

Rezerwacje stolikow w restauracji

Generated by Doxygen 1.9.6



<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 File Index</b>	<b>5</b>
3.1 File List	5
<b>4 Class Documentation</b>	<b>7</b>
4.1 Date Class Reference	7
4.1.1 Detailed Description	8
4.1.2 Constructor & Destructor Documentation	8
4.1.2.1 Date() [1/2]	8
4.1.2.2 Date() [2/2]	8
4.1.3 Member Function Documentation	8
4.1.3.1 display()	8
4.1.3.2 operator==( )	9
4.2 List_res Class Reference	9
4.2.1 Detailed Description	10
4.2.2 Constructor & Destructor Documentation	10
4.2.2.1 List_res() [1/2]	10
4.2.2.2 List_res() [2/2]	10
4.2.3 Member Function Documentation	10
4.2.3.1 addtolist()	10
4.2.3.2 checkdate()	11
4.2.3.3 display()	11
4.2.3.4 getdate()	11
4.2.3.5 getIndexFromUser()	12
4.2.3.6 getReservationNumber()	12
4.2.3.7 getSeat()	12
4.2.3.8 load_from_file()	12
4.2.3.9 removeFromList()	13
4.2.3.10 savetofile()	13
4.2.4 Friends And Related Function Documentation	13
4.2.4.1 operator<<	13
4.3 List_table Class Reference	14
4.3.1 Detailed Description	15
4.3.2 Constructor & Destructor Documentation	15
4.3.2.1 List_table()	15
4.3.3 Member Function Documentation	15
4.3.3.1 check()	15
4.3.3.2 check2()	16

4.3.3.3 choose_table()	16
4.3.3.4 display_cities()	16
4.3.3.5 display_seats()	17
4.3.3.6 display_tables()	17
4.3.3.7 load_from_file()	17
4.3.4 Friends And Related Function Documentation	17
4.3.4.1 operator<< [1/2]	17
4.3.4.2 operator<< [2/2]	18
4.4 ListOrigin Class Reference	18
4.4.1 Detailed Description	18
4.5 Reservation Class Reference	19
4.5.1 Detailed Description	20
4.5.2 Constructor & Destructor Documentation	20
4.5.2.1 Reservation() [1/3]	20
4.5.2.2 Reservation() [2/3]	20
4.5.2.3 Reservation() [3/3]	21
4.5.3 Member Function Documentation	21
4.5.3.1 display()	21
4.5.4 Friends And Related Function Documentation	21
4.5.4.1 operator<< [1/2]	21
4.5.4.2 operator<< [2/2]	22
4.6 Table Class Reference	22
4.6.1 Constructor & Destructor Documentation	23
4.6.1.1 Table() [1/2]	23
4.6.1.2 Table() [2/2]	24
4.6.2 Member Function Documentation	24
4.6.2.1 display()	24
4.6.3 Friends And Related Function Documentation	24
4.6.3.1 operator<< [1/2]	24
4.6.3.2 operator<< [2/2]	25
4.7 TableOrigin Class Reference	25
4.7.1 Detailed Description	25
4.7.2 Member Function Documentation	25
4.7.2.1 display()	25
<b>5 File Documentation</b>	<b>27</b>
5.1 Date.h File Reference	27
5.2 Date.h	27
5.3 List_res.h File Reference	27
5.4 List_res.h	28
5.5 List_table.h File Reference	28
5.6 List_table.h	29

---

5.7 ListOrigin.h File Reference . . . . .	29
5.8 ListOrigin.h . . . . .	29
5.9 Reservation.h File Reference . . . . .	30
5.10 Reservation.h . . . . .	30
5.11 Table.h File Reference . . . . .	31
5.12 Table.h . . . . .	31
5.13 TableOrigin.h File Reference . . . . .	32
5.14 TableOrigin.h . . . . .	32
<b>Index</b>	<b>33</b>



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

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

ListOrigin . . . . .	18
List_res . . . . .	9
List_table . . . . .	14
TableOrigin . . . . .	25
Date . . . . .	7
Table . . . . .	22
Reservation . . . . .	19





## Chapter 2

# Class Index

### 2.1 Class List

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

<b>Date</b>	7
<b>List_res</b>	9
<b>List_table</b>	14
<b>ListOrigin</b>	18
<b>Reservation</b>	19
<b>Table</b>	22
<b>TableOrigin</b>	25



