

## Pizzeria - Semestral Project (OOP)

Michal Ručka - RUC0066

Generated by Doxygen 1.9.1



<b>1 Pizzeria: Semestral Project (OOP)</b>	<b>1</b>
<b>2 Hierarchical Index</b>	<b>3</b>
2.1 Class Hierarchy . . . . .	3
<b>3 Class Index</b>	<b>5</b>
3.1 Class List . . . . .	5
<b>4 Class Documentation</b>	<b>7</b>
4.1 AbstractEmployee Class Reference . . . . .	7
4.2 Cook Class Reference . . . . .	7
4.2.1 Member Function Documentation . . . . .	7
4.2.1.1 EmployeeInfo() . . . . .	8
4.2.1.2 GetPizzasMade() . . . . .	8
4.3 Customer Class Reference . . . . .	8
4.3.1 Constructor & Destructor Documentation . . . . .	8
4.3.1.1 Customer() . . . . .	8
4.3.2 Member Function Documentation . . . . .	9
4.3.2.1 GetCustAddress() . . . . .	9
4.3.2.2 GetCustID() . . . . .	9
4.3.2.3 GetCustName() . . . . .	9
4.3.2.4 GetOrder() . . . . .	9
4.3.2.5 GetOrdersCount() . . . . .	10
4.4 Delivery Class Reference . . . . .	10
4.4.1 Member Function Documentation . . . . .	10
4.4.1.1 CalculateGasMoney() . . . . .	11
4.4.1.2 EmployeeInfo() . . . . .	11
4.4.1.3 GetGasMoney() . . . . .	11
4.4.1.4 GetOrdersDelivered() . . . . .	11
4.5 Drink Class Reference . . . . .	12
4.5.1 Constructor & Destructor Documentation . . . . .	12
4.5.1.1 Drink() . . . . .	12
4.5.2 Member Function Documentation . . . . .	13
4.5.2.1 GetDrinkName() . . . . .	13
4.5.2.2 SetDrinkID() . . . . .	13
4.6 Employee Class Reference . . . . .	13
4.6.1 Constructor & Destructor Documentation . . . . .	14
4.6.1.1 Employee() . . . . .	14
4.6.1.2 ~Employee() . . . . .	14
4.6.2 Member Function Documentation . . . . .	14
4.6.2.1 CalculateSalary() . . . . .	15
4.6.2.2 CanChangePosition() . . . . .	15
4.6.2.3 CanCook() . . . . .	15

4.6.2.4 CanDrive()	15
4.6.2.5 EmployeeInfo()	15
4.6.2.6 GetID()	16
4.6.2.7 GetName()	16
4.6.2.8 GetSalary()	16
4.6.2.9 GetWorkedHours()	16
4.6.2.10 GetWorkedYears()	17
4.6.2.11 SetID()	17
4.7 Menu Class Reference	18
4.7.1 Member Function Documentation	19
4.7.1.1 AddDrink()	19
4.7.1.2 AddPizza()	19
4.7.1.3 DeletionDrinks()	19
4.7.1.4 DrinkExist()	20
4.7.1.5 GetDrink()	20
4.7.1.6 GetDrinkIndex()	20
4.7.1.7 GetDrinksCount()	21
4.7.1.8 GetPizza()	21
4.7.1.9 GetPizzaIndex()	21
4.7.1.10 GetPizzasCount()	21
4.7.1.11 PizzaExist()	22
4.7.1.12 PrintWholeMenu()	22
4.7.1.13 RemoveDrink()	22
4.7.1.14 RemovePizza()	22
4.7.1.15 SeekDrinkName()	23
4.7.1.16 SeekPizzaName()	23
4.8 Order Class Reference	24
4.8.1 Member Function Documentation	24
4.8.1.1 AddDrink()	24
4.8.1.2 AddPizza()	24
4.8.1.3 GetDrink()	25
4.8.1.4 GetDrinksInOrder()	25
4.8.1.5 GetOrderID()	25
4.8.1.6 GetPizza()	25
4.8.1.7 GetPizzasInOrder()	26
4.9 Pizza Class Reference	26
4.9.1 Constructor & Destructor Documentation	26
4.9.1.1 Pizza()	26
4.9.2 Member Function Documentation	27
4.9.2.1 GetPizzaID()	27
4.9.2.2 GetPizzaName()	27
4.9.2.3 GetPizzaPrice()	27

4.9.2.4 GetPizzaProfit()	27
4.9.2.5 GetStaticID()	28
4.9.2.6 SetPizzalD()	28
4.10 Pizzeria Class Reference	28
4.10.1 Constructor & Destructor Documentation	30
4.10.1.1 Pizzeria() [1/4]	30
4.10.1.2 Pizzeria() [2/4]	30
4.10.1.3 Pizzeria() [3/4]	30
4.10.1.4 Pizzeria() [4/4]	31
4.10.2 Member Function Documentation	31
4.10.2.1 AddAsCook()	31
4.10.2.2 AddAsDelivery()	31
4.10.2.3 AddCustomer()	32
4.10.2.4 AddEmployee()	32
4.10.2.5 DeleteEmployee()	32
4.10.2.6 EmplExist()	33
4.10.2.7 GetCook()	33
4.10.2.8 GetCookIndex()	33
4.10.2.9 GetCustomer()	34
4.10.2.10 GetDelivery()	34
4.10.2.11 GetDeliveryIndex()	34
4.10.2.12 GetEmployee()	35
4.10.2.13 GetEmployeeIndex()	35
4.10.2.14 GetEmployeesCount()	35
4.10.2.15 GetPizzeriaName()	35
4.10.2.16 GetPizzeriaTel()	36
4.10.2.17 GetPizzeriaWeb()	36
4.10.2.18 IntroducePizzeria()	36
4.10.2.19 MakeSpace()	36
4.10.2.20 OrderForCustomer()	36
4.10.2.21 PrintCook()	37
4.10.2.22 PrintCustomers()	37
4.10.2.23 PrintDelivery()	37
4.10.2.24 PrintEmployees()	37
4.10.2.25 ShiftOrderCook()	37
4.10.2.26 ShiftOrderDeliv()	38
4.10.2.27 ShiftOrderEmplo()	38
4.10.2.28 StopProgram()	38
<b>Index</b>	<b>39</b>



