonwertowania
-------	----------------------------

#### Returns

przekonwertowany napis na podany typ

#### 5.4.1.2 pokaz\_menu\_dzialania\_wstep()

```
void pokaz_menu_dzialania_wstep ( )
```

Funkcja prezentujaca menu dzialania na poczatku programu

#### 5.4.1.3 pokaz\_menu\_dzialania\_zamowienie()

```
void pokaz_menu_dzialania_zamowienie ( )
```

Funkcja prezentujaca menu dzialania zwiazane ze skladaniem zamowienia

#### 5.4.1.4 wyznacz\_id\_wolny\_stolik\_min\_os()

Funkcja wyznaczajaca id pierwszego wolnego stolika jesli liczba chetnyc

#### **Parameters**

stoliki	vector slolikow
pozycje	vector pozycji

5.5 funkcje.h

Returns

pozycja wolnego stolika na liscie stolikow

zakolegowanie klasy Stolik z funkcja wyznacz\_id\_wolny\_stolik\_min\_os

#### 5.5 funkcje.h

Go to the documentation of this file.

```
3 #pragma once
4 #include <vector>
5 #include <string>
6 #include "stolik.h"
7 #include "windows.h"
15 int wyznacz_id_wolny_stolik_min_os(std::vector<Stolik>stoliki, std::vector<int> pozycje);
18 void pokaz_menu_dzialania_zamowienie();
21 void pokaz_menu_dzialania_wstep();
27 template <typename T>
28 T konwerter_ze_stringa(std::string wyraz)
29 {
3.0
      T wynik = 0;
      auto it = wyraz.find('.');
31
      if (it == std::string::npos)
32
33
35
36
      bool calkowite = true;
37
       int i_po_przecinku=1;
38
       for (int i = 0; i <wyraz.size(); i++)</pre>
39
41
           if (wyraz[i] == '.')
42
43
               i++;
               calkowite = false;
44
               if (i >= wyraz.size()) break;
45
           if (calkowite)
48
               wynik += ((T) wyraz[i] - '0') * pow(10, wyraz.size()-it-i-1);
49
50
           }
51
           else
               wynik += ((T)wyraz[i] - '0') * pow(10, (0 - i_po_przecinku));
5.5
               i_po_przecinku++;
56
       return wynik;
60 }
```

# 5.6 C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal\_gastronomiczny/Project1/main.cpp File Reference

```
#include <iostream>
#include <string>
#include "funkcje.h"
#include "dania.h"
#include "pracownicy.h"
#include "zamowienia.h"
```

```
#include "zrealizowane_zamowienie.h"
#include "stolik.h"
#include <cstdio>
#include <stdio.h>
#include "windows.h"
#include <conio.h>
```

#### **Functions**

• int main ()

### 5.7 C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11repo/Projekt/Lokal\_gastronomiczny/Project1/pracownicy.h File Reference

```
#include <iostream>
#include <vector>
#include <fstream>
#include <string>
#include <windows.h>
#include "funkcje.h"
#include "zamowienia.h"
#include "zrealizowane_zamowienie.h"
#include "dania.h"
#include <iomanip>
```

#### **Classes**

- · class Pracownik
- · class Kucharz
- · class Kelner
- · class Kuchnia

### 5.8 pracownicy.h

```
1
3 #pragma once
4 #include <iostream>
5 #include <vector>
6 #include <fstream>
7 #include <string>
8 #include <string>
9 #include "funkcje.h"
10 #include "zamowienia.h"
11 #include "zrealizowane_zamowienie.h"
12 #include "dania.h"
13 #include <iomanip>
14
15
17 class Pracownik: public Obserwator
18 {
19 protected:
```