## Chapter 3

# File Index

### 3.1 File List

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

<b>Date.h</b>	27
<b>List_res.h</b>	27
<b>List_table.h</b>	28
<b>ListOrigin.h</b>	29
<b>Reservation.h</b>	30
<b>Table.h</b>	31
<b>TableOrigin.h</b>	32



## Chapter 4

# Class Documentation

### 4.1 Date Class Reference

```
#include <Date.h>
```

Inheritance diagram for Date:

Collaboration diagram for Date:

#### Public Member Functions

- void **display** ()
- bool **operator==** (const **Date** &d)
- **Date** (int &y, int &m, int &d, int &h, int &min)
- **Date** ()
  
- virtual void **display** ()=0

#### Public Attributes

- int **year**  
*Zmienna przechowująca rok rezerwacji stolika.*
- int **month**  
*Zmienna przechowująca miesiąc rezerwacji stolika.*
- int **day**  
*Zmienna przechowująca dzień rezerwacji stolika.*
- int **hour**  
*Zmienna przechowująca godzinę rezerwacji stolika.*
- int **minute**  
*Zmienna przechowująca minutę rezerwacji stolika.*

#### Friends

- class **List\_res**
- class **Reservation**

### 4.1.1 Detailed Description

Klasa **Date** (p. 7) zawierajaca dane dotyczace daty zarezerwacji

### 4.1.2 Constructor & Destructor Documentation

#### 4.1.2.1 Date() [1/2]

```
Date::Date (
    int & y,
    int & m,
    int & d,
    int & h,
    int & min )
```

Konstruktor piecioargumentowy

##### Parameters

<i>y</i>	Rok rezerwacji stolika
<i>d</i>	Dzien rezerwacji stolika
<i>h</i>	Godzina rezerwacji stolika
<i>min</i>	Minuta rezerwacji stolika

#### 4.1.2.2 Date() [2/2]

```
Date::Date ( )
```

Konstruktor bezargumentowy

### 4.1.3 Member Function Documentation

#### 4.1.3.1 display()

```
void Date::display ( ) [virtual]
```

Metoda sluzaca do wyswietlania daty

Implements **TableOrigin** (p. 25).

## 4.1.3.2 operator==( )

```
bool Date::operator== (
    const Date & d )
```

Przeciazony operator porownania

## Parameters

<i>d</i>	Obiekt porownywany
----------	--------------------

## Returns

Wartosc True gdy obiekty maja taki sam dzien, miesiac i rok oraz roznica miedzy godzinami wynosi mniej niz 120 minut, w innym wypadku False

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

- **Date.h**
- Date.cpp

## 4.2 List\_res Class Reference

```
#include <List_res.h>
```

Inheritance diagram for List\_res:

Collaboration diagram for List\_res:

## Public Member Functions

- int **getReservationNumber** ()
- void **load\_from\_file** (const string &name)
- bool **checkdate** (string &city, string &stringdate)
- void **getSeat** (string &chosenSeat)
- void **addtolist** ( Reservation &res)
- bool **removeFromList** (int primaryKey)
- void **display** ()
- void **savetofile** (const string &name)
- Date **getdate** (string &stringdate)
- List\_res (smart2 &resptr)
- List\_res ()
- int **getIndexFromUser** ()

## Public Member Functions inherited from ListOrigin

- virtual void **load\_from\_file** (const std::string &name)=0

## Friends

- class **List\_table**
- ostream & **operator**<< (ostream &s, **List\_res** &list)

### 4.2.1 Detailed Description

Klasa bedaca lista rezerwacji wczytanych z pliku i wprowadzonych przez uzytkownika

### 4.2.2 Constructor & Destructor Documentation

#### 4.2.2.1 List\_res() [1/2]

```
List_res::List_res (
    smart2 & resptr )
```

Konstruktor jednoargumentowy

##### Parameters

<i>resptr</i>	Wskaźnik na obiekt typu <b>Reservation</b> (p. 19)
---------------	--