# Chapter 1

## Pizzeria: Semestral Project (OOP)

1. Over 7 classes, each one of them is in cpp/h file
2. [Pizzeria](#) class is created and used in main()
3. Dynamic array is implemented in [Pizzeria](#), [Menu](#), [Order](#), Employees
4. Overloaded constructor is in pizzeria
5. Static is used in [Order](#), [Pizza](#), [Drink](#)
6. Inheritance is used in thsoe classes: [AbstractEmployee](#), [Employee](#), [Cook](#), [Delivery](#)
7. EployeeInfo() in all employees
8. Pizzeria-Documentation.pdf
9. Protected is used in [Pizza](#), [Menu](#), [Drink](#), [Employee](#)
10. [AbstractEmployee](#) is an abstract class
11. Polymorphism is used in [Pizzeria::TestPolym\(\)](#) function





## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

AbstractEmployee . . . . .	7
Employee . . . . .	13
Cook . . . . .	7
Delivery . . . . .	10
Customer . . . . .	8
Drink . . . . .	12
Menu . . . . .	18
Order . . . . .	24
Pizza . . . . .	26
Pizzeria . . . . .	28



## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">AbstractEmployee</a>	7
<a href="#">Cook</a>	7
<a href="#">Customer</a>	8
<a href="#">Delivery</a>	10
<a href="#">Drink</a>	12
<a href="#">Employee</a>	13
<a href="#">Menu</a>	18
<a href="#">Order</a>	24
<a href="#">Pizza</a>	26
<a href="#">Pizzeria</a>	28



## Chapter 4

# Class Documentation

### 4.1 AbstractEmployee Class Reference

Inheritance diagram for AbstractEmployee:

### 4.2 Cook Class Reference

Inheritance diagram for Cook:

Collaboration diagram for Cook:

#### Public Member Functions

- [Cook](#) (std::string nam, bool cook, bool drive)  
*Cook class constructor.*
- [~Cook](#) ()  
*Cook class constructor.*
- virtual void [EmployeeInfo](#) ()  
*Bacis info about this cook.*
- int [GetPizzasMade](#) ()

#### Additional Inherited Members

#### 4.2.1 Member Function Documentation

#### 4.2.1.1 EmployeeInfo()

```
void Cook::EmployeeInfo ( ) [virtual]
```

Basis info about this cook.

Prints name, position and how many pizzas he made

Reimplemented from [Employee](#).

#### 4.2.1.2 GetPizzasMade()

```
int Cook::GetPizzasMade ( )
```

##### Returns

Number of pizzas made by this employee

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

- Cook.h
- Cook.cpp

## 4.3 Customer Class Reference

### Public Member Functions

- [Customer](#) (std::string nam, std::string addr)  
*Customer class constructor.*
- [~Customer](#) ()  
*Customer class destructor.*
- int [GetCustID](#) ()
- std::string [GetCustAddress](#) ()
- std::string [GetCustName](#) ()
- int [GetOrdersCount](#) ()
- void [CreateOrder](#) ()  
*Creation of new order.*
- [Order](#) \* [GetOrder](#) (int index)

### Static Public Attributes

- static int **overall\_orders\_count** = 0

## 4.3.1 Constructor & Destructor Documentation

#### 4.3.1.1 Customer()

```
Customer::Customer (
    std::string nam,
    std::string addr )
```

[Customer](#) class constructor.

## Parameters

<i>nam</i>	Name of the customer
<i>addr</i>	Adress of the customer

## 4.3.2 Member Function Documentation

### 4.3.2.1 GetCustAddress()

```
std::string Customer::GetCustAddress ( )
```

## Returns

Customer's adress

### 4.3.2.2 GetCustID()

```
int Customer::GetCustID ( )
```

## Returns

Customer's ID

### 4.3.2.3 GetCustName()

```
std::string Customer::GetCustName ( )
```

## Returns

Customer's name

### 4.3.2.4 GetOrder()

```
Order * Customer::GetOrder (
    int index )
```

#### Parameters

<i>index</i>	ID of the seeked order
--------------	------------------------

#### Returns

[Order](#) with selected ID

#### 4.3.2.5 GetOrdersCount()

```
int Customer::GetOrdersCount ( )
```

#### Returns

[Customer](#)'s orders count

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

- Customer.h
- Customer.cpp

## 4.4 Delivery Class Reference

Inheritance diagram for Delivery:

Collaboration diagram for Delivery:

### Public Member Functions

- [Delivery](#) (std::string nam, bool cook, bool drive)  
*[Delivery](#) class constructor.*
- [~Delivery](#) ()  
*[Delivery](#) class destructor.*
- int [GetGasMoney](#) ()
- int [GetOrdersDelivered](#) ()
- virtual void [EmployeeInfo](#) ()  
*Bacis info about this delivery guy.*
- void [CalculateGasMoney](#) ()  
*Calculate money spent on gas.*

### Additional Inherited Members

#### 4.4.1 Member Function Documentation



#### 4.4.1.1 CalculateGasMoney()

```
void Delivery::CalculateGasMoney ( )
```

Calculate money spent on gas.