5.8 pracownicy.h 43

```
20
       std::string imie;
       std::string nazwisko;
21
       bool aktywny;
22
       int czas_na_ile_zajety;
2.3
2.4
       /* Metoda wirtualna aktualizuj */
25
       virtual void aktualizuj(Zamowienie& h) {};
26 public:
31
       Pracownik(std::string imie, std::string nazwisko) :imie(imie), nazwisko(nazwisko), aktywny(false),
       czas_na_ile_zajety(0) {}
32
36
       std::string dostep_imie() { return imie; }
37
       std::string dostep_nazwisko() { return nazwisko; }
41
42
46
       bool dostep_aktywny() { return aktywny; }
47
       virtual void wyswietl_pracownika() = 0;
49
50
52
       virtual Pracownik* znajdz_wolnego_pracownika()=0;
       virtual void zajmij_wolnego_pracownika()=0;
55
56 };
57
59 class Kucharz :public Pracownik
60 {
61 public:
       Kucharz(std::string imie, std::string nazwisko) :Pracownik(imie, nazwisko) {}
66
67
71
       void aktualizuj(Zamowienie& zam) override
72
73
            std::cout « "Kucharz w akcji\n";
74
           aktywny = true;
75
            auto zamowione=zam.pobierz_zamowione();
76
            for (int i = 0; i < zamowione.size(); i++)</pre>
77
78
                czas_na_ile_zajety+=zamowione[i].pokaz_czas();
79
           std::cout « "Kucharz zajety na:" « czas_na_ile_zajety « " sec\n";
80
            //Sleep(czas_na_ile_zajety * 1000);
82
            czas_na_ile_zajety = 0;
83
            aktywny = false;
            std::cout « "\n\nZamowienie dla stolika nr: " « zam.dostep_stolik().dostep_numer_stolika() « "
84
       jest gotowe\n";
85
86
88
       void odwolaj() override
89
90
           aktywny = false;
       }
91
92
94
       void wyswietl_pracownika() override
95
       std::cout « std::setw(10) « "Kucharz | "«std::setw(13) « imie « " | " « std::setw(16) « nazwisko « " | " «std::setw(8) « aktywny «" | " « std::setw(15) « czas_na_ile_zajety « "\n";
96
97
101
        Pracownik* znajdz wolnego pracownika() override
102
103
             if (!aktywny)
104
105
                 return this;
106
107
             return nullptr;
108
        }
109
111
        void zajmij_wolnego_pracownika() override
112
113
             aktywny = true;
114
115 };
116
118 class Kelner :public Pracownik
119 {
120 public:
        Kelner(std::string imie, std::string nazwisko) :Pracownik(imie, nazwisko) {}
125
126
130
        void aktualizuj(Zamowienie& h) override
131
             aktywny = true;
std::cout « "\nKelner w akcji\n";
132
133
             Sleep (5000);
134
             std::cout « "Zamowienie dostarczone do stolika nr:
135
       "wh.dostep_stolik().dostep_numer_stolika()w"\n\n";
136
             Sleep(2000);
137
             aktywny = false;
138
        }
139
141
        void odwolai() override
```

```
142
        {
143
             aktywny = false;
144
         }
145
         void wyswietl_pracownika() override
147
148
        std::cout « std::setw(10) « "Kelner | " « std::setw(13) « imie « " | " « std::setw(16) « nazwisko « " | " « std::setw(8) « aktywny « " | " « std::setw(15) « czas_na_ile_zajety « "\n";
149
150
151
         Pracownik* znajdz_wolnego_pracownika() override
155
156
157
             if (!aktywny)
158
159
                  return this;
160
161
             return nullptr:
        }
162
163
165
         void zajmij_wolnego_pracownika() override
166
167
             aktywny = true;
168
169 };
171 class Kuchnia
172 {
173 protected:
174
         std::vector<Pracownik*> pracownicy;
176 public:
180
         Kuchnia(std::string nazwa_pliku)
181
182
             std::ifstream plikwej(nazwa_pliku);
183
             if (!plikwej)
184
185
                  std::cerr « "Error: nie mozna otworzyc pliku :( " « nazwa_pliku « "\n";
186
187
             while (true)
188
189
                  std::string rola, imie, nazwisko;
190
                  plikwej » rola » imie » nazwisko;
191
                  if (!plikwej.fail())
192
193
194
                       if (rola == "kucharz")
195
196
                           Pracownik* pracownik = new Kucharz(imie, nazwisko);
197
                           pracownicy.push_back(pracownik);
198
                      else if (rola == "kelner")
199
200
                           Pracownik* pracownik = new Kelner(imie, nazwisko);
201
202
                           pracownicy.push_back(pracownik);
203
204
205
                  else
206
                      break;
207
208
         };
209
211
         void pokaz_sklad_kuchni()
212
             std::cout « "\nLista pracownikow:\n";
std::cout « std::setw(10) « "Rola | " « std::setw(13) « "Imie" « " | " « std::setw(16) «
213
214
        "Nazwisko" « " | " « std::setw(8) « "Aktywny | " « std::setw(15) « "Na ile zajety" « "\n";
215
             std::cout «"---
216
217
             for (int i = 0; i < pracownicy.size(); i++)</pre>
218
219
                  pracownicy[i]->wyswietl_pracownika();
220
221
             Sleep(3000);
222
223
227
         Pracownik* znajdz_wolnego_kelnera()
228
229
             for (int i = 0; i < pracownicy.size(); i++)</pre>
230
231
                  auto b = pracownicy[i]->znajdz_wolnego_pracownika();
232
                  if (b == nullptr)
                      continue;
233
                  Kelner* ptr = dynamic_cast<Kelner*>(b);
234
235
                  if (ptr)
236
                  {
237
                       zajmij_pracownika(b);
238
                      return b;
239
                  }
240
             }
```

```
std::cout « "\nBrak wolnego kelnera!!\nSprobuj ponownie za chwile\n";
            return nullptr;
243
2.4.4
       Pracownik* znajdz_wolnego_kucharza()
248
249
            for (int i = 0; i < pracownicy.size(); i++)</pre>
251
252
                auto b = pracownicy[i]->znajdz_wolnego_pracownika();
253
                if (b == nullptr)
254
                     continue;
                Kucharz* ptr = dynamic_cast<Kucharz*>(b);
255
256
                if (ptr)
257
258
                     zajmij_pracownika(b);
259
260
261
            std::cout « "\nBrak wolnego kucharza!!\nSprobuj ponownie za chwile\n";
262
263
            return nullptr;
264
265
269
        void zajmij_pracownika(Pracownik* pracownik)
270
            pracownik->zajmij_wolnego_pracownika();
272
273
275
        ~Kuchnia()
276
            for (int i = 0; i < pracownicy.size(); i++)</pre>
277
278
                delete(pracownicy[i]);
280
281
282 };
```