#### 4.2.2.2 List\_res() [2/2]

```
List_res::List_res ( )
```

Konstruktor bezargumentowy

### 4.2.3 Member Function Documentation

#### 4.2.3.1 addtolist()

```
void List_res::addtolist (
    Reservation & res )
```

Metoda dodajaca rezerwacje do listy



## Parameters

<i>res</i>	Rezerwacja ktora ma zostac dodana do listy
------------	--

#### 4.2.3.2 checkdate()

```
bool List_res::checkdate (
    string & city,
    string & stringdate )
```

Metoda sprawdzajaca poprawnosc wprowadzonej przez uzytkownika daty

## Parameters

<i>city</i>	Wybor miasta wprowadzony przez uzytkownika
<i>seats</i>	Liczba siedzen przy stoliku wprowadzona przez uzytkownika
<i>stringdate</i>	Wybor daty wprowadzony przez uzytkownika return True jesli data jest poprawna, w przeciwnym wypadku wartosc false

#### 4.2.3.3 display()

```
void List_res::display ( )
```

Metoda wypisujaca zawartosc listy

#### 4.2.3.4 getdate()

```
Date List_res::getdate (
    string & stringdate )
```

Metoda pobierajaca wartosc daty ze znakow wprowadzonych przez uzytkownika

## Parameters

<i>stringdate</i>	Wybor daty wprowadzony przez uzytkownika
-------------------	--

## Returns

Obiekt typu **Date** (p. 7) bedacy data o danych wprowadzonych przez uzytkownika

#### 4.2.3.5 getIndexFromUser()

```
int List_res::getIndexFromUser ( )
```

Metoda pobierająca numer rezerwacji do usunięcia i zamieniająca go z liczbą

##### Returns

Wybrany numer rezerwacji

#### 4.2.3.6 getReservationNumber()

```
int List_res::getReservationNumber ( )
```

Metoda pobierająca liczbę rezerwacji

##### Returns

Liczba rezerwacji

#### 4.2.3.7 getSeat()

```
void List_res::getSeat (
    string & chosenSeat )
```

Metoda pobierająca numery stolików do rezerwacji od użytkownika a następnie zamienia je w liczbę

##### Parameters

<i>chosenSeat</i>	Numer stolika wprowadzony przez użytkownika
-------------------	---

#### 4.2.3.8 load\_from\_file()

```
void List_res::load_from_file (
    const string & name )
```

Metoda wczytująca z pliku dokonane rezerwacje

##### Parameters

<i>name</i>	Nazwa pliku z którego mają być wczytane rezerwacje
-------------	--

#### 4.2.3.9 removeFromList()

```
bool List_res::removeFromList (
    int primaryKey )
```

Metoda usuwajaca rezerwacje z listy

##### Parameters

<i>primaryKey</i>	Numer rezerwacji do usuniecia wprowadzony przez uzytkownika
-------------------	---

##### Returns

True gdy rezerwacja zostanie poprawnie usunieta, przeciwnym wypadku False

#### 4.2.3.10 savetofile()

```
void List_res::savetofile (
    const string & name )
```

Metoda zapisujaca liste rezerwacji do pliku

##### Parameters

<i>name</i>	Nazwa pliku do ktorego maja zostac zapisane rezerwacje
-------------	--

### 4.2.4 Friends And Related Function Documentation

#### 4.2.4.1 operator<<

```
ostream & operator<< (
    ostream & s,
    List_res & list ) [friend]
```

Operator strumieniowy wyjscia wypisujacy zawartosc listy

##### Parameters

<i>s</i>	Operator strumieniowy wyjscia
<i>list</i>	Lista do wypisania

#### Returns

Operator strumieniowy wyjscia

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

- **List\_res.h**
- List\_res.cpp

## 4.3 List\_table Class Reference

```
#include <List_table.h>
```

Inheritance diagram for List\_table:

Collaboration diagram for List\_table:

### Public Member Functions

- void **load\_from\_file** (const string &name)
- void **display\_cities** ()
- void **display\_seats** (string &city)
- void **display\_tables** (string &city)
- vector< **Table** > \* **choose\_table** ( **Date** &dat, string &city, string &seats, **List\_res** &thelist)
- bool **check** (string &city)
- bool **check2** (string &city, string &seats)
- **List\_table** (smart &tableptr)

### Public Member Functions inherited from List\_res

- int **getReservationNumber** ()
- void **load\_from\_file** (const string &name)
- bool **checkdate** (string &city, string &stringdate)
- void **getSeat** (string &chosenSeat)
- void **addtolist** ( **Reservation** &res)
- bool **removeFromList** (int primaryKey)
- void **display** ()
- void **savetofile** (const string &name)
- **Date** **getdate** (string &stringdate)
- **List\_res** (smart2 &resptr)
- **List\_res** ()
- int **getIndexFromUser** ()