[Pizzeria](#) gives 30czk for every order their employee delivered. So this is just (orders\_delivered) \* 30.

#### 4.4.1.2 EmployeeInfo()

```
void Delivery::EmployeeInfo ( ) [virtual]
```

Basis info about this delivery guy.

Prints name, position, total orders delivered and money spent on gas

Reimplemented from [Employee](#).

#### 4.4.1.3 GetGasMoney()

```
int Delivery::GetGasMoney ( )
```

##### Returns

Money spent on gas

#### 4.4.1.4 GetOrdersDelivered()

```
int Delivery::GetOrdersDelivered ( )
```

##### Returns

Total orders delivered

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

- [Delivery.h](#)
- [Delivery.cpp](#)

## 4.5 Drink Class Reference

### Public Member Functions

- [Drink](#) (std::string nam, int cost, int prof)  
*Drink class constructor.*
- [~Drink](#) ()  
*Drink class destructor.*
- void [SetDrinkID](#) (int id)  
*Sets ID to selected drink.*
- std::string [GetDrinkName](#) ()
- int [GetDrinkPrice](#) ()  
*Returns price of the drink.*
- int [GetDrinkProfit](#) ()  
*Returns profit that pizzeria make on this exact drink.*
- int [GetDrinkID](#) ()  
*Returns ID of the drink.*
- int [GetStaticID](#) ()  
*Returns static ID of the drinks.*

### Protected Attributes

- std::string **drink\_name**
- int **cost**
- int **profit**
- int **drink\_ID**

### Static Protected Attributes

- static int **static\_drinkID** = 0

### 4.5.1 Constructor & Destructor Documentation

#### 4.5.1.1 Drink()

```
Drink::Drink (
    std::string nam,
    int cost,
    int prof )
```

[Drink](#) class constructor.

#### Parameters

<i>nam</i>	Name of the drink
<i>cost</i>	Price of the drink
<i>prof</i>	Profit that pizzeria make every time they sell this product

## 4.5.2 Member Function Documentation

### 4.5.2.1 GetDrinkName()

```
std::string Drink::GetDrinkName ( )
```

#### Returns

Name of the drink

### 4.5.2.2 SetDrinkID()

```
void Drink::SetDrinkID (
    int id )
```

Sets ID to selected drink.

#### Parameters

<i>id</i>	New ID of the drink
-----------	---------------------

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

- Drink.h
- Drink.cpp

## 4.6 Employee Class Reference

Inheritance diagram for Employee:

Collaboration diagram for Employee:

### Public Member Functions

- [Employee](#) (std::string nam, bool cook, bool drive)
- [~Employee](#) ()
- void [SetID](#) (int id)  
*Sets the id of the employee.*
- std::string [GetName](#) ()
- int [GetID](#) ()
- int [GetWorkedHours](#) ()
- int [GetWorkedYears](#) ()

- int [GetSalary](#) ()
- bool [CanCook](#) ()
- bool [CanDrive](#) ()
- virtual void [EmployeeInfo](#) ()  
*Prints basic info about this exact employee.*
- void [CalculateSalary](#) ()  
*Calculates salary.*
- bool [CanChangePosition](#) ()

## Protected Attributes

- std::string **position**

## 4.6.1 Constructor & Destructor Documentation

### 4.6.1.1 Employee()

```
Employee::Employee (
    std::string nam,
    bool cook,
    bool drive )
```

[Employee](#) class constructor

#### Parameters

<i>nam</i>	Name of the employee
<i>cook</i>	Can this person cook?
<i>drive</i>	Can this person drive?

### 4.6.1.2 ~Employee()

```
Employee::~Employee ( )
```

[Employee](#) class destructor

## 4.6.2 Member Function Documentation

#### 4.6.2.1 CalculateSalary()

```
void Employee::CalculateSalary ( )
```

Calculates salary.

This calculates salary based on years worked in this pizzeria

#### 4.6.2.2 CanChangePosition()

```
bool Employee::CanChangePosition ( )
```

Return values

<i>TRUE</i>	If the person have ability to change position
<i>FALSE</i>	If the person does not have an ability to change position

#### 4.6.2.3 CanCook()

```
bool Employee::CanCook ( )
```

Return values

<i>TRUE</i>	if this person can cook
<i>FALSE</i>	if this persno cannot cook

#### 4.6.2.4 CanDrive()

```
bool Employee::CanDrive ( )
```

Return values

<i>TRUE</i>	if this person can drive
<i>FALSE</i>	if this persno cannot drive

#### 4.6.2.5 EmployeeInfo()

```
void Employee::EmployeeInfo ( ) [virtual]
```

Prints basic info about this exact employee.

This will print name and current position

Implements [AbstractEmployee](#).