## 5.9 C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal\_gastronomiczny/Project1/stolik.h File Reference

```
#include <iostream>
#include <vector>
#include <fstream>
#include <string>
#include "funkcje.h"
#include "windows.h"
```

#### **Classes**

- · class Stolik
- · class ListaStolikow

#### 5.10 stolik.h

```
1
3  #pragma once
4  #include <iostream>
5  #include <vector>
6  #include <fstream>
7  #include <string>
8  #include "funkcje.h"
9  #include "windows.h"
10
12  class Stolik
13 {
```

```
14 protected:
15
      int numer;
16
      int max_ilosc_osob;
17
      int aktualni_uzytkownicy;
18
      bool wolny;
19 public:
      Stolik (int numer, int max_ilosc_osob) : numer(numer), max_ilosc_osob(max_ilosc_osob),
      aktualni_uzytkownicy(0), wolny(true) {}
25
29
      int dostep_do_maxa() { max_ilosc_osob; }
30
      int dostep_numer_stolika() { return numer; }
34
35
37
      void pokaz_stolik()
38
      39
40
42
      void zajmij()
43
44
          wolny = false;
45
      friend class ListaStolikow;
46
47
      friend int wyznacz_id_wolny_stolik_min_os(std::vector<Stolik>stoliki, std::vector<int> pozycje);
48 };
49
51 class ListaStolikow
52
53 protected:
54
      std::vector<Stolik>stoliki;
55 public:
59
      std::vector<Stolik> dostep_stoliki() { return stoliki; }
60
64
      ListaStolikow(std::string nazwa_pliku)
65
           std::ifstream plikwej(nazwa_pliku);
66
67
           if (!plikwej)
69
              std::cerr « "Error: nie mozna otworzyc pliku :( " « nazwa_pliku « "\n";
70
          while (true)
71
72
73
              int nr, ilosc;
              plikwej » nr » ilosc;
75
               if (!plikwej.fail())
76
                  Stolik stolik(nr, ilosc);
77
78
                  stoliki.push_back(stolik);
79
80
              else
81
                  break;
82
83
      } ;
84
      void wyswietl_stoliki()
86
88
           for (int i = 0; i < stoliki.size(); i++)</pre>
89
90
              stoliki[i].pokaz_stolik();
91
92
      };
93
      void zwolnij_stolik(int poz)
96
97
           stoliki[poz].wolny = true;
          stoliki[poz].aktualni_uzytkownicy = 0;
98
99
      }
100
105
        int szukaj_stolika(int ilosc_chetnych)
106
107
           bool tmp = false;
108
           int pozycja_zajeta=0;
            for (int i = 0; i < stoliki.size(); i++)</pre>
109
110
111
                if (stoliki[i].wolny == true && stoliki[i].max_ilosc_osob == ilosc_chetnych)
112
113
                    stoliki[i].zajmij();
114
                   stoliki[i].aktualni_uzytkownicy = ilosc_chetnych;
115
                   pozycja_zajeta = i;
116
                   tmp = true;
                   break;
117
118
119
120
           std::vector<int>nr_stolika_do_rezerwacji;
121
           nr_stolika_do_rezerwacji.resize(0);
122
           if (tmp == false)
```

```
124
                for (int i = 0; i < stoliki.size(); i++)</pre>
125
                     if (stoliki[i].wolny == true && stoliki[i].max_ilosc_osob >= ilosc_chetnych)
126
127
                         nr_stolika_do_rezerwacji.push_back(i);
128
129
130
131
                 if (!nr_stolika_do_rezerwacji.empty())
132
                     int poz = wyznacz_id_wolny_stolik_min_os(stoliki, nr_stolika_do_rezerwacji);
133
134
                     stoliki[poz].zajmij();
135
                     stoliki[poz].aktualni_uzytkownicy = ilosc_chetnych;
136
                    pozycja_zajeta = poz;
137
                     tmp = true;
138
139
         return pozycja_zajeta;
if (tmp == false)
140
141
143
                std::cout « "Nie udalo sie znalezc stolika :(\n";
           return -1:
144
           Sleep(2000);
145
146
147 };
```

# 5.11 C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal\_gastronomiczny/Project1/zamowienia.h File Reference

```
#include <iostream>
#include <vector>
#include <fstream>
#include <string>
#include <set>
#include "dania.h"
#include "memory>
#include "windows.h"
```