### Public Member Functions inherited from ListOrigin

- virtual void **load\_from\_file** (const std::string &name)=0

## Friends

- ostream & **operator**<< (ostream &s, set< string > c)
- ostream & **operator**<< (ostream &s, **List\_table** &list)

### 4.3.1 Detailed Description

Klasa bedaca lista stolikow wczytanych z pliku

### 4.3.2 Constructor & Destructor Documentation

#### 4.3.2.1 List\_table()

```
List_table::List_table (
    smart & tableptr )
```

Konstruktor jednoargumentowy

##### Parameters

<i>tableptr</i>	Wskaźnik na obiekt typu <b>Table</b> (p. 22)
-----------------	--

### 4.3.3 Member Function Documentation

#### 4.3.3.1 check()

```
bool List_table::check (
    string & city )
```

Metoda sprawdzająca poprawność wprowadzonego miasta

##### Parameters

<i>city</i>	Ciąg znaków wprowadzony od użytkownika
-------------	--

##### Returns

True gdy miasto zostało wprowadzone poprawnie, w przeciwnym wypadku wartość False

#### 4.3.3.2 check2()

```
bool List_table::check2 (
    string & city,
    string & seats )
```

Metoda sprawdzająca czy ilość wprowadzonych miejsc przy stoliku jest poprawna

##### Parameters

<i>city</i>	Wybor miasta wprowadzony przez uzytkownika
<i>seats</i>	Ciag znakow wprowadzony przez uzytkownika

##### Returns

True gdy liczba miejsc została wprowadzona porownie, w przeciwnym wypadku wartosc False

#### 4.3.3.3 choose\_table()

```
vector< Table > * List_table::choose_table (
    Date & dat,
    string & city,
    string & seats,
    List_res & thelist )
```

Metoda pozwalająca zarezerwować dowolną liczbę stolików w danym mieście

##### Parameters

<i>dat</i>	Data wprowadzona przez uzytkownika
<i>city</i>	Miasto wprowadzone przez uzytkownika
<i>seats</i>	Liczba siedzen przy stoliku
<i>thelist</i>	Lista zarezerwowanych stolikow, sluzaca do porownania daty rezerwacji

##### Returns

Wektor przechowujący obiekty typu **Table** (p. 22) które zostały zarezerwowane

#### 4.3.3.4 display\_cities()

```
void List_table::display_cities ( )
```

Metoda wyświetlająca miasta

#### 4.3.3.5 display\_seats()

```
void List_table::display_seats (
    string & city )
```

Metoda wyswietlajaca ilosc siedzen przy stolikach, dostepnych w danym miescie

##### Parameters

<i>city</i>	Miasto, ktorego stoliki sa wyswietlane
-------------	--

#### 4.3.3.6 display\_tables()

```
void List_table::display_tables (
    string & city )
```

Metoda wyswietlajace wszystkie stoliki dostepne do zarezerwowania w danym miescie

##### Parameters

<i>city</i>	Miasto dla ktorego stoliki sa wyswietlane
-------------	---

#### 4.3.3.7 load\_from\_file()

```
void List_table::load_from_file (
    const string & name )
```

Metoda wczytujaca dane dotyczace stolikow z pliku

##### Parameters

<i>name</i>	Nazwa pliku do wczytania
-------------	--------------------------

### 4.3.4 Friends And Related Function Documentation

#### 4.3.4.1 operator<< [1/2]

```
ostream & operator<< (
    ostream & s,
    List_table & list ) [friend]
```

Operator strumieniowy wyjscia sluzacy do wypisania zawartosci listy (stolikow wczytanych)

**Parameters**

<b>s</b>	Operator strumieniowy wyjscia @list Lista stolikow wczytanych z pliku
----------	---

**Returns**

Operator strumieniowy wyjscia

**4.3.4.2 operator<< [2/2]**

```
ostream & operator<< (
    ostream & s,
    set< string > c ) [friend]
```

Operator strumieniowy wyjscia sluzacy do wypisania zawartosci zbioru miast

**Parameters**

<b>s</b>	Operator strumieniowy wyjscia
<b>c</b>	Zbior miast

**Returns**

Operator strumieniowy wyjscia

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

- **List\_table.h**
- List\_table.cpp

**4.4 ListOrigin Class Reference**

```
#include <ListOrigin.h>
```

Inheritance diagram for ListOrigin:

**Public Member Functions**

- virtual void **load\_from\_file** (const std::string &name)=0

**4.4.1 Detailed Description**

Klasa bazowa dla klasy **List\_res** (p.9), zawierajaca wirtualna metode load\_from\_file sluzaca do wczytywania danych z pliku

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

- **ListOrigin.h**



## 4.5 Reservation Class Reference

```
#include <Reservation.h>
```

Inheritance diagram for Reservation:

Collaboration diagram for Reservation:

### Public Member Functions

- **Reservation** (string &c, int &n, int &s, string &resname, string &com, **Date** &dat, smart2 next2)
- **Reservation** (string &c, int &n, int &s, string &resname, string &com, **Date** &dat)
- **Reservation** (const **Reservation** &r)
- void **display** ()

### Public Member Functions inherited from Table

- **Table** (string &c, int &n, int &s, **smart** &next)
- **Table** (string &c, int &n, int &s)
- **Table** ()
- void **display** ()
- virtual void **display** ()=0

### Friends

- class **List\_res**
- class **List\_table**
- ostream & **operator**<< (ostream &s, **List\_res** &list)
- ostream & **operator**<< (ostream &s, **Reservation** &res)

### Additional Inherited Members

#### Public Types inherited from Table

- typedef shared\_ptr< **Table** > **smart**  
*Typedef dla wskaźnika na kolejny obiekt typu **Reservation** (p. 19).*

#### Public Attributes inherited from Table

- string **city**  
*Zmienna przechowująca miasto w którym znajduje się stolik.*
- int **number**  
*Zmienna przechowująca numer stolika.*
- int **seats**  
*Zmienna przechowująca liczbę miejsc przy stoliku.*

### 4.5.1 Detailed Description

Klasa bedaca rezerwacja stolika dokonana przez uzytkownika

### 4.5.2 Constructor & Destructor Documentation

#### 4.5.2.1 Reservation() [1/3]

```
Reservation::Reservation (
    string & c,
    int & n,
    int & s,
    string & resname,
    string & com,
    Date & dat,
    smart2 next2 )
```

Konstruktor siedmioargumentowy

##### Parameters

<i>c</i>	Miasto
<i>n</i>	Numer stolika
<i>resname</i>	Nazwisko rezerwacji
<i>com</i>	Dodatkowe uwagi
<i>dat</i>	Data rezerwacji
<i>next2</i>	Wskaźnik na następny obiekt

#### 4.5.2.2 Reservation() [2/3]

```
Reservation::Reservation (
    string & c,
    int & n,
    int & s,
    string & resname,
    string & com,
    Date & dat )
```

Konstruktor szescioargumentowy

##### Parameters

<i>c</i>	Miasto
<i>n</i>	Numer stolika
<i>resname</i>	Nazwisko rezerwacji
<i>com</i>	Dodatkowe uwagi
<i>dat</i>	Data rezerwacji

### 4.5.2.3 Reservation() [3/3]

```
Reservation::Reservation (
    const Reservation & r )
```

Konstruktor kopiujacy

#### Parameters

<i>r</i>	Obiekt ktorego dane sa kopiowane
----------	----------------------------------

## 4.5.3 Member Function Documentation

### 4.5.3.1 display()

```
void Reservation::display ( ) [virtual]
```

Metoda wypisujaca dane klasy

Implements **TableOrigin** (p. 25).

## 4.5.4 Friends And Related Function Documentation

### 4.5.4.1 operator<< [1/2]

```
ostream & operator<< (
    ostream & s,
    List_res & list ) [friend]
```

Operator strumieniowy wyjscia

#### Parameters

<i>s</i>	Operator strumieniowy wyjscia
<i>list</i>	Lista do wypisania

**Returns**

Operator strumieniowy wyjscia

Operator strumieniowy wyjscia wypisujacy zawartosc listy

**Parameters**

<i>s</i>	Operator strumieniowy wyjscia
<i>list</i>	Lista do wypisania

**Returns**

Operator strumieniowy wyjscia

**4.5.4.2 operator<< [2/2]**

```
ostream & operator<< (
    ostream & s,
    Reservation & res ) [friend]
```

Operator strumieniowy wyjscia

**Parameters**

<i>s</i>	Operator strumieniowy wyjscia
<i>res</i>	Rezerwacja do wypisania

**Returns**

Operator strumieniowy wyjscia

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

- **Reservation.h**
- Reservation.cpp

**4.6 Table Class Reference**

Inheritance diagram for Table:

Collaboration diagram for Table:

**Public Types**

- typedef shared\_ptr< **Table** > **smart**  
*Typydef dla wskaznika na kolejny obiekt typu **Reservation** (p. 19).*

## Public Member Functions

- **Table** (string &c, int &n, int &s, **smart** &next)
- **Table** (string &c, int &n, int &s)
- **Table** ()
- void **display** ()
  
- virtual void **display** ()=0

## Public Attributes

- string **city**  
*Zmienna przechowująca miasto w którym znajduje się stolik.*
- int **number**  
*Zmienna przechowująca numer stolika.*
- int **seats**  
*Zmienna przechowująca liczbę miejsc przy stoliku.*

## Friends

- class **List\_table**
- ostream & **operator**<< (ostream &s, **List\_table** &list)
- ostream & **operator**<< (ostream &s, **Table** &t)

### 4.6.1 Constructor & Destructor Documentation

#### 4.6.1.1 Table() [1/2]

```
Table::Table (
    string & c,
    int & n,
    int & s,
    smart & next )
```

Konstruktor czteroargumentowy

#### Parameters

<i>c</i>	Miasto
<i>n</i>	Numer stolika
<i>s</i>	Liczba miejsc przy stoliku
<i>next</i>	Wskaźnik na następny obiekt typu <b>Table</b> (p. 22)

#### 4.6.1.2 Table() [2/2]

```
Table::Table ( )
```

Konstruktor bezargumentowy

### 4.6.2 Member Function Documentation

#### 4.6.2.1 display()

```
void Table::display ( ) [virtual]
```

Metoda wyswietlajaca dane stolika

Implements **TableOrigin** (p. 25).

### 4.6.3 Friends And Related Function Documentation

#### 4.6.3.1 operator<< [1/2]

```
ostream & operator<< (
    ostream & s,
    List_table & list ) [friend]
```

Operator strumieniowy wyjscia

##### Parameters

<i>s</i>	Operator strumieniowy wyjscia
<i>list</i>	Lista stolikow wczytanych z pliku

##### Returns

Operator strumieniowy wyjscia

Operator strumieniowy wyjscia sluzacy do wypisania zawartosci listy (stolikow wczytanych)

##### Parameters

<i>s</i>	Operator strumieniowy wyjscia @list Lista stolikow wczytanych z pliku
----------	---

**Returns**

Operator strumieniowy wyjscia

**4.6.3.2 operator<< [2/2]**

```
ostream & operator<< (
    ostream & s,
    Table & t ) [friend]
```

Operator strumieniowy wyjscia

**Parameters**

<i>s</i>	operator strumieniowy wyjscia
<i>t</i>	Stolik ktorego dane zostana wypisane

**Returns**

Operator strumieniowy wyjscia

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

- **Table.h**
- Table.cpp

**4.7 TableOrigin Class Reference**

```
#include <TableOrigin.h>
```

Inheritance diagram for TableOrigin:

**Public Member Functions**

- virtual void **display** ()=0

**4.7.1 Detailed Description**

Klasa bazowa dla klasy **Table** (p. 22), zawierajaca wirtualna metode display, sluzaca do wyswietlania na ekranie

**4.7.2 Member Function Documentation****4.7.2.1 display()**

```
virtual void TableOrigin::display ( ) [pure virtual]
```

Implemented in **Date** (p. 8), **Reservation** (p. 21), and **Table** (p. 24).

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

- **TableOrigin.h**





## Chapter 5

# File Documentation

### 5.1 Date.h File Reference

```
#include "TableOrigin.h"
```

Include dependency graph for Date.h:

### 5.2 Date.h

**Go to the documentation of this file.**

```
00001
00002 #pragma once
00003 #ifndef DATE_H
00004 #define DATE_H
00005
00006 #include "TableOrigin.h"
00007
00009 class Date : public TableOrigin {
00010 public:
00011     int year;
00012
00013     int month;
00014
00015     int day;
00016
00017     int hour;
00018
00019     int minute;
00020
00021 public:
00022
00024     void display();
00025
00029     bool operator == (const Date& d);
00030
00036     Date(int& y, int& m, int& d, int& h, int& min);
00037
00039     Date();
00040
00041     friend class List_res;
00042
00043     friend class Reservation;
00044
00045 };
00046
00047 #endif
```

