

Delivery Service

Generated by Doxygen 1.9.8

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 abstract Class Reference	7
4.2 Client Class Reference	8
4.2.1 Constructor & Destructor Documentation	8
4.2.1.1 Client() [1/2]	8
4.2.1.2 Client() [2/2]	8
4.2.2 Member Function Documentation	9
4.2.2.1 getAddress()	9
4.2.2.2 getName()	9
4.2.2.3 getNumber()	9
4.2.2.4 placeAnOrder()	9
4.2.2.5 setAddress()	10
4.2.2.6 setName()	10
4.2.2.7 setNumber()	10
4.2.2.8 takeDish()	10
4.3 Cook Class Reference	11
4.3.1 Constructor & Destructor Documentation	11
4.3.1.1 Cook() [1/2]	11
4.3.1.2 Cook() [2/2]	11
4.3.2 Member Function Documentation	12
4.3.2.1 passOnDishes()	12
4.3.2.2 recieveAnOrder()	12
4.3.2.3 toWork()	12
4.4 Courier Class Reference	13
4.4.1 Constructor & Destructor Documentation	13
4.4.1.1 Courier() [1/2]	13
4.4.1.2 Courier() [2/2]	13
4.4.2 Member Function Documentation	14
4.4.2.1 putInDishes()	14
4.4.2.2 recieveAnOrder()	14
4.4.2.3 toWork()	14
4.5 DeliveryService Class Reference	15
4.5.1 Constructor & Destructor Documentation	15
4.5.1.1 DeliveryService() [1/2]	15

4.5.1.2 DeliveryService() [2/2]	15
4.5.2 Member Function Documentation	16
4.5.2.1 addAnEmployee()	16
4.5.2.2 addRestaurant()	16
4.5.2.3 getCouriers()	16
4.5.2.4 getName()	16
4.5.2.5 getRestaurants()	17
4.5.2.6 sentInRestaurant()	17
4.5.2.7 setName()	17
4.6 Dish Class Reference	17
4.6.1 Constructor & Destructor Documentation	18
4.6.1.1 Dish() [1/2]	18
4.6.1.2 Dish() [2/2]	18
4.6.2 Member Function Documentation	18
4.6.2.1 getName()	18
4.6.2.2 getPrice()	19
4.6.2.3 operator==()	19
4.6.2.4 setName()	19
4.6.2.5 setPrice()	19
4.7 DishDataBase Class Reference	21
4.7.1 Constructor & Destructor Documentation	21
4.7.1.1 DishDataBase()	21
4.7.2 Member Function Documentation	21
4.7.2.1 getRandomDish()	21
4.8 Menu Class Reference	22
4.8.1 Constructor & Destructor Documentation	22
4.8.1.1 Menu() [1/2]	22
4.8.1.2 Menu() [2/2]	22
4.8.2 Member Function Documentation	23
4.8.2.1 addDish() [1/2]	23
4.8.2.2 addDish() [2/2]	23
4.8.2.3 delDish() [1/3]	23
4.8.2.4 delDish() [2/3]	23
4.8.2.5 delDish() [3/3]	24
4.8.2.6 getDish()	24
4.8.2.7 show()	24
4.9 Order Class Reference	24
4.9.1 Constructor & Destructor Documentation	25
4.9.1.1 Order()	25
4.9.2 Member Function Documentation	25
4.9.2.1 addDish()	25
4.9.2.2 delDish()	25

4.9.2.3 getClient()	26
4.9.2.4 getDishes()	26
4.9.2.5 getNumber()	26
4.9.2.6 getStatus()	26
4.9.2.7 operator==()	26
4.9.2.8 setStatus()	27
4.10 Restaurant Class Reference	27
4.10.1 Constructor & Destructor Documentation	28
4.10.1.1 Restaurant() [1/2]	28
4.10.1.2 Restaurant() [2/2]	28
4.10.2 Member Function Documentation	28
4.10.2.1 addAnEmployee()	28
4.10.2.2 dishTransfer()	29
4.10.2.3 getCapital()	29
4.10.2.4 getCooks()	29
4.10.2.5 getMenu()	29
4.10.2.6 getName()	30
4.10.2.7 getRating()	30
4.10.2.8 orderDistribution()	30
4.10.2.9 revieveAnOrder()	30
4.10.2.10 setCapital()	30
4.10.2.11 setMenu()	31
4.10.2.12 setName()	31
4.10.2.13 setRating()	31
4.10.2.14 showInfo()	31
5 File Documentation	33
5.1 Client.cpp File Reference	33
5.1.1 Detailed Description	33
5.2 Client.h	33
5.3 Cook.cpp File Reference	34
5.3.1 Detailed Description	34
5.4 Cook.h	34
5.5 Courier.cpp File Reference	35
5.5.1 Detailed Description	35
5.6 Courier.h	35
5.7 Delivery.h	35
5.8 DeliveryService.cpp File Reference	36
5.8.1 Detailed Description	36
5.9 DeliveryService.h	36
5.10 Dish.cpp File Reference	37
5.10.1 Detailed Description	37

5.11 Dish.h	37
5.12 DishDB.h	37
5.13 Employee.cpp File Reference	38
5.13.1 Detailed Description	38
5.14 Employee.h	38
5.15 Menu.cpp File Reference	38
5.15.1 Detailed Description	39
5.16 Menu.h	39
5.17 Order.cpp File Reference	39
5.17.1 Detailed Description	39
5.18 Order.h	40
5.19 Restaurant.cpp File Reference	40
5.19.1 Detailed Description	40
5.20 Restaurant.h	41
Index	43

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

abstract	7
Client	8
DeliveryService	15
Dish	17
DishDataBase	21
Employee	
Cook	11
Courier	13
Menu	22
Order	24
Restaurant	27