Reimplemented in [Delivery](#), and [Cook](#).

#### 4.6.2.6 GetID()

```
int Employee::GetID ( )
```

##### Returns

ID of the employee

#### 4.6.2.7 GetName()

```
std::string Employee::GetName ( )
```

##### Returns

Name of the employee

#### 4.6.2.8 GetSalary()

```
int Employee::GetSalary ( )
```

##### Returns

[Employee](#)'s salary

#### 4.6.2.9 GetWorkedHours()

```
int Employee::GetWorkedHours ( )
```

##### Returns

Amount of hours this employee have worked in this month

#### 4.6.2.10 GetWorkedYears()

```
int Employee::GetWorkedYears ( )
```

##### Returns

Amount of years this employee have been working for this pizzeria

#### 4.6.2.11 SetID()

```
void Employee::SetID (
    int id )
```

Sets the id of the employee.

## Parameters

<i>id</i>	New ID of the employee
-----------	------------------------

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

- Employee.h
- Employee.cpp

## 4.7 Menu Class Reference

Collaboration diagram for Menu:

### Public Member Functions

- [Menu](#) ()  
*Menu class constructor.*
- [~Menu](#) ()  
*Menu class destructor.*
- void [AddDrink](#) (std::string nam, int cst, int prof)  
*Adds drink into menu.*
- void [AddPizza](#) (std::string nam, int cst, int prof)  
*Adds pizza into menu.*
- void [RemoveDrink](#) (int id)
- void [RemovePizza](#) (int id)
- [Drink](#) \* [GetDrink](#) (int id)
- [Pizza](#) \* [GetPizza](#) (int id)
- bool [DrinkExist](#) (int id)
- bool [PizzaExist](#) (int id)
- bool [SeekDrinkName](#) (std::string nam)
- bool [SeekPizzaName](#) (std::string nam)
- int [GetDrinksCount](#) ()
- int [GetPizzasCount](#) ()
- int [GetDrinkIndex](#) (int id)
- int [GetPizzaIndex](#) (int id)
- void [PrintWholeMenu](#) ()  
*Prints the whole menu.*
- void [DeletionDrinks](#) ()  
*Starts the drink deletion spree.*
- void [DeletionPizzas](#) ()  
*Starts the pizza deletion spree Deletion process works until theres '-1' on the input.*

### Protected Attributes

- int **stuff\_in\_menu**
- int **drinks\_count**
- int **pizzas\_count**
- [Drink](#) \* **drinks** [DRI]
- [Pizza](#) \* **pizzas** [PIZ]



## 4.7.1 Member Function Documentation

### 4.7.1.1 AddDrink()

```
void Menu::AddDrink (
    std::string nam,
    int cst,
    int prof )
```

Adds drink into menu.

Checks if drink is already in the menu and also if the total drinks number is lower than possible. If conditions are met, then it adds this drink into menu.

#### Parameters

<i>nam</i>	Name of the drink
<i>cst</i>	Cost of the drink
<i>prof</i>	Profit pizzeria make on this drink

### 4.7.1.2 AddPizza()

```
void Menu::AddPizza (
    std::string nam,
    int cst,
    int prof )
```

Adds pizza into menu.

Checks if pizza is already in the menu and also if the total pizzas number is lower than possible. If conditions are met, then it adds this pizza into menu.

#### Parameters

<i>nam</i>	Name of the pizza
<i>cst</i>	Cost of the pizza
<i>prof</i>	Profit pizzeria make on this pizza

### 4.7.1.3 DeletionDrinks()

```
void Menu::DeletionDrinks ( )
```

Starts the drink deletion spree.

Deletion process works until theres '-1' on the input.

#### 4.7.1.4 DrinkExist()

```
bool Menu::DrinkExist (
    int id )
```

Checks if the drink exists

##### Parameters

<i>id</i>	ID of the seeked drink
-----------	------------------------

##### Return values

<i>TRUE</i>	if exists
<i>FALSE</i>	if do not exists

#### 4.7.1.5 GetDrink()

```
Drink * Menu::GetDrink (
    int id )
```

##### Parameters

<i>id</i>	Id of the drink
-----------	-----------------

##### Returns

[Drink](#) based on inputed ID

#### 4.7.1.6 GetDrinkIndex()

```
int Menu::GetDrinkIndex (
    int id )
```

##### Parameters

<i>id</i>	ID of the seeked drink
-----------	------------------------

##### Returns

Index (order) of this specific drink in the menu

#### 4.7.1.7 GetDrinksCount()

```
int Menu::GetDrinksCount ( )
```

##### Returns

Total drinks count

#### 4.7.1.8 GetPizza()

```
Pizza * Menu::GetPizza (
    int id )
```

##### Parameters

<i>id</i>	Id of the pizza
-----------	-----------------

##### Returns

[Pizza](#) based on inputed ID

#### 4.7.1.9 GetPizzaIndex()

```
int Menu::GetPizzaIndex (
    int id )
```

##### Parameters

<i>id</i>	ID of the seeked pizza
-----------	------------------------

##### Returns

Index (order) of this speciffic pizza in the menu

#### 4.7.1.10 GetPizzasCount()

```
int Menu::GetPizzasCount ( )
```

##### Returns

Total pizza count

#### 4.7.1.11 PizzaExist()

```
bool Menu::PizzaExist (
    int id )
```

Checks if the drink exists

##### Parameters

<i>id</i>	ID of the seeked drink
-----------	------------------------

##### Return values

<i>TRUE</i>	if exists
<i>FALSE</i>	if do not exists

#### 4.7.1.12 PrintWholeMenu()

```
void Menu::PrintWholeMenu ( )
```

Prints the whole menu.