#### Classes

- · class Obserwator
- · class Zamowienie

#### 5.12 zamowienia.h

```
1
3 #pragma once
4 #include <iostream>
5 #include <vector>
6 #include <fstream>
7 #include <string>
8 #include <set>
9 #include "dania.h"
10 #include "mindows.h"
12
13
15 class Zamowienie;
16
18 class Obserwator {
19 public:
```

```
21
       virtual ~Obserwator() {}
26
       virtual void aktualizuj(Zamowienie& zamowienie) = 0;
2.7
       virtual void odwolaj() = 0;
31
32 };
33
35 class Zamowienie
36 {
37 protected:
38
       static int ctr;
       Stolik stolik:
39
40
       std::vector<Danie>zamowione_dania;
41
       Menu* menu;
42
       std::set<Obserwator*> obserwatorzy;
       void powiadom() {
   for (auto& o : obserwatorzy) {
4.5
46
               o->aktualizuj(*this);
47
48
49
       }
50
52
       void anuluj() {
         for (auto& o : obserwatorzy) {
5.3
54
                o->odwolaj();
55
56
57 public:
62
       Zamowienie (Menu* menu, Stolik stolik) : menu(menu), stolik (stolik) { ctr++; };
63
67
       std::vector<Danie>pobierz_zamowione() const { return zamowione_dania; }
68
72
       Stolik dostep_stolik() { return stolik; }
73
77
       void dopisz(Obserwator* obserwator) {
78
           obserwatorzy.insert(obserwator);
79
80
84
       void wypisz(Obserwator* obserwator) {
85
           obserwatorzy.erase(obserwator);
86
87
91
       bool zloz zamowienie()
92
93
           while (true)
           {
95
                pokaz_menu_dzialania_zamowienie();
                prezentuj_aktualne_zakupy();
std::cout « "\n\nTwoj wybor: ";
96
97
                char instruktator;
98
99
                std::cin » instruktator;
100
101
                 switch (instruktator)
102
                 case 'd':
103
104
                    for (;;)
105
                     {
106
                         menu->wyswietl_menu();
107
                         prezentuj_aktualne_zakupy();
                          std::cout « "\nDodaj danie do zamowienia: ";
108
109
                         static int numer;
110
                         std::cin » numer;
                         if (numer > 0 && numer <= menu->lista_dan.size())
111
112
113
                              dodaj_danie_do_zamowienia(numer);
114
115
                         else if (numer > 0)
116
                              prezentuj_aktualne_zakupy();
117
118
                              menu->wyswietl_menu();
                              std::cout « "\nBrak potrawy o podanym numerze :(";
119
120
                              Sleep(2000);
121
122
                         else
123
                             break:
124
125
                     break;
126
                 case 'u':
127
                     for (;;)
128
129
                         menu->wyswietl menu();
                         prezentuj_aktualne_zakupy();
130
131
                          int numer;
132
                         std::cout « "Podaj pozycje dania do usuniecia: "; std::cin » numer;
133
                         if (numer > 0 && numer <= zamowione_dania.size())</pre>
134
135
                              zamowione_dania.erase(zamowione_dania.begin() + numer - 1);
136
                              prezentuj_aktualne_zakupy();
```

5.12 zamowienia.h 49

```
137
138
                         else if (numer > 0)
139
140
                              prezentuj_aktualne_zakupy();
                             menu->wyswietl_menu();
std::cout « "\nBrak potrawy o podanym numerze :(";
141
142
143
                              Sleep(2000);
144
145
146
                              break;
147
                     }
148
                     break:
                 case 'v':
149
150
                     if (!zamowione_dania.empty())
151
152
                         std::cout « "Twoje zamowienie zostalo dodane do kolejki w kuchni\n";
                         powiadom();
153
154
155
                     else
156
                     {
157
                         std::cout « "\nTwoja lista zakupow jest pusta, zamowienie nie zostalo
       zrealizowane \n";
158
                         anului();
159
160
                     Sleep(3000);
                     system("cls");
161
                     return true;
162
163
                 case 'x':
164
                     anuluj();
165
                     return false;
166
167
168
169
170
        }
171
175
        void dodaj danie do zamowienia (int poz)
176
177
             zamowione_dania.push_back(menu->lista_dan[poz-1]);
178
             pokaz_zamowione_rzeczy();
179
             oblicz_czas_przygotowania();
180
            oblicz_koszt();
181
        }
182
186
        double oblicz_koszt()
187
188
             double koszt = 0;
             for (int i = 0; i < zamowione_dania.size(); i++)</pre>
189
190
191
                 koszt += zamowione dania[i].cena;
192
193
             std::cout « "\nKoszt twojego zamowienia: " « koszt«"zl\n";
194
             return koszt;
195
        }
196
200
        double oblicz czas przygotowania()
201
202
             double czas = 0;
203
             for (int i = 0; i < zamowione_dania.size(); i++)</pre>
204
205
                 czas += zamowione_dania[i].czas_przygotwania;
206
207
            std::cout « "\nCzas oczekiwania na twoje zamowienie: " « czas«"sec";
208
            return czas;
209
        }
210
212
        void pokaz_zamowione_rzeczy()
213
214
             std::cout « "\nTwoje zamowienie:\n";
             for (int i = 0; i < zamowione_dania.size(); i++)</pre>
215
216
                 std::cout « i + 1 « ". " « zamowione_dania[i].nazwa«"\n";
217
218
            }
219
221
        void prezentuj_aktualne_zakupy()
222
223
             std::cout « "\n\nZamowienie nr: "«ctr«" dla stolika nr : " « stolik.dostep_numer_stolika()«"\n";
224
             pokaz_zamowione_rzeczy();
225
             oblicz_czas_przygotowania();
226
            oblicz_koszt();
227
228 };
```