Chapter 2

Class Index

2.1 Class List

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

abstract	7
Client	8
Cook	11
Courier	13
DeliveryService	15
Dish	17
DishDataBase	21
Menu	22
Order	24
Restaurant	27

Chapter 3

File Index

3.1 File List

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

Client.cpp	
File containing the implementation of the Client (p. 8) class	33
Client.h	33
Cook.cpp	
File containing the implementation of the Cook (p. 11) class	34
Cook.h	34
Courier.cpp	
File containing the implementation of the Courier (p. 13) class	35
Courier.h	35
Delivery.h	35
DeliveryService.cpp	
File containing the implementation of the DeliveryService (p. 15) class	36
DeliveryService.h	36
Dish.cpp	
File containing the implementation of the Dish (p. 17) class	37
Dish.h	37
DishDB.h	37
Employee.cpp	
File containing the implementation of the Employee class	38
Employee.h	38
Menu.cpp	
File containing the implementation of the Menu (p. 22) class	38
Menu.h	39
Order.cpp	
File containing the implementation of the Order (p. 24) class	39
Order.h	40
Restaurant.cpp	
File containing the implementation of the Restaurant (p. 27) class	40
Restaurant.h	41

Chapter 4

Class Documentation

4.1 abstract Class Reference

Public Member Functions

- **Employee** (const string &name, const string &position, const double &salary)
- string **getName** () const
- void **setName** (const string &name)
- string **getPosition** () const
- void **setPosition** (const string &position)
- double **getSalary** () const
- void **setSalary** (const double &salary)
- bool **getBusy** () const
- void **setBusy** (const int &busy)
- vector< **Order** > & **getOrders** ()
- virtual void **recieveAnOrder** (**Order** &order)=0
- virtual void **toWork** ()=0

Protected Attributes

- string **name_**
- string **position_**
- double **salary_**
- bool **busy_**
- vector< **Order** > **orders**

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

- Employee.h

4.2 Client Class Reference

Public Member Functions

- **Client** ()
*Default constructor of the **Client** (p. 8) class.*
- **Client** (const string &name, const string &address, const string &number)
*Constructor of the **Client** (p. 8) class.*
- string **getName** () const
Get the client's name.
- void **setName** (const string &name)
Set the client's name.
- string **getAddress** () const
Get the client's address.
- void **setAddress** (const string &address)
Set the client's address.
- string **getNumber** () const
Get the client's phone number.
- void **setNumber** (const string &number)
Set the client's phone number.
- void **takeDish** (const **Dish** &dish)
Take a dish.
- void **placeAnOrder** (**DeliveryService** &delServ, bool manual)
Place an order.

4.2.1 Constructor & Destructor Documentation

4.2.1.1 Client() [1/2]

```
Client::Client ( )
```

Default constructor of the **Client** (p. 8) class.

The default constructor initializes a **Client** (p. 8) object with a default name, address, and phone number.

4.2.1.2 Client() [2/2]

```
Client::Client (
    const string & name,
    const string & address,
    const string & number )
```

Constructor of the **Client** (p. 8) class.

The constructor initializes a **Client** (p. 8) object with a given name, address, and phone number.

Parameters

<i>name</i>	The client's name.
<i>address</i>	The client's address.
<i>number</i>	The client's phone number.

4.2.2 Member Function Documentation

4.2.2.1 getAddress()

```
string Client::getAddress ( ) const
```

Get the client's address.

Returns

The client's address.

4.2.2.2 getName()

```
string Client::getName ( ) const
```

Get the client's name.

Returns

The client's name.

4.2.2.3 getNumber()

```
string Client::getNumber ( ) const
```

Get the client's phone number.

Returns

The client's phone number.

4.2.2.4 placeAnOrder()

```
void Client::placeAnOrder (
    DeliveryService & delServ,
    bool manual )
```

Place an order.

Places an order with a delivery service. If the order is manual, the client is asked to choose a restaurant and dishes. If the order is not manual, a dish is chosen from the first restaurant's menu.

Parameters

<i>delServ</i>	The delivery service.
<i>manual</i>	Whether the order is manual.

4.2.2.5 setAddress()

```
void Client::setAddress (
    const string & address )
```

Set the client's address.

Parameters

<i>address</i>	The new address of the client.
----------------	--------------------------------

4.2.2.6 setName()

```
void Client::setName (
    const string & name )
```

Set the client's name.

Parameters

<i>name</i>	The new name of the client.
-------------	-----------------------------

4.2.2.7 setNumber()

```
void Client::setNumber (
    const string & number )
```

Set the client's phone number.

Parameters

<i>number</i>	The new phone number of the client.
---------------	-------------------------------------

4.2.2.8 takeDish()

```
void Client::takeDish (
    const Dish & dish )
```

Take a dish.

Adds a dish to the client's dishes.

Parameters

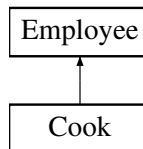
<i>dish</i>	The dish to take.
-------------	-------------------

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

- Client.h
- **Client.cpp**

4.3 Cook Class Reference

Inheritance diagram for Cook:



Public Member Functions

- **Cook** ()
*Constructor of the **Cook** (p. 11) class.*
- **Cook** (const string &name, const double &salary, const string &position)
*Constructor of the **Cook** (p. 11) class.*
- void **recieveAnOrder** (**Order** &order) override
Receive an order for preparation.
- void **toWork** () override
Start working on orders.
- vector< **Dish** > **passOnDishes** ()
Pass on prepared dishes.

4.3.1 Constructor & Destructor Documentation

4.3.1.1 Cook() [1/2]

```
Cook::Cook ( )
```

Constructor of the **Cook** (p. 11) class.

The default constructor initializes a **Cook** (p. 11) object with a given name, position, and salary.

4.3.1.2 Cook() [2/2]

```
Cook::Cook (
    const string & name,
    const double & salary,
    const string & position = "Cook" )
```

Constructor of the **Cook** (p. 11) class.