Prints the whole menu - every pizza thats in the menu, every drink thats in the menu and their ID and name.

#### 4.7.1.13 RemoveDrink()

```
void Menu::RemoveDrink (
    int id )
```

Removes drink from menu

##### Parameters

<i>id</i>	ID of the drink that should be removed
-----------	--

#### 4.7.1.14 RemovePizza()

```
void Menu::RemovePizza (
    int id )
```

Removes pizza from menu

## Parameters

<i>id</i>	ID of the pizza that should be removed
-----------	--

**4.7.1.15 SeekDrinkName()**

```
bool Menu::SeekDrinkName (
    std::string nam )
```

Checks if the drink exists

## Parameters

<i>nam</i>	Name of the seeked drink
------------	--------------------------

## Return values

<i>TRUE</i>	if exists
<i>FALSE</i>	if do not exists

**4.7.1.16 SeekPizzaName()**

```
bool Menu::SeekPizzaName (
    std::string nam )
```

Checks if the pizza exists

## Parameters

<i>nam</i>	Name of the seeked pizza
------------	--------------------------

## Return values

<i>TRUE</i>	if exists
<i>FALSE</i>	if do not exists

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

- Menu.h
- Menu.cpp

## 4.8 Order Class Reference

### Public Member Functions

- [Order](#) ()  
*Order class constructor.*
- [~Order](#) ()  
*Order class destructor.*
- void [AddPizza](#) ([Pizza](#) \*pizza)  
*Adds pizza into order.*
- void [AddDrink](#) ([Drink](#) \*drink)  
*Adds drink into order.*
- int [GetPizzasInOrder](#) ()
- int [GetDrinksInOrder](#) ()
- [Pizza](#) \* [GetPizza](#) (int index)
- [Drink](#) \* [GetDrink](#) (int index)
- int [GetOrderID](#) ()

### 4.8.1 Member Function Documentation

#### 4.8.1.1 AddDrink()

```
void Order::AddDrink (
    Drink * drink )
```

Adds drink into order.

##### Parameters

<i>drink</i>	<a href="#">Drink</a> from the menu, that will be added into order
--------------	--

#### 4.8.1.2 AddPizza()

```
void Order::AddPizza (
    Pizza * pizza )
```

Adds pizza into order.

##### Parameters

<i>pizza</i>	<a href="#">Pizza</a> from the menu, that will be added into order
--------------	--

#### 4.8.1.3 GetDrink()

```
Drink * Order::GetDrink (
    int index )
```

##### Parameters

<i>index</i>	Index of the drink in this exact order
--------------	--

##### Returns

Drink with selected index

#### 4.8.1.4 GetDrinksInOrder()

```
int Order::GetDrinksInOrder ( )
```

##### Returns

Total drinks count in order

#### 4.8.1.5 GetOrderID()

```
int Order::GetOrderID ( )
```

##### Returns

ID of this exact order

#### 4.8.1.6 GetPizza()

```
Pizza * Order::GetPizza (
    int index )
```

##### Parameters

<i>index</i>	Index of the pizza in this exact order
--------------	--

#### Returns

[Pizza](#) with selected index

#### 4.8.1.7 GetPizzasInOrder()

```
int Order::GetPizzasInOrder ( )
```

#### Returns

Total pizzas count in order

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

- Order.h
- Order.cpp

## 4.9 Pizza Class Reference

### Public Member Functions

- [Pizza](#) (std::string nam, int cost, int prof)  
*[Pizza](#) class constructor.*
- [~Pizza](#) ()  
*[Pizzeria](#) class destructor.*
- void [SetPizzaID](#) (int id)  
*Set this pizza ID.*
- std::string [GetPizzaName](#) ()
- int [GetPizzaPrice](#) ()
- int [GetPizzaProfit](#) ()
- int [GetPizzaID](#) ()
- int [GetStaticID](#) ()

### Protected Attributes

- std::string **pizza\_name**
- int **cost**
- int **profit**
- int **pizza\_ID**

### Static Protected Attributes

- static int **static\_pizzaID** = 0

## 4.9.1 Constructor & Destructor Documentation

### 4.9.1.1 Pizza()

```
Pizza::Pizza (
    std::string nam,
    int cost,
    int prof )
```

[Pizza](#) class constructor.



## Parameters

<i>nam</i>	Name of the pizza
<i>cost</i>	Cost of the pizza
<i>prof</i>	<a href="#">Pizzeria</a> 's profit on this pizza

## 4.9.2 Member Function Documentation

### 4.9.2.1 GetPizzaID()

```
int Pizza::GetPizzaID ( )
```

## Returns

This pizza's ID

### 4.9.2.2 GetPizzaName()

```
std::string Pizza::GetPizzaName ( )
```

## Returns

This pizza's name

### 4.9.2.3 GetPizzaPrice()

```
int Pizza::GetPizzaPrice ( )
```

## Returns

This pizza's price

### 4.9.2.4 GetPizzaProfit()

```
int Pizza::GetPizzaProfit ( )
```

## Returns

This pizza's profit

#### 4.9.2.5 GetStaticID()

```
int Pizza::GetStaticID ( )
```

##### Returns

Static pizzas ID

#### 4.9.2.6 SetPizzaID()

```
void Pizza::SetPizzaID (
    int id )
```

Set this pizza ID.