# 5.13 C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-gr11-repo/Projekt/Lokal\_gastronomiczny/Project1/zrealizowane\_ zamowienie.h File Reference

```
#include <iostream>
#include <vector>
#include <fstream>
#include <string>
#include "funkcje.h"
#include "dania.h"
#include "zamowienia.h"
#include <map>
```

#### **Classes**

· class Zrealizowane\_zamowienie

#### 5.14 zrealizowane zamowienie.h

```
3 #pragma once
4 #include <iostream>
5 #include <vector>
6 #include <fstream>
7 #include <string>
8 #include "funkcje.h"
9 #include "dania.h"
10 #include "zamowienia.h"
11 #include <map>
14 class Zrealizowane_zamowienie :public Obserwator
16 protected:
       std::map<Danie, int> ilosc;
17
        static std::vector<Zamowienie>zamowienia;
18
20 public:
        void aktualizuj(Zamowienie& zam) override
26
             auto zamowione = zam.pobierz_zamowione();
for (int i = 0; i < zamowione.size(); i++)</pre>
2.7
2.8
29
30
                  ilosc[zamowione[i]]++;
32
33
        void odwolaj() override {}
35
36
        void pokaz_statystyki()
38
             std::cout « "Statystyki:\n\n";
41
42
             for (auto &wpis : ilosc)
43
                  std::cout « wpis.first.nazwa « " " « wpis.second«"\n";
             Sleep(2000);
47
        friend class Zamowienie;
48
50 };
```

### Index

```
\simKuchnia
                                                      Danie, 7
     Kuchnia, 16
                                                          cena, 10
\simObserwator
                                                          czas przygotwania, 10
    Obserwator, 22
                                                          Danie, 7
                                                          nazwa, 10
aktualizuj
                                                          operator << , 9
    Kelner, 12
                                                          operator<=>, 9
    Kucharz, 14
                                                          pokaz_czas, 9
    Obserwator, 22
                                                          wielkosc_porcji, 10
     Pracownik, 24
                                                          Zamowienie, 9
    Zrealizowane_zamowienie, 35
                                                          Zrealizowane zamowienie, 9
aktualni uzytkownicy
                                                      dodaj danie do zamowienia
     Stolik, 29
                                                          Zamowienie, 31
aktywny
                                                      dopisz
    Pracownik, 26
                                                          Zamowienie, 31
anuluj
                                                      dostep_aktywny
    Zamowienie, 31
                                                          Pracownik, 24
                                                      dostep_do_maxa
C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-Stolik, 27
         gr11-repo/Projekt/Lokal_gastronomiczny/Project1/dapia.hmie
                                                           Pracownik, 25
gr11-repo/Projekt/Lokal_gastronomiczny/Project1/funkcie cophik, 25
38 dostep_numer_stolika C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-Stolik, 28
         gr11-repo/Projekt/Lokal_gastronomiczny/Project1/funkcjestolik
                                                          Zamowienie, 31
C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-p_stoliki
         gr11-repo/Projekt/Lokal_gastronomiczny/Project1/main_cpa_tolikow, 18
C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4efl/Akcje.cpp
         gr11-repo/Projekt/Lokal_gastronomiczny/Project1/praepwraewraewrae
                                                          pokaz menu dzialania wstep, 39
C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-pokaz_menu_dzialania_zamowienie, 39
         gr11-repo/Projekt/Lokal_gastronomiczny/Project1/stolikybnacz_id_wolny_stolik_min_os, 39
                                                      funkcje.h
C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-konwerter_ze_stringa, 40
         gr11-repo/Projekt/Lokal_gastronomiczny/Project1/zamenu_dzialania_wstep, 40
         47
                                                          pokaz_menu_dzialania_zamowienie, 40
C:/Users/pawel/source/repos/polsl-aei-pk2-katowice/76b4e121-wyznacz id wolny_stolik_min_os, 40
         gr11-repo/Projekt/Lokal_gastronomiczny/Project1/zrealizowane_zamowienie.h,
         50
                                                      ilosc
cena
                                                          Zrealizowane zamowienie, 36
     Danie, 10
                                                      imie
                                                          Pracownik, 26
ctr
    Zamowienie, 33
                                                      Kelner, 11
czas na ile zajety
                                                          aktualizuj, 12
     Pracownik, 26
                                                          Kelner, 11
czas_przygotwania
                                                          odwolai, 12
     Danie, 10
```