The constructor initializes a **Cook** (p. 11) object with a given name, position, and salary.

Parameters

<i>name</i>	The cook's name.
<i>salary</i>	The cook's salary.
<i>position</i>	The cook's position.

4.3.2 Member Function Documentation

4.3.2.1 passOnDishes()

```
vector< Dish > Cook::passOnDishes ( )
```

Pass on prepared dishes.

Returns a vector containing all prepared dishes.

Returns

A vector containing all prepared dishes.

4.3.2.2 recieveAnOrder()

```
void Cook::recieveAnOrder (
    Order & order ) [override]
```

Receive an order for preparation.

If the number of orders is less than the maximum, then the order is added to the list of orders. Otherwise, a message is displayed that the cook has received the maximum number of orders.

Parameters

<i>order</i>	The order to prepare.
--------------	-----------------------

4.3.2.3 toWork()

```
void Cook::toWork ( ) [override]
```

Start working on orders.

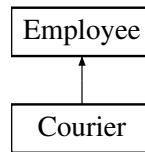
The cook starts preparing dishes for each order in the list of orders. After completing the preparation of each order, the order status is set to "ready", and a message is displayed about the order readiness. After completing work on all orders, the cook's busy status is set to "not busy".

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

- Cook.h
- **Cook.cpp**

4.4 Courier Class Reference

Inheritance diagram for Courier:



Public Member Functions

- **Courier** ()
*Constructor of the **Courier** (p. 13) class.*
- **Courier** (const string &name, const double &salary, const string &position)
*Constructor of the **Courier** (p. 13) class.*
- void **recieveAnOrder** (**Order** &order) override
Receive an order for delivery.
- void **putInDishes** (const vector< **Dish** > &preparedDishes)
Put dishes in the courier's bag.
- void **toWork** () override
Start working on delivering orders.

4.4.1 Constructor & Destructor Documentation

4.4.1.1 Courier() [1/2]

```
Courier::Courier ( )
```

Constructor of the **Courier** (p. 13) class.

The default constructor initializes a **Courier** (p. 13) object with a given name, position, and salary.

4.4.1.2 Courier() [2/2]

```
Courier::Courier (
    const string & name,
    const double & salary,
    const string & position = "Courier" )
```

Constructor of the **Courier** (p. 13) class.

The constructor initializes a **Courier** (p. 13) object with a given name, position, and salary.

Parameters

<i>name</i>	The courier's name.
<i>salary</i>	The courier's salary.
<i>position</i>	The courier's position.

4.4.2 Member Function Documentation

4.4.2.1 putInDishes()

```
void Courier::putInDishes (
    const vector< Dish > & preparedDishes )
```

Put dishes in the courier's bag.

Adds dishes to the courier's bag.

Parameters

<i>preparedDishes</i>	The dishes to add to the bag.
-----------------------	-------------------------------

4.4.2.2 recieveAnOrder()

```
void Courier::recieveAnOrder (
    Order & order ) [override]
```

Receive an order for delivery.

If the number of orders is less than the maximum and the number of dishes in the order does not exceed the remaining space in the bag, then the order is added to the list of orders. Otherwise, a message is displayed that the courier's bag is full or the courier has received the maximum number of orders.

Parameters

<i>order</i>	The order to deliver.
--------------	-----------------------

4.4.2.3 toWork()

```
void Courier::toWork ( ) [override]
```

Start working on delivering orders.

The courier starts delivering dishes for each order in the list of orders. After delivering each dish, the order status is set to "delivered", and a message is displayed about the order delivery. After delivering all dishes for all orders, the courier's busy status is set to "not busy".

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

- Courier.h
- **Courier.cpp**

4.5 DeliveryService Class Reference

Public Member Functions

- **DeliveryService** ()
*Default constructor of the **DeliveryService** (p. 15) class.*
- **DeliveryService** (const string &name)
*Constructor of the **DeliveryService** (p. 15) class.*
- string **getName** () const
Get the delivery service's name.
- void **setName** (const string &name)
Set the delivery service's name.
- vector< **Restaurant** * > **getRestaurants** ()
Get the list of restaurants.
- void **addRestaurant** (**Restaurant** &restaurant)
Add a restaurant to the delivery service.
- void **addAnEmployee** (**Courier** &courier)
Add a courier to the delivery service.
- vector< Employee * > & **getCouriers** ()
Get the list of couriers.
- void **sentInRestaurant** (size_t index, **Order** &order)
Send an order to the restaurant.

4.5.1 Constructor & Destructor Documentation

4.5.1.1 DeliveryService() [1/2]

```
DeliveryService::DeliveryService ( )
```

Default constructor of the **DeliveryService** (p. 15) class.

The default constructor initializes a **DeliveryService** (p. 15) object with a default name.

4.5.1.2 DeliveryService() [2/2]

```
DeliveryService::DeliveryService (
    const string & name )
```

Constructor of the **DeliveryService** (p. 15) class.

The constructor initializes a **DeliveryService** (p. 15) object with a given name.

Parameters

<i>name</i>	The delivery service's name.
-------------	------------------------------

4.5.2 Member Function Documentation

4.5.2.1 addAnEmployee()

```
void DeliveryService::addAnEmployee (
    Courier & courier )
```

Add a courier to the delivery service.

Adds a courier to the list of couriers of the delivery service.

Parameters

<i>courier</i>	The courier to add.
----------------	---------------------

4.5.2.2 addRestaurant()

```
void DeliveryService::addRestaurant (
    Restaurant & restaurant )
```

Add a restaurant to the delivery service.

Adds a restaurant to the list of restaurants of the delivery service.

Parameters

<i>restaurant</i>	The restaurant to add.
-------------------	------------------------

4.5.2.3 getCouriers()

```
vector< Employee * > & DeliveryService::getCouriers ( )
```

Get the list of couriers.