##### Parameters

<i>id</i>	New pizza's ID
-----------	----------------

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

- Pizza.h
- Pizza.cpp

## 4.10 Pizzeria Class Reference

Collaboration diagram for Pizzeria:

### Public Member Functions

- [Pizzeria](#) (std::string nam)  
*class constructor*
- [Pizzeria](#) (std::string nam, int tel)  
*class constructor*
- [Pizzeria](#) (std::string nam, std::string web)  
*Pizzeria class constructor.*
- [Pizzeria](#) (std::string nam, int tel, std::string web)  
*class constructor*
- std::string [GetPizzeriaName](#) ()
- int [GetPizzeriaTel](#) ()
- std::string [GetPizzeriaWeb](#) ()
- int [GetEmployeesCount](#) ()
- [Employee](#) \* [GetEmployee](#) (int empl\_id)
- [Cook](#) \* [GetCook](#) (int empl\_id)
- [Delivery](#) \* [GetDelivery](#) (int empl\_id)

- [Customer](#) \* [GetCustomer](#) (int id)
- int [GetEmployeeIndex](#) (int empl\_id)
- int [GetDeliveryIndex](#) (int empl\_id)
- int [GetCookIndex](#) (int empl\_id)
- void [OrderForCustomer](#) (int cust\_id)  
*Create new order for customer with this ID.*
- void [AddCustomer](#) (std::string nam, std::string addrs)  
*Adds new customer to the pizzeria database.*
- void [AddAsDelivery](#) (std::string nam, bool cook, bool drive)  
*Adds new employee as delivery to the pizzeria database.*
- void [AddAsCook](#) (std::string nam, bool cook, bool drive)  
*Adds new employee as cook to the pizzeria database.*
- void [AddEmployee](#) (std::string nam, bool cook, bool drive)  
*Adds new employee to the pizzeria database.*
- void [IntroducePizzeria](#) ()  
*Prints info about pizzeria.*
- void [PrintEmployees](#) ()  
*Prints all employees.*
- void [PrintDelivery](#) ()  
*Prints all deliveries.*
- void [PrintCook](#) ()  
*Prints all cooks.*
- void [PrintCustomers](#) ()  
*Prints all customers.*
- void [ShiftOrderEmplo](#) (int empl\_id)  
*Shifts order in pizzeria's employee array.*
- void [ShiftOrderCook](#) (int empl\_id)  
*Shifts order in pizzeria's cook array.*
- void [ShiftOrderDeliv](#) (int empl\_id)  
*Shifts order in pizzeria's delivery array.*
- void [DeleteEmployee](#) (int empl\_id)  
*Deletion of exact employee.*
- void [Deletion](#) ()  
*Employee deletion init with header.*
- void [StopProgram](#) ()  
*Programm stopping method.*
- void [MakeSpace](#) ()  
*Makes 18 lines of void.*
- bool [EmplExist](#) (int ID)  
*Checks if seeked employee exists.*
- void [TestVirtual](#) ()  
*Tests of the virtual methods.*
- void [TestPolym](#) ()  
*Testing of the polymorphism.*

## Public Attributes

- [Menu](#) menu

## 4.10.1 Constructor & Destructor Documentation

### 4.10.1.1 Pizzeria() [1/4]

```
Pizzeria::Pizzeria (
    std::string nam )
```

class constructor

#### Parameters

<i>nam</i>	Name of the pizzeria
------------	----------------------

### 4.10.1.2 Pizzeria() [2/4]

```
Pizzeria::Pizzeria (
    std::string nam,
    int tel )
```

class constructor

#### Parameters

<i>nam</i>	Name of the pizzeria
<i>tel</i>	Telephone number of the pizzeria

### 4.10.1.3 Pizzeria() [3/4]

```
Pizzeria::Pizzeria (
    std::string nam,
    std::string web )
```

[Pizzeria](#) class constructor.

#### Parameters

<i>nam</i>	Name of the pizzeria
<i>web</i>	Website of the pizzeria

**4.10.1.4 Pizzeria()** [4/4]

```
Pizzeria::Pizzeria (
    std::string nam,
    int tel,
    std::string web )
```

class constructor

**Parameters**

<i>nam</i>	Name of the pizzeria
<i>tel</i>	Telephone number of the pizzeria
<i>web</i>	Website of the pizzeria

**4.10.2 Member Function Documentation****4.10.2.1 AddAsCook()**

```
void Pizzeria::AddAsCook (
    std::string nam,
    bool cook,
    bool drive )
```

Adds new employee as cook to the pizzeria database.

**Parameters**

<i>nam</i>	New employee's name
<i>cook</i>	Can cook?
<i>drive</i>	Can drive?

**4.10.2.2 AddAsDelivery()**

```
void Pizzeria::AddAsDelivery (
    std::string nam,
    bool cook,
    bool drive )
```

Adds new employee as delivery to the pizzeria database.

**Parameters**

<i>nam</i>	New employee's name
<i>cook</i>	Can cook?
<i>drive</i>	Can drive?