52 INDEX

wyswietl_pracownika, 12	
wyswieti_pracownika, 12	odwolaj
zajmij wolnego pracownika, 12	Kelner, 12
znajdz_wolnego_pracownika, 12	Kucharz, 14
konwerter_ze_stringa	Obserwator, 22
funkcje.h, 40	Zrealizowane_zamowienie, 36
Kucharz, 13	
	operator<<
aktualizuj, 14	Danie, 9
Kucharz, 14	funkcje.cpp, <mark>38</mark>
odwolaj, 14	operator<=>
wyswietl_pracownika, 14	Danie, 9
zajmij_wolnego_pracownika, 15	
znajdz_wolnego_pracownika, 15	pobierz_zamowione
Kuchnia, 15	Zamowienie, 32
∼Kuchnia, 16	pokaz_czas
Kuchnia, 16	Danie, 9
pokaz_sklad_kuchni, 16	pokaz_menu_dzialania_wstep
• – –	funkcje.cpp, 39
pracownicy, 17	- · · ·
zajmij_pracownika, 16	funkcje.h, 40
znajdz_wolnego_kelnera, 17	pokaz_menu_dzialania_zamowienie
znajdz_wolnego_kucharza, 17	funkcje.cpp, <mark>39</mark>
	funkcje.h, 40
lista_dan	pokaz_sklad_kuchni
Menu, 21	Kuchnia, 16
ListaStolikow, 17	pokaz statystyki
dostep_stoliki, 18	Zrealizowane_zamowienie, 36
ListaStolikow, 18	pokaz_stolik
	•
Stolik, 28	Stolik, 28
stoliki, 19	pokaz_zamowione_rzeczy
szukaj_stolika, 18	Zamowienie, 32
wyswietl_stoliki, 19	powiadom
zwolnij_stolik, 19	Zamowienie, 32
	pracownicy
max_ilosc_osob	Kuchnia, 17
Stolik, 29	Pracownik, 23
Ctomit, 20	aktualizuj, <mark>24</mark>
Menu, 20	
Menu, 20	
Menu, 20 lista_dan, 21	aktywny, <mark>26</mark>
Menu, 20 lista_dan, 21 Menu, 20	aktywny, 26 czas_na_ile_zajety, 26
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20 Zamowienie, 21	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24 dostep_imie, 25
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20 Zamowienie, 21 menu	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24 dostep_imie, 25 dostep_nazwisko, 25
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20 Zamowienie, 21	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24 dostep_imie, 25
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20 Zamowienie, 21 menu Zamowienie, 33	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24 dostep_imie, 25 dostep_nazwisko, 25
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20 Zamowienie, 21 menu Zamowienie, 33	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24 dostep_imie, 25 dostep_nazwisko, 25 imie, 26
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20 Zamowienie, 21 menu Zamowienie, 33	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24 dostep_imie, 25 dostep_nazwisko, 25 imie, 26 nazwisko, 26 Pracownik, 24
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20 Zamowienie, 21 menu Zamowienie, 33	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24 dostep_imie, 25 dostep_nazwisko, 25 imie, 26 nazwisko, 26 Pracownik, 24 wyswietl_pracownika, 25
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20 Zamowienie, 21 menu Zamowienie, 33  nazwa Danie, 10	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24 dostep_imie, 25 dostep_nazwisko, 25 imie, 26 nazwisko, 26 Pracownik, 24 wyswietl_pracownika, 25 zajmij_wolnego_pracownika, 25
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20 Zamowienie, 21 menu Zamowienie, 33  nazwa Danie, 10 nazwisko	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24 dostep_imie, 25 dostep_nazwisko, 25 imie, 26 nazwisko, 26 Pracownik, 24 wyswietl_pracownika, 25 zajmij_wolnego_pracownika, 25 znajdz_wolnego_pracownika, 25
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20 Zamowienie, 21 menu Zamowienie, 33  nazwa Danie, 10 nazwisko Pracownik, 26	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24 dostep_imie, 25 dostep_nazwisko, 25 imie, 26 nazwisko, 26 Pracownik, 24 wyswietl_pracownika, 25 zajmij_wolnego_pracownika, 25 znajdz_wolnego_pracownika, 25 prezentuj_aktualne_zakupy
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20 Zamowienie, 21 menu Zamowienie, 33  nazwa Danie, 10 nazwisko Pracownik, 26 numer	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24 dostep_imie, 25 dostep_nazwisko, 25 imie, 26 nazwisko, 26 Pracownik, 24 wyswietl_pracownika, 25 zajmij_wolnego_pracownika, 25 znajdz_wolnego_pracownika, 25
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20 Zamowienie, 21 menu Zamowienie, 33  nazwa Danie, 10 nazwisko Pracownik, 26 numer Stolik, 29	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24 dostep_imie, 25 dostep_nazwisko, 25 imie, 26 nazwisko, 26 Pracownik, 24 wyswietl_pracownika, 25 zajmij_wolnego_pracownika, 25 znajdz_wolnego_pracownika, 25 prezentuj_aktualne_zakupy Zamowienie, 32
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20 Zamowienie, 21 menu Zamowienie, 33  nazwa Danie, 10 nazwisko Pracownik, 26 numer Stolik, 29  oblicz_czas_przygotowania	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24 dostep_imie, 25 dostep_nazwisko, 25 imie, 26 nazwisko, 26 Pracownik, 24 wyswietl_pracownika, 25 zajmij_wolnego_pracownika, 25 znajdz_wolnego_pracownika, 25 prezentuj_aktualne_zakupy Zamowienie, 32  Stolik, 26
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20 Zamowienie, 21 menu Zamowienie, 33  nazwa Danie, 10 nazwisko Pracownik, 26 numer Stolik, 29  oblicz_czas_przygotowania Zamowienie, 31	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24 dostep_imie, 25 dostep_nazwisko, 25 imie, 26 nazwisko, 26 Pracownik, 24 wyswietl_pracownika, 25 zajmij_wolnego_pracownika, 25 znajdz_wolnego_pracownika, 25 prezentuj_aktualne_zakupy Zamowienie, 32  Stolik, 26 aktualni_uzytkownicy, 29