Returns

The list of couriers.

4.5.2.4 getName()

```
string DeliveryService::getName ( ) const
```

Get the delivery service's name.

Returns

The delivery service's name.

4.5.2.5 getRestaurants()

```
vector< Restaurant * > DeliveryService::getRestaurants ( )
```

Get the list of restaurants.

Returns

The list of restaurants.

4.5.2.6 sentInRestaurant()

```
void DeliveryService::sentInRestaurant (
    size_t index,
    Order & order )
```

Send an order to the restaurant.

Sends an order to the restaurant and adds it to the courier's list of orders.

Parameters

<i>index</i>	The index of the restaurant.
<i>order</i>	The order to send.

4.5.2.7 setName()

```
void DeliveryService::setName (
    const string & name )
```

Set the delivery service's name.

Parameters

<i>name</i>	The new name of the delivery service.
-------------	---------------------------------------

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

- DeliveryService.h
- **DeliveryService.cpp**

4.6 Dish Class Reference

Public Member Functions

- **Dish** ()

Constructor of the **Dish** (p. 17) class.

- **Dish** (const string &name, const double &price)

Constructor of the **Dish** (p. 17) class.

- string **getName** () const

Get the name of the dish.

- void **setName** (const string &name)

Set the name of the dish.

- double **getPrice** () const

Get the price of the dish.

- void **setPrice** (const double &price)

Set the price of the dish.

- bool **operator==** (const **Dish** &dish) const

Comparison operator for two **Dish** (p. 17) objects.

4.6.1 Constructor & Destructor Documentation

4.6.1.1 Dish() [1/2]

```
Dish::Dish ( )
```

Constructor of the **Dish** (p. 17) class.

The constructor initializes a **DishDataBase** (p. 21) object and gets a random dish from the database. It also sets the dish price to 10.

4.6.1.2 Dish() [2/2]

```
Dish::Dish (
    const string & name,
    const double & price )
```

Constructor of the **Dish** (p. 17) class.

The constructor initializes a **Dish** (p. 17) object with a given name and price.

Parameters

<i>name</i>	Name of the dish.
<i>price</i>	Price of the dish.

4.6.2 Member Function Documentation

4.6.2.1 getName()

```
string Dish::getName ( ) const
```

Get the name of the dish.

Returns

A string containing the name of the dish.

4.6.2.2 getPrice()

```
double Dish::getPrice ( ) const
```

Get the price of the dish.

Returns

Price of the dish.

4.6.2.3 operator==()

```
bool Dish::operator== (
    const Dish & dish ) const
```

Comparison operator for two **Dish** (p. 17) objects.

The operator compares the names of the dishes of two **Dish** (p. 17) objects.

Parameters

<i>dish</i>	Dish (p. 17) object to compare.
-------------	--

Returns

Returns true if the dish names match, otherwise returns false.

4.6.2.4 setName()

```
void Dish::setName (
    const string & name )
```

Set the name of the dish.

Parameters

<i>name</i>	Name of the dish.
-------------	-------------------

4.6.2.5 setPrice()

```
void Dish::setPrice (
    const double & price )
```

Set the price of the dish.

Parameters

<i>price</i>	Price of the dish.
--------------	--------------------

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

- Dish.h
- **Dish.cpp**

4.7 DishDataBase Class Reference

Public Member Functions

- **DishDataBase** ()
*Constructor of the **DishDataBase** (p. 21) class.*
- std::string **getRandomDish** ()
Get a random dish.

4.7.1 Constructor & Destructor Documentation

4.7.1.1 DishDataBase()

```
DishDataBase::DishDataBase ( )
```

Constructor of the **DishDataBase** (p. 21) class.

The constructor initializes the vector dishNames with dish names and selects a random dish.

4.7.2 Member Function Documentation

4.7.2.1 getRandomDish()

```
std::string DishDataBase::getRandomDish ( )
```

Get a random dish.

Returns

A string containing the name of the random dish.

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

- DishDB.h
- DishDB.cpp

4.8 Menu Class Reference

Public Member Functions

- **Menu** ()
*Constructor of the **Menu** (p. 22) class.*
- **Menu** (const size_t &size)
*Constructor of the **Menu** (p. 22) class.*
- **Dish** **getDish** (size_t i) const
Get a dish by index.
- void **addDish** (const **Dish** &dish)
Add a dish to the menu.
- void **addDish** (const string &name, const double &price)
Add a dish to the menu.
- void **delDish** (const **Dish** &dish)
Remove a dish from the menu.
- void **delDish** (const string &name)
Remove a dish from the menu.
- void **delDish** (const size_t &number)
Remove a dish from the menu.
- void **show** ()
Display the menu.

4.8.1 Constructor & Destructor Documentation

4.8.1.1 Menu() [1/2]

```
Menu::Menu ( ) [default]
```

Constructor of the **Menu** (p. 22) class.

Default constructor.

4.8.1.2 Menu() [2/2]

```
Menu::Menu (
    const size_t & size )
```

Constructor of the **Menu** (p. 22) class.

The constructor initializes a menu of a given size.

Parameters

<i>size</i>	Size of the menu.
-------------	-------------------

4.8.2 Member Function Documentation

4.8.2.1 addDish() [1/2]

```
void Menu::addDish (
    const Dish & dish )
```

Add a dish to the menu.

Parameters

<i>dish</i>	The dish to add.
-------------	------------------

4.8.2.2 addDish() [2/2]

```
void Menu::addDish (
    const string & name,
    const double & price = 10 )
```

Add a dish to the menu.

Parameters

<i>name</i>	Name of the dish.
<i>price</i>	Price of the dish.

4.8.2.3 delDish() [1/3]

```
void Menu::delDish (
    const Dish & dish )
```

Remove a dish from the menu.

Parameters

<i>dish</i>	The dish to remove.
-------------	---------------------

4.8.2.4 delDish() [2/3]

```
void Menu::delDish (
    const size_t & number )
```

Remove a dish from the menu.