#### 4.10.2.3 AddCustomer()

```
void Pizzeria::AddCustomer (
    std::string nam,
    std::string addrs )
```

Adds new customer to the pizzeria database.

##### Parameters

<i>nam</i>	New customer's name
<i>addr</i> s	New customer's adress

#### 4.10.2.4 AddEmployee()

```
void Pizzeria::AddEmployee (
    std::string nam,
    bool cook,
    bool drive )
```

Adds new employee to the pizzeria database.

##### Parameters

<i>nam</i>	New employee's name
<i>cook</i>	Can cook?
<i>drive</i>	Can drive?

#### 4.10.2.5 DeleteEmployee()

```
void Pizzeria::DeleteEmployee (
    int empl_id )
```

Deletion of exact employee.

##### Parameters

<i>empl</i> ↵ <i>_id</i>	ID of the employee that should be deleted
-----------------------------	---

#### 4.10.2.6 EmplExist()

```
bool Pizzeria::EmplExist (
    int ID )
```

Checks if seeked employee exists.

##### Parameters

<i>ID</i>	ID of the employee
-----------	--------------------

##### Return values

<i>TRUE</i>	if employee exists
<i>FALSE</i>	if employee does not exist

#### 4.10.2.7 GetCook()

```
Cook * Pizzeria::GetCook (
    int empl_id )
```

##### Parameters

<i>empl_id</i>	ID of the cook we wanna seek
----------------	------------------------------

##### Returns

*Cook* with the exact ID as in input

#### 4.10.2.8 GetCookIndex()

```
int Pizzeria::GetCookIndex (
    int empl_id )
```

##### Parameters

<i>empl_id</i>	ID of the cook we wanna seek
----------------	------------------------------

##### Returns

Index in array of the deliveries of this cook

#### 4.10.2.9 GetCustomer()

```
Customer * Pizzeria::GetCustomer (
    int id )
```

##### Parameters

<i>id</i>	ID of the customer we wanna seek
-----------	----------------------------------

##### Returns

ustomer with the exact ID as in input

#### 4.10.2.10 GetDelivery()

```
Delivery * Pizzeria::GetDelivery (
    int empl_id )
```

##### Parameters

<i>empl_id</i>	ID of the delivery we wanna seek
----------------	----------------------------------

##### Returns

[Delivery](#) with the exact ID as in input

#### 4.10.2.11 GetDeliveryIndex()

```
int Pizzeria::GetDeliveryIndex (
    int empl_id )
```

##### Parameters

<i>empl_id</i>	ID of the delivery we wanna seek
----------------	----------------------------------

##### Returns

Index in array of the deliveries of this delivery



#### 4.10.2.12 GetEmployee()

```
Employee * Pizzeria::GetEmployee (
    int empl_id )
```

##### Parameters

<i>empl_id</i>	ID of the employee we wanna seek
----------------	----------------------------------

##### Returns

[Employee](#) with the exact ID as in input

#### 4.10.2.13 GetEmployeeIndex()

```
int Pizzeria::GetEmployeeIndex (
    int empl_id )
```

##### Parameters

<i>empl_id</i>	ID of the employee we wanna seek
----------------	----------------------------------

##### Returns

Index in array of the employees of this employee

#### 4.10.2.14 GetEmployeesCount()

```
int Pizzeria::GetEmployeesCount ( )
```

##### Returns

[Pizzeria](#)'s employees count

#### 4.10.2.15 GetPizzeriaName()

```
std::string Pizzeria::GetPizzeriaName ( )
```

##### Returns

[Pizzeria](#)'s name

#### 4.10.2.16 GetPizzeriaTel()

```
int Pizzeria::GetPizzeriaTel ( )
```

##### Returns

Pizzeria's telephone

#### 4.10.2.17 GetPizzeriaWeb()

```
std::string Pizzeria::GetPizzeriaWeb ( )
```

##### Returns

Pizzeria's website

#### 4.10.2.18 IntroducePizzeria()

```
void Pizzeria::IntroducePizzeria ( )
```

Prints info about pizzeria.

Detailed info about pizzeria. It will print for exmaple telephone number, website adress, all current employees, customers ect.

#### 4.10.2.19 MakeSpace()

```
void Pizzeria::MakeSpace ( )
```

Makes 18 lines of void.

Totally useless, just so this program is not that

#### 4.10.2.20 OrderForCustomer()

```
void Pizzeria::OrderForCustomer (
    int cust_id )
```

Create new order for customer with this ID.

It finds the customer with this ID and creates new order for him Then there will be posibilitty to add whatever pizza and drink from menu. Lastly after his order was created, it will be printed as an final overview.

## Parameters

<i>cust</i> ↔ <i>_id</i>	ID of the customer that will have this order
-----------------------------	--

**4.10.2.21 PrintCook()**

```
void Pizzeria::PrintCook ( )
```

Prints all cooks.

Print every cook of this pizzeria. It prints their ID and name.

**4.10.2.22 PrintCustomers()**

```
void Pizzeria::PrintCustomers ( )
```

Prints all customers.

Print every customer of this pizzeria. It prints their ID and name.

**4.10.2.23 PrintDelivery()**

```
void Pizzeria::PrintDelivery ( )
```

Prints all deliveries.

Print every delivery of this pizzeria. It prints their ID and name.

**4.10.2.24 PrintEmployees()**

```
void Pizzeria::PrintEmployees ( )
```

Prints all employees.