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20 Zamowienie, 21 menu Zamowienie, 33  nazwa Danie, 10 nazwisko Pracownik, 26 numer Stolik, 29  oblicz_czas_przygotowania Zamowienie, 31 oblicz_koszt	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24 dostep_imie, 25 dostep_nazwisko, 25 imie, 26 nazwisko, 26 Pracownik, 24 wyswietl_pracownika, 25 zajmij_wolnego_pracownika, 25 znajdz_wolnego_pracownika, 25 prezentuj_aktualne_zakupy Zamowienie, 32  Stolik, 26 aktualni_uzytkownicy, 29 dostep_do_maxa, 27
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20 Zamowienie, 21 menu Zamowienie, 33  nazwa Danie, 10 nazwisko Pracownik, 26 numer Stolik, 29  oblicz_czas_przygotowania Zamowienie, 31 oblicz_koszt Zamowienie, 32	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24 dostep_imie, 25 dostep_nazwisko, 25 imie, 26 nazwisko, 26 Pracownik, 24 wyswietl_pracownika, 25 zajmij_wolnego_pracownika, 25 znajdz_wolnego_pracownika, 25 prezentuj_aktualne_zakupy Zamowienie, 32  Stolik, 26 aktualni_uzytkownicy, 29 dostep_do_maxa, 27 dostep_numer_stolika, 28
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20 Zamowienie, 21 menu Zamowienie, 33  nazwa Danie, 10 nazwisko Pracownik, 26 numer Stolik, 29  oblicz_czas_przygotowania Zamowienie, 31 oblicz_koszt Zamowienie, 32 Obserwator, 21	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24 dostep_imie, 25 dostep_nazwisko, 25 imie, 26 nazwisko, 26 Pracownik, 24 wyswietl_pracownika, 25 zajmij_wolnego_pracownika, 25 znajdz_wolnego_pracownika, 25 prezentuj_aktualne_zakupy Zamowienie, 32  Stolik, 26 aktualni_uzytkownicy, 29 dostep_do_maxa, 27 dostep_numer_stolika, 28 ListaStolikow, 28
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20 Zamowienie, 21 menu Zamowienie, 33  nazwa Danie, 10 nazwisko Pracownik, 26 numer Stolik, 29  oblicz_czas_przygotowania Zamowienie, 31  oblicz_koszt Zamowienie, 32  Obserwator, 21 ~Obserwator, 22	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24 dostep_imie, 25 dostep_nazwisko, 25 imie, 26 nazwisko, 26 Pracownik, 24 wyswietl_pracownika, 25 zajmij_wolnego_pracownika, 25 znajdz_wolnego_pracownika, 25 prezentuj_aktualne_zakupy Zamowienie, 32  Stolik, 26 aktualni_uzytkownicy, 29 dostep_do_maxa, 27 dostep_numer_stolika, 28
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20 Zamowienie, 21 menu Zamowienie, 33  nazwa Danie, 10 nazwisko Pracownik, 26 numer Stolik, 29  oblicz_czas_przygotowania Zamowienie, 31 oblicz_koszt Zamowienie, 32 Obserwator, 21 ~Obserwator, 22 aktualizuj, 22	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24 dostep_imie, 25 dostep_nazwisko, 25 imie, 26 nazwisko, 26 Pracownik, 24 wyswietl_pracownika, 25 zajmij_wolnego_pracownika, 25 znajdz_wolnego_pracownika, 25 prezentuj_aktualne_zakupy Zamowienie, 32  Stolik, 26 aktualni_uzytkownicy, 29 dostep_do_maxa, 27 dostep_numer_stolika, 28 ListaStolikow, 28
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20 Zamowienie, 21 menu Zamowienie, 33  nazwa Danie, 10 nazwisko Pracownik, 26 numer Stolik, 29  oblicz_czas_przygotowania Zamowienie, 31 oblicz_koszt Zamowienie, 32 Obserwator, 21 ~Obserwator, 22 aktualizuj, 22 odwolaj, 22	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24 dostep_imie, 25 dostep_nazwisko, 25 imie, 26 nazwisko, 26 Pracownik, 24 wyswietl_pracownika, 25 zajmij_wolnego_pracownika, 25 znajdz_wolnego_pracownika, 25 prezentuj_aktualne_zakupy Zamowienie, 32  Stolik, 26 aktualni_uzytkownicy, 29 dostep_do_maxa, 27 dostep_numer_stolika, 28 ListaStolikow, 28 max_ilosc_osob, 29
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20 Zamowienie, 21 menu Zamowienie, 33  nazwa Danie, 10 nazwisko Pracownik, 26 numer Stolik, 29  oblicz_czas_przygotowania Zamowienie, 31 oblicz_koszt Zamowienie, 32 Obserwator, 21 ~Obserwator, 22 aktualizuj, 22 odwolaj, 22 obserwatorzy	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24 dostep_imie, 25 dostep_nazwisko, 25 imie, 26 nazwisko, 26 Pracownik, 24 wyswietl_pracownika, 25 zajmij_wolnego_pracownika, 25 znajdz_wolnego_pracownika, 25 prezentuj_aktualne_zakupy Zamowienie, 32  Stolik, 26 aktualni_uzytkownicy, 29 dostep_do_maxa, 27 dostep_numer_stolika, 28 ListaStolikow, 28 max_ilosc_osob, 29 numer, 29 pokaz_stolik, 28
Menu, 20 lista_dan, 21 Menu, 20 wyswietl_menu, 20 Zamowienie, 21 menu Zamowienie, 33  nazwa Danie, 10 nazwisko Pracownik, 26 numer Stolik, 29  oblicz_czas_przygotowania Zamowienie, 31 oblicz_koszt Zamowienie, 32 Obserwator, 21 ~Obserwator, 22 aktualizuj, 22 odwolaj, 22	aktywny, 26 czas_na_ile_zajety, 26 dostep_aktywny, 24 dostep_imie, 25 dostep_nazwisko, 25 imie, 26 nazwisko, 26 Pracownik, 24 wyswietl_pracownika, 25 zajmij_wolnego_pracownika, 25 znajdz_wolnego_pracownika, 25 prezentuj_aktualne_zakupy Zamowienie, 32  Stolik, 26 aktualni_uzytkownicy, 29 dostep_do_maxa, 27 dostep_numer_stolika, 28 ListaStolikow, 28 max_ilosc_osob, 29 numer, 29