Parameters

<i>number</i>	Number of the dish to remove.
---------------	-------------------------------

4.8.2.5 delDish() [3/3]

```
void Menu::delDish (
    const string & name )
```

Remove a dish from the menu.

Parameters

<i>name</i>	Name of the dish to remove.
-------------	-----------------------------

4.8.2.6 getDish()

```
Dish Menu::getDish (
    size_t i ) const
```

Get a dish by index.

Parameters

<i>i</i>	Index of the dish.
----------	--------------------

Returns

The dish at the given index.

4.8.2.7 show()

```
void Menu::show ( )
```

Display the menu.

Displays all the dishes in the menu.

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

- Menu.h
- **Menu.cpp**

4.9 Order Class Reference

Public Member Functions

- **Order** (**Client** &client)
*Constructor of the **Order** (p. 24) class.*
- bool **getStatus** () const
Get the status of the order.

- void **setStatus** (bool status)
Set the status of the order.
- size_t **getNumber** () const
Get the order number.
- Client * **getClient** () const
Get the client for whom the order was created.
- vector< **Dish** > **getDishes** ()
Get all dishes in the order.
- void **addDish** (const **Dish** &dish)
Add a dish to the order.
- void **delDish** (const **Dish** &dish)
Remove a dish from the order.
- bool **operator==** (const **Order** &other) const
*Comparison operator for two **Order** (p. 24) objects.*

4.9.1 Constructor & Destructor Documentation

4.9.1.1 Order()

```
Order::Order (
    Client & client )
```

Constructor of the **Order** (p. 24) class.

The constructor initializes an **Order** (p. 24) object with a given client.

Parameters

<i>client</i>	The client for whom the order is created.
---------------	---

4.9.2 Member Function Documentation

4.9.2.1 addDish()

```
void Order::addDish (
    const Dish & dish )
```

Add a dish to the order.

Parameters

<i>dish</i>	The dish to add to the order.
-------------	-------------------------------

4.9.2.2 delDish()

```
void Order::delDish (
    const Dish & dish )
```

Remove a dish from the order.

Parameters

<i>dish</i>	The dish to remove from the order.
-------------	------------------------------------

4.9.2.3 getClient()

```
Client * Order::getClient ( ) const
```

Get the client for whom the order was created.

Returns

The client for whom the order was created.

4.9.2.4 getDishes()

```
vector< Dish > Order::getDishes ( )
```

Get all dishes in the order.

Returns

A vector containing all the dishes in the order.

4.9.2.5 getNumber()

```
size_t Order::getNumber ( ) const
```

Get the order number.

Returns

Order (p. 24) number.

4.9.2.6 getStatus()

```
bool Order::getStatus ( ) const
```

Get the status of the order.

Returns

Order (p. 24) status.

4.9.2.7 operator==()

```
bool Order::operator== (
    const Order & other ) const
```

Comparison operator for two **Order** (p. 24) objects.

The operator compares the numbers of two **Order** (p. 24) objects.

Parameters

<i>other</i>	Order (p. 24) object to compare.
--------------	---

Returns

Returns true if the numbers match, otherwise returns false.

4.9.2.8 setStatus()

```
void Order::setStatus (
    bool status )
```

Set the status of the order.

Parameters

<i>status</i>	New order status.
---------------	-------------------

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

- Order.h
- **Order.cpp**

4.10 Restaurant Class Reference

Public Member Functions

- **Restaurant** ()
*Default constructor of the **Restaurant** (p. 27) class.*
- **Restaurant** (string name, **Menu** menu, double capital, size_t rating)
*Constructor of the **Restaurant** (p. 27) class.*
- string **getName** () const
Get the restaurant's name.
- void **setName** (const string &name)
Set the restaurant's name.
- float **getRating** () const
Get the restaurant's rating.
- void **setRating** (const float &rating)
Set the restaurant's rating.
- **Menu** **getMenu** () const
Get the restaurant's menu.
- void **setMenu** (const **Menu** &menu)
Set the restaurant's menu.
- double **getCapital** () const
Get the restaurant's capital.
- void **setCapital** (const double &capital)

- Set the restaurant's capital.*
- `vector< Employee * > & getCooks ()`
Get the list of cooks.
- `void revieveAnOrder (const Order & Order)`
Receive an order.
- `void addAnEmployee (Cook &cook)`
Add an employee to the restaurant.
- `void orderDistribution ()`
Distribute orders.
- `void dishTransfer (vector< Employee * > &couriers)`
Transfer dishes to couriers.
- `void showInfo () const`
Show restaurant information.

4.10.1 Constructor & Destructor Documentation

4.10.1.1 Restaurant() [1/2]

```
Restaurant::Restaurant ( )
```

Default constructor of the **Restaurant** (p. 27) class.

The default constructor initializes a **Restaurant** (p. 27) object with a default name, menu, capital, and rating.

4.10.1.2 Restaurant() [2/2]

```
Restaurant::Restaurant (
    string name,
    Menu menu,
    double capital,
    size_t rating = 0 )
```

Constructor of the **Restaurant** (p. 27) class.

The constructor initializes a **Restaurant** (p. 27) object with a given name, menu, capital, and rating.

Parameters

<i>name</i>	The restaurant's name.
<i>menu</i>	The restaurant's menu.
<i>capital</i>	The restaurant's capital.
<i>rating</i>	The restaurant's rating.

4.10.2 Member Function Documentation

4.10.2.1 addAnEmployee()

```
void Restaurant::addAnEmployee (
    Cook & cook )
```

Add an employee to the restaurant.

Adds cooks to the restaurant.

Parameters

<i>cook</i>	The cook.
-------------	-----------

4.10.2.2 dishTransfer()

```
void Restaurant::dishTransfer (
    vector< Employee * > & couriers )
```

Transfer dishes to couriers.