### 5.3 List\_res.h File Reference

```
#include "ListOrigin.h"
#include "Reservation.h"
```

Include dependency graph for List\_res.h: This graph shows which files directly or indirectly include this file:

## Classes

- class `List_res`

## 5.4 List\_res.h

Go to the documentation of this file.

```

00001
00002 #pragma once
00003 #ifndef LIST_RES_H
00004 #define LIST_RES_H
00005
00006 #include "ListOrigin.h"
00007 #include "Reservation.h"
00008 using namespace std;
00009
00011 class List_res : public ListOrigin
00012 {
00013 private:
00014     smart2 ptr;
00015     int reservation_number;
00016 public:
00017
00020     int getReservationNumber(); //
00021
00024     void load_from_file(const string& name);
00025
00031     bool checkdate(string& city, string& stringdate);
00032
00035     void getSeat(string& chosenSeat);
00036
00039     void addtolist(Reservation& res);
00040
00044     bool removeFromList(int primaryKey);
00045
00047     void display();
00048
00051     void savetofile(const string &name);
00052
00056     Date getdate(string& stringdate);
00057
00060     List_res(smart2& resptr);
00061
00063     List_res();
00064
00065     friend class List_table;
00066
00071     friend ostream& operator << (ostream& s, List_res& list);
00072
00075     int getIndexFromUser();
00076 };
00077 #endif

```

## 5.5 List\_table.h File Reference

```

#include "ListOrigin.h"
#include "List_res.h"
#include "Table.h"
#include <vector>
#include <set>

```

Include dependency graph for List\_table.h:

## Classes

- class `List_table`

## 5.6 List\_table.h

[Go to the documentation of this file.](#)

```

00001
00002 #pragma once
00003 #ifndef LIST_TABLE_H
00004 #define LIST_TABLE_H
00005
00006 #include "ListOrigin.h"
00007 #include "List_res.h"
00008 #include "Table.h"
00009 #include <vector>
00010 #include <set>
00011
00013 class List_table : public List_res
00014 {
00015 private:
00016     smart ptr;
00017     set <string> cities;
00018 public:
00019
00022     void load_from_file(const string& name);
00023
00025     void display_cities();
00026
00029     void display_seats(string& city);
00030
00033     void display_tables(string& city);
00034
00041     vector<Table>* choose_table(Date& dat, string& city, string& seats, List_res& thelist);
00042
00047     friend ostream& operator << (ostream& s, set<string> c);
00048
00053     friend ostream& operator << (ostream& s, List_table& list);
00054
00058     bool check(string& city);
00059
00064     bool check2(string& city, string& seats);
00065
00068     List_table(smart& tableptr);
00069 };
00070
00071 #endif

```

## 5.7 ListOrigin.h File Reference

```
#include <string>
```

Include dependency graph for ListOrigin.h: This graph shows which files directly or indirectly include this file:

### Classes

- class `ListOrigin`

## 5.8 ListOrigin.h

[Go to the documentation of this file.](#)

```

00001
00002 #pragma once
00003 #ifndef LIST_ORIGIN_H
00004 #define LIST_ORIGIN_H
00005
00006 #pragma once
00007 #include <string>
00008
00010 class ListOrigin
00011 {
00012 public:
00013
00014     virtual void load_from_file(const std::string &name) = 0;
00015 };
00016
00017 #endif

```

## 5.9 Reservation.h File Reference

```
#include "Table.h"
```

```
#include "Date.h"
```

Include dependency graph for Reservation.h: This graph shows which files directly or indirectly include this file:

### Classes

- class **Reservation**

### Typedefs

- typedef shared\_ptr< **Reservation** > smart2

## 5.10 Reservation.h

**Go to the documentation of this file.**

```
00001
00002 #pragma once
00003 #ifndef RESERVATION_H
00004 #define RESERVATION_H
00005
00006 #include "Table.h"
00007 #include "Date.h"
00008
00010 class Reservation : public Table
00011 {
00012 private:
00013
00014     static int counter;
00015     string name;
00016     string comments;
00017     Date reservdate;
00018
00019     typedef shared_ptr <Reservation> smart2;
00020     smart2 pointer2;
00021
00022     friend class List_res;
00023
00024 public:
00025
00033     Reservation(string& c, int& n, int& s, string& resname, string& com, Date& dat, smart2 next2);
00034
00041     Reservation(string& c, int& n, int& s, string& resname, string& com, Date& dat);
00042
00045     Reservation(const Reservation& r);
00046     ~Reservation();
00047
00049     void display();
00050
00051     friend class List_table;
00052
00057     friend ostream& operator << (ostream& s, List_res& list);
00058
00063     friend ostream& operator << (ostream& s, Reservation& res);
00064
00065 };
00066
00067 typedef shared_ptr <Reservation> smart2;
00068
00069
00070
00071 #endif
```