INDEX 53

wyznacz_id_wolny_stolik_min_os, 28 zajmij, 28 stolik Zamowienie, 33 stoliki ListaStolikow, 19 szukaj_stolika	Zrealizowane_zamowienie, 36 zamowione_dania Zamowienie, 33 zloz_zamowienie Zamowienie, 33 znajdz_wolnego_kelnera Kuchnia, 17
ListaStolikow, 18	znajdz_wolnego_kucharza Kuchnia, 17
wielkosc_porcji Danie, 10 wolny Stolik, 29	znajdz_wolnego_pracownika Kelner, 12 Kucharz, 15 Pracownik, 25
wypisz Zamowienie, 32 wyswietl_menu Menu, 20 wyswietl_pracownika	Zrealizowane_zamowienie, 34 aktualizuj, 35 Danie, 9 ilosc, 36
Kelner, 12 Kucharz, 14 Pracownik, 25 wyswietl stoliki	odwolaj, 36 pokaz_statystyki, 36 zamowienia, 36 Zamowienie, 36 zwolnij_stolik
ListaStolikow, 19 wyznacz_id_wolny_stolik_min_os funkcje.cpp, 39 funkcje.h, 40 Stolik, 28	ListaStolikow, 19
zajmij Stolik, 28	
zajmij_pracownika Kuchnia, 16	
zajmij_wolnego_pracownika Kelner, 12 Kucharz, 15	
Pracownik, 25 zamowienia	
Zrealizowane_zamowienie, 36 Zamowienie, 29	
anuluj, 31 ctr, 33 Danie, 9	
dodaj_danie_do_zamowienia, 31 dopisz, 31 dostep_stolik, 31	
Menu, 21 menu, 33	
oblicz_czas_przygotowania, 31 oblicz_koszt, 32 obserwatorzy, 33	
pobierz_zamowione, 32 pokaz_zamowione_rzeczy, 32 powiadom, 32	
prezentuj_aktualne_zakupy, 32 stolik, 33	
wypisz, 32 Zamowienie, 30 zamowione_dania, 33	
zloz zamowienie, 33	