Transfers prepared dishes to couriers.

Parameters

<i>couriers</i>	The couriers.
-----------------	---------------

4.10.2.3 getCapital()

```
double Restaurant::getCapital ( ) const
```

Get the restaurant's capital.

Returns

The restaurant's capital.

4.10.2.4 getCooks()

```
vector< Employee * > & Restaurant::getCooks ( )
```

Get the list of cooks.

Returns

The list of cooks.

4.10.2.5 getMenu()

```
Menu Restaurant::getMenu ( ) const
```

Get the restaurant's menu.

Returns

The restaurant's menu.

4.10.2.6 getName()

```
string Restaurant::getName ( ) const
```

Get the restaurant's name.

Returns

The restaurant's name.

4.10.2.7 getRating()

```
float Restaurant::getRating ( ) const
```

Get the restaurant's rating.

Returns

The restaurant's rating.

4.10.2.8 orderDistribution()

```
void Restaurant::orderDistribution ( )
```

Distribute orders.

Distributes orders among cooks.

4.10.2.9 revieveAnOrder()

```
void Restaurant::revieveAnOrder (
    const Order & order )
```

Receive an order.

Adds an order to the list of orders.

Parameters

<i>order</i>	The order to receive.
--------------	-----------------------

4.10.2.10 setCapital()

```
void Restaurant::setCapital (
    const double & capital )
```

Set the restaurant's capital.

Parameters

<i>capital</i>	The new capital of the restaurant.
----------------	------------------------------------

4.10.2.11 setMenu()

```
void Restaurant::setMenu (
    const Menu & menu )
```

Set the restaurant's menu.

Parameters

<i>menu</i>	The new menu of the restaurant.
-------------	---------------------------------

4.10.2.12 setName()

```
void Restaurant::setName (
    const string & name )
```

Set the restaurant's name.

Parameters

<i>name</i>	The new name of the restaurant.
-------------	---------------------------------

4.10.2.13 setRating()

```
void Restaurant::setRating (
    const float & rating )
```

Set the restaurant's rating.

Parameters

<i>rating</i>	The new rating of the restaurant.
---------------	-----------------------------------

4.10.2.14 showInfo()

```
void Restaurant::showInfo ( ) const
```

Show restaurant information.

Outputs information about the restaurant.

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

- Restaurant.h
- **Restaurant.cpp**

Chapter 5

File Documentation

5.1 Client.cpp File Reference

File containing the implementation of the **Client** (p. 8) class.

```
#include "Client.h"
#include "DeliveryService.h"
```

5.1.1 Detailed Description

File containing the implementation of the **Client** (p. 8) class.

Author

Verkovich E.V.

Date

November 26, 2023

5.2 Client.h

```
00001 #pragma once
00002
00003 #include <iostream>
00004 #include <string>
00005 #include "Dish.h"
00006 #include "Order.h"
00007
00008
00009 using namespace std;
00010 class DeliveryService;
00011 class Restaurant;
00012 class Client {
00013 private:
00014     string name_;
00015     string address_;
00016     string number_;
00017     vector<Dish>dishes;
00018
00019     //Restaurant* ChooseRestaurant(DeliveryService& delServ, size_t i);
00020 public:
```

```

00021     Client();
00022     Client(const string& name, const string& address, const string& number);
00023
00024     string getName() const;
00025     void setName(const string& name);
00026
00027     string getAddress() const;
00028     void setAddress(const string& address);
00029
00030     string getNumber() const;
00031     void setNumber(const string& number);
00032
00033     void takeDish(const Dish& dish);
00034
00035     void placeAnOrder(DeliveryService& delServ, bool manual);
00036 };

```

5.3 Cook.cpp File Reference

File containing the implementation of the **Cook** (p. 11) class.

```
#include "Cook.h"
```

5.3.1 Detailed Description

File containing the implementation of the **Cook** (p. 11) class.

Author

Verkovich E.V.

Date

November 26, 2023

5.4 Cook.h

```

00001 #pragma once
00002
00003 #include <vector>
00004 #include <iostream>
00005 #include <utility>
00006 #include <string>
00007 #include <algorithm>
00008 #include <memory>
00009 #include "Employee.h"
00010
00011 #define MROT 500
00012
00013 using namespace std;
00014
00015 class Cook : public Employee {
00016 private:
00017     const size_t maxOrderQuantity_ = 10;
00018     vector<Dish>preparedDishes;
00019
00020 public:
00021     Cook();
00022     Cook(const string& name, const double& salary, const string& position);
00023
00024     void recieveAnOrder(Order& order) override;
00025     void toWork() override;
00026
00027     vector <Dish> passOnDishes();
00028 };

```


5.5 Courier.cpp File Reference

File containing the implementation of the **Courier** (p. 13) class.

```
#include "Courier.h"
#include "DeliveryService.h"
```

5.5.1 Detailed Description

File containing the implementation of the **Courier** (p. 13) class.

Author

Verkovich E.V.

Date

November 26, 2023

5.6 Courier.h

```
00001 #pragma once
00002
00003 #include "Employee.h"
00004 #include "Order.h"
00005 #include "Client.h"
00006 #include <memory>
00007
00008 #define MROT 500
00009
00010 using namespace std;
00011
00012 class Courier : public Employee {
00013 private:
00014     const size_t maxOrderQuantity_ = 10;
00015     const size_t bagSize_ = 10;
00016     vector<Dish>bag;
00017
00018
00019     bool findDish(const Dish& dish) const;
00020     Dish takeOut(const Dish& dish);
00021
00022 public:
00023     Courier();
00024     Courier(const string& name, const double& salary, const string& position);
00025
00026     void recieveAnOrder(Order& order) override;
00027
00028     void putInDishes(const vector<Dish>& preparedDishes);
00029
00030     void toWork() override;
00031 };
```

5.7 Delivery.h

```
00001 #pragma once
00002 #include <vector>
00003 #include <iostream>
00004 #include <utility>
00005 #include <string>
00006 #include <algorithm>
00007 #include <memory>
00008 #include "DishDB.h"
00009
```

```
00010 #define MROT 500
00011 using namespace std;
00012
00013 class Client;
00014 class DeliveryService;
00015
00016
00017
00018
00019
00020
00021
00022
00023
00024
00025
00026
00027
00028
00029
```

5.8 DeliveryService.cpp File Reference

File containing the implementation of the **DeliveryService** (p. 15) class.

```
#include "DeliveryService.h"
```

5.8.1 Detailed Description

File containing the implementation of the **DeliveryService** (p. 15) class.