## 5.11 Table.h File Reference

```
#include <string>
#include <memory>
#include "TableOrigin.h"
```

Include dependency graph for Table.h: This graph shows which files directly or indirectly include this file:

### Classes

- class `Table`

### Typedefs

- typedef `shared_ptr< Table > smart`

## 5.12 Table.h

**Go to the documentation of this file.**

```
00001
00002 #pragma once
00003 #ifndef TABLE_H
00004 #define TABLE_H
00005
00006 #include <string>
00007 #include <memory>
00008 #include "TableOrigin.h"
00009
00010 using namespace std;
00011
00012
00013 class Table : public TableOrigin
00014 {
00015 public:
00016     string city;
00017     int number;
00018     int seats;
00019
00020     typedef shared_ptr <Table> smart;
00021
00022     Table(string& c, int& n, int& s, smart& next);
00023
00024     /**Konstruktor trojargumentowy
00025     @param c Miasto
00026     @param n Numer stolika
00027     @param s Liczba miejsc przy stoliku */
00028     Table(string& c, int& n, int& s);
00029
00030     Table();
00031
00032     void display();
00033
00034 private:
00035     smart pointer;
00036
00037     friend class List_table;
00038
00039     friend ostream& operator << (ostream& s, List_table& list);
00040
00041     friend ostream& operator << (ostream& s, Table& t);
00042 };
00043
00044 typedef shared_ptr <Table> smart;
00045 #endif
```

## 5.13 TableOrigin.h File Reference

This graph shows which files directly or indirectly include this file:

### Classes

- class `TableOrigin`

## 5.14 TableOrigin.h

**Go to the documentation of this file.**

```
00001
00002 #pragma once
00003 #ifndef TABLE_ORIGIN_H
00004 #define TABLE_ORIGIN_H
00005
00007 class TableOrigin
00008 {
00009 public:
00010     virtual void display() = 0;
00011 };
00012 #endif
```

# Index

addtolist  
    List\_res, 10

check  
    List\_table, 15

check2  
    List\_table, 15

checkdate  
    List\_res, 11

choose\_table  
    List\_table, 16

Date, 7  
    Date, 8  
    display, 8  
    operator==, 8

Date.h, 27

display  
    Date, 8  
    List\_res, 11  
    Reservation, 21  
    Table, 24  
    TableOrigin, 25

display\_cities  
    List\_table, 16

display\_seats  
    List\_table, 16

display\_tables  
    List\_table, 17

getdate  
    List\_res, 11

getIndexFromUser  
    List\_res, 11

getReservationNumber  
    List\_res, 12

getSeat  
    List\_res, 12

List\_res, 9  
    addtolist, 10  
    checkdate, 11  
    display, 11  
    getdate, 11  
    getIndexFromUser, 11  
    getReservationNumber, 12  
    getSeat, 12  
    List\_res, 10  
    load\_from\_file, 12  
    operator<<, 13  
    removeFromList, 13  
    savetofile, 13

List\_res.h, 27

List\_table, 14  
    check, 15  
    check2, 15  
    choose\_table, 16  
    display\_cities, 16  
    display\_seats, 16  
    display\_tables, 17  
    List\_table, 15  
    load\_from\_file, 17  
    operator<<, 17, 18

List\_table.h, 28

ListOrigin, 18

ListOrigin.h, 29

load\_from\_file  
    List\_res, 12  
    List\_table, 17

operator<<  
    List\_res, 13  
    List\_table, 17, 18  
    Reservation, 21, 22  
    Table, 24, 25

operator==  
    Date, 8

removeFromList  
    List\_res, 13

Reservation, 19  
    display, 21  
    operator<<, 21, 22  
    Reservation, 20, 21

Reservation.h, 30

savetofile  
    List\_res, 13

Table, 22  
    display, 24  
    operator<<, 24, 25  
    Table, 23

Table.h, 31

TableOrigin, 25  
    display, 25

TableOrigin.h, 32