Print every employee of this pizzeria. It prints their ID and name.

**4.10.2.25 ShiftOrderCook()**

```
void Pizzeria::ShiftOrderCook (
    int empl_id )
```

Shifts order in pizzeria's cook array.

## Parameters

<i>empl</i> ↔ _id	ID of the cook, that should be replaced
----------------------	---

**4.10.2.26 ShiftOrderDeliv()**

```
void Pizzeria::ShiftOrderDeliv (
    int empl_id )
```

Shifts order in pizzeria's delivery array.

## Parameters

<i>empl</i> ↔ _id	ID of the delivery, that should be replaced
----------------------	---

**4.10.2.27 ShiftOrderEmplo()**

```
void Pizzeria::ShiftOrderEmplo (
    int empl_id )
```

Shifts order in pizzeria's employee array.

## Parameters

<i>empl</i> ↔ _id	ID of the employee, that should be replaced
----------------------	---

**4.10.2.28 StopProgram()**

```
void Pizzeria::StopProgram ( )
```

Programm stopping method.

Totally useless, just so the program can be stopped and all functionalities can be overviewed.

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

- Pizzeria.h
- Pizzeria.cpp

# Index

~Employee  
Employee, [14](#)

AbstractEmployee, [7](#)

AddAsCook  
Pizzeria, [31](#)

AddAsDelivery  
Pizzeria, [31](#)

AddCustomer  
Pizzeria, [32](#)

AddDrink  
Menu, [19](#)  
Order, [24](#)

AddEmployee  
Pizzeria, [32](#)

AddPizza  
Menu, [19](#)  
Order, [24](#)

CalculateGasMoney  
Delivery, [10](#)

CalculateSalary  
Employee, [14](#)

CanChangePosition  
Employee, [15](#)

CanCook  
Employee, [15](#)

CanDrive  
Employee, [15](#)

Cook, [7](#)  
EmployeeInfo, [7](#)  
GetPizzasMade, [8](#)

Customer, [8](#)  
Customer, [8](#)  
GetCustAddress, [9](#)  
GetCustID, [9](#)  
GetCustName, [9](#)  
GetOrder, [9](#)  
GetOrdersCount, [10](#)

DeleteEmployee  
Pizzeria, [32](#)

DeletionDrinks  
Menu, [19](#)

Delivery, [10](#)  
CalculateGasMoney, [10](#)  
EmployeeInfo, [11](#)  
GetGasMoney, [11](#)  
GetOrdersDelivered, [11](#)

Drink, [12](#)

Drink, [12](#)  
GetDrinkName, [13](#)  
SetDrinkID, [13](#)

DrinkExist  
Menu, [19](#)

EmplExist  
Pizzeria, [32](#)

Employee, [13](#)  
~Employee, [14](#)  
CalculateSalary, [14](#)  
CanChangePosition, [15](#)  
CanCook, [15](#)  
CanDrive, [15](#)  
Employee, [14](#)  
EmployeeInfo, [15](#)  
GetID, [16](#)  
GetName, [16](#)  
GetSalary, [16](#)  
GetWorkedHours, [16](#)  
GetWorkedYears, [16](#)  
SetID, [17](#)

EmployeeInfo  
Cook, [7](#)  
Delivery, [11](#)  
Employee, [15](#)

GetCook  
Pizzeria, [33](#)

GetCookIndex  
Pizzeria, [33](#)

GetCustAddress  
Customer, [9](#)

GetCustID  
Customer, [9](#)

GetCustName  
Customer, [9](#)

GetCustomer  
Pizzeria, [33](#)

GetDelivery  
Pizzeria, [34](#)

GetDeliveryIndex  
Pizzeria, [34](#)

GetDrink  
Menu, [20](#)  
Order, [24](#)

GetDrinkIndex  
Menu, [20](#)

GetDrinkName  
Drink, [13](#)

- GetDrinksCount
  - Menu, [20](#)
- GetDrinksInOrder
  - Order, [25](#)
- GetEmployee
  - Pizzeria, [34](#)
- GetEmployeeIndex
  - Pizzeria, [35](#)
- GetEmployeesCount
  - Pizzeria, [35](#)
- GetGasMoney
  - Delivery, [11](#)
- GetID
  - Employee, [16](#)
- GetName
  - Employee, [16](#)
- GetOrder
  - Customer, [9](#)
- GetOrderID
  - Order, [25](#)
- GetOrdersCount
  - Customer, [10](#)
- GetOrdersDelivered
  - Delivery, [11](#)
- GetPizza
  - Menu, [21](#)
  - Order, [25](#)
- GetPizzaID
  - Pizza, [27](#)
- GetPizzaIndex
  - Menu, [21](#)
- GetPizzaName
  - Pizza, [27](#)
- GetPizzaPrice
  - Pizza, [27](#)
- GetPizzaProfit
  - Pizza, [27](#)
- GetPizzasCount
  - Menu, [21](#)
- GetPizzasInOrder
  - Order, [26](#)
- GetPizzasMade
  - Cook, [8](#)
- GetPizzeriaName
  - Pizzeria, [35](#)
- GetPizzeriaTel
  - Pizzeria, [35](#)
- GetPizzeriaWeb
  - Pizzeria, [36](#)
- GetSalary
  - Employee, [16](#)
- GetStaticID
  - Pizza, [27](#)
- GetWorkedHours
  