Author

Verkovich E.V.

Date

November 26, 2023

5.9 DeliveryService.h

```
00001 #pragma once
00002
00003 #include "Restaurant.h"
00004
00005 #define MROT 500
00006
00007 using namespace std;
00008
00009
00010 class DeliveryService {
00011 private:
00012     string name_;
00013     vector<Restaurant*> restaurants;
00014     vector<Employee*> couriers;
00015
00016 public:
00017     DeliveryService();
00018     DeliveryService(const string& name);
00019
00020     string getName() const;
00021     void setName(const string& name);
00022
00023     vector<Restaurant*> getRestaurants();
00024     void addRestaurant(Restaurant& restaurant);
00025
00026     void addAnEmployee(Courier& courier);
00027     vector<Employee*>& getCouriers();
00028
00029     void sentInRestaurant(size_t index, Order& order);
00030 };
```

5.10 Dish.cpp File Reference

File containing the implementation of the **Dish** (p. 17) class.

```
#include "Dish.h"
```

5.10.1 Detailed Description

File containing the implementation of the **Dish** (p. 17) class.

Author

Verkovich.E.V.

Date

November 26, 2023

5.11 Dish.h

```
00001 #pragma once
00002 #include <string>
00003 #include "DishDB.h"
00004
00005 using namespace std;
00006 class Dish {
00007 private:
00008     string name_;
00009     double price_;
00010
00011 public:
00012     Dish();
00013     Dish(const string& name, const double& price);
00014
00015     string getName() const;
00016     void setName(const string& name);
00017     double getPrice() const;
00018     void setPrice(const double& price);
00019
00020     bool operator==(const Dish& dish) const;
00021 };
```

5.12 DishDB.h

```
00001 #pragma once
00002
00003 #include <string>
00004 #include<vector>
00005
00006 class DishDataBase {
00007 public:
00008     DishDataBase();
00009     std::string getRandomDish();
00010
00011 private:
00012     std::vector<std::string> dishNames;
00013     size_t randomDishIndex;
00014     std::string randomDish;
00015 };
00016
```

5.13 Employee.cpp File Reference

File containing the implementation of the Employee class.

```
#include "Employee.h"
```

5.13.1 Detailed Description

File containing the implementation of the Employee class.

Author

Verkovich E.V.

Date

November 26, 2023

5.14 Employee.h

```
00001 #pragma once
00002
00003 #include "Order.h"
00004
00005 using namespace std;
00006
00007 class Employee abstract {
00008 protected:
00009     string name_;
00010     string position_;
00011     double salary_;
00012     bool busy_; //0 -- , 1 --
00013     vector<Order>orders;
00014 public:
00015     Employee(const string& name, const string& position, const double& salary);
00016
00017     string getName() const;
00018     void setName(const string& name);
00019
00020     string getPosition() const;
00021     void setPosition(const string& position);
00022
00023     double getSalary() const;
00024     void setSalary(const double& salary);
00025
00026     bool getBusy()const;
00027     void setBusy(const int& busy);
00028
00029     vector<Order>& getOrders();
00030
00031     virtual void recieveAnOrder(Order& order) = 0;
00032     virtual void toWork() = 0;
00033 };
```

5.15 Menu.cpp File Reference

File containing the implementation of the **Menu** (p. 22) class.

```
#include "Menu.h"
#include <iostream>
#include <vector>
```

5.15.1 Detailed Description

File containing the implementation of the **Menu** (p. 22) class.

Author

Verkovic E.V.

Date

November 26, 2023

5.16 Menu.h

```
00001 #pragma once
00002 #include "Dish.h"
00003 #include <iostream>
00004 #include <vector>
00005 using namespace std;
00006
00007 class Menu {
00008 private:
00009     vector<pair<size_t, Dish>> dishes;
00010     bool isSameInMenu(const Dish& dish);
00011
00012 public:
00013     Menu();
00014     Menu(const size_t& size);
00015
00016     Dish getDish(size_t i) const;
00017
00018     void addDish(const Dish& dish);
00019     void addDish(const string& name, const double& price);
00020     void delDish(const Dish& dish);
00021     void delDish(const string& name);
00022     void delDish(const size_t& number);
00023
00024     void show();
00025 };
```

5.17 Order.cpp File Reference

File containing the implementation of the **Order** (p. 24) class.

```
#include "Order.h"
#include "Client.h"
```

5.17.1 Detailed Description

File containing the implementation of the **Order** (p. 24) class.

Author

Verkovich E.V.

Date

November 26, 2023

5.18 Order.h

```
00001 #pragma once
00002
00003 #include <iostream>
00004 #include <memory>
00005 #include <algorithm>
00006 #include "Dish.h"
00007
00008 using namespace std;
00009
00010 class Client;
00011 class Order {
00012 private:
00013     static size_t currentNumber_;
00014     size_t number_;
00015     bool status_;    //0 -- (), 1 -- , 2 -- , 3 --
00016     vector<Dish> dishes;
00017     Client* client_;
00018
00019 public:
00020     Order(Client& client);
00021     bool getStatus() const;
00022     void setStatus(bool status);
00023
00024     size_t getNumber() const;
00025
00026     Client* getClient() const;
00027
00028     vector<Dish> getDishes();
00029
00030     void addDish(const Dish& dish);
00031     void delDish(const Dish& dish);
00032
00033     bool operator==(const Order& other) const;
00034 };
```

5.19 Restaurant.cpp File Reference

File containing the implementation of the **Restaurant** (p. 27) class.

```
#include "Restaurant.h"
```

5.19.1 Detailed Description

File containing the implementation of the **Restaurant** (p. 27) class.

Author

Your Name

Date

November 26, 2023

5.20 Restaurant.h

```
00001 #pragma once
00002
00003 #include "Menu.h"
00004 #include "Cook.h"
00005 #include "Courier.h"
00006
00007
00008 #define MROT 500
00009
00010 using namespace std;
00011
00012 class Restaurant {
00013 private:
00014     string name_;
00015     float rating_;
00016     Menu menu_;
00017     double capital_;
00018
00019     vector<Employee*> cooks;
00020     vector<Order> orders;
00021
00022     bool checkOrderMatch(Employee* cook, Employee* courier);
00023 public:
00024     Restaurant();
00025     Restaurant(string name, Menu menu, double capital, size_t rating);
00026
00027     string getName() const;
00028     void setName(const string& name);
00029
00030     float getRating() const;
00031     void setRating(const float& rating);
00032
00033     Menu getMenu() const;
00034     void setMenu(const Menu& menu);
00035
00036     double getCapital() const;
00037     void setCapital(const double& capital);
00038
00039     vector<Employee*>& getCooks();
00040
00041     void reviewAnOrder(const Order& Order);
00042
00043     void addAnEmployee(Cook& cook);
00044
00045     void orderDistribution();
00046
00047     void dishTransfer(vector<Employee*>& couriers);
00048
00049     void showInfo() const;
00050 };
```


Index

- abstract, 7
- addAnEmployee
 - DeliveryService, 16
 - Restaurant, 28
- addDish
 - Menu, 23
 - Order, 25
- addRestaurant
 - DeliveryService, 16
- Client, 8
 - Client, 8
 - getAddress, 9
 - getName, 9
 - getNumber, 9
 - placeAnOrder, 9
 - setAddress, 9
 - setName, 10
 - setNumber, 10
 - takeDish, 10
- Client.cpp, 33
- Cook, 11
 - Cook, 11
 - passOnDishes, 12
 - recieveAnOrder, 12
 - toWork, 12
- Cook.cpp, 34
- Courier, 13
 - Courier, 13
 - putInDishes, 14
 - recieveAnOrder, 14
 - toWork, 14
- Courier.cpp, 35
- delDish
 - Menu, 23, 24
 - Order, 25
- DeliveryService, 15
 - addAnEmployee, 16
 - addRestaurant, 16
 - DeliveryService, 15
 - getCouriers, 16
 - getName, 16
 - getRestaurants, 16
 - sentInRestaurant, 17
 - setName, 17
- DeliveryService.cpp, 36
- Dish, 17
 - Dish, 18
 - getName, 18
 - getPrice, 19
 - operator==, 19
 - setName, 19
 - setPrice, 19
- Dish.cpp, 37
- DishDataBase, 21
 - DishDataBase, 21
 - getRandomDish, 21
- dishTransfer
 - Restaurant, 29
- Employee.cpp, 38
- getAddress
 - Client, 9
- getCapital
 - Restaurant, 29
- getClient
 - Order, 26
- getCooks
 - Restaurant, 29
- getCouriers
 - DeliveryService, 16
- getDish
 - Menu, 24
- getDishes
 - Order, 26
- getMenu
 - Restaurant, 29
- getName
 - Client, 9
 - DeliveryService, 16
 - Dish, 18
 - Restaurant, 29
- getNumber
 - Client, 9
 - Order, 26
- getPrice
 - Dish, 19
- getRandomDish
 - DishDataBase, 21
- getRating
 - Restaurant, 30
- getRestaurants
 - DeliveryService, 16
- getStatus
 - Order, 26
- Menu, 22
 - addDish, 23

- delDish, 23, 24
- getDish, 24
- Menu, 22
- show, 24
- Menu.cpp, 38
- operator==
 - Dish, 19
 - Order, 26
- Order, 24
 - addDish, 25
 - delDish, 25
 - getClient, 26
 - getDishes, 26
 - getNumber, 26
 - getStatus, 26
 - operator==, 26
 - Order, 25
 - setStatus, 27
- Order.cpp, 39
- orderDistribution
 - Restaurant, 30
- passOnDishes
 - Cook, 12
- placeAnOrder
 - Client, 9
- putInDishes
 - Courier, 14
- recieveAnOrder
 - Cook, 12
 - Courier, 14
- Restaurant, 27
 - addAnEmployee, 28
 - dishTransfer, 29
 - getCapital, 29
 - getCooks, 29
 - getMenu, 29
 - getName, 29
 - getRating, 30
 - orderDistribution, 30
 - Restaurant, 28
 - revieveAnOrder, 30
 - setCapital, 30
 - setMenu, 31
 - setName, 31
 - setRating, 31
 - showInfo, 31
- Restaurant.cpp, 40
- revieveAnOrder
 - Restaurant, 30
- sentInRestaurant
 - DeliveryService, 17
- setAddress
 - Client, 9
- setCapital
 - Restaurant, 30
- setMenu
 - Restaurant, 31
- setName
 - Client, 10
 - DeliveryService, 17
 - Dish, 19
 - Restaurant, 31
- setNumber
 - Client, 10
- setPrice
 - Dish, 19
- setRating
 - Restaurant, 31
- setStatus
 - Order, 27
- show
 - Menu, 24
- showInfo
 - Restaurant, 31
- takeDish
 - Client, 10
- toWork
 - Cook, 12
 - Courier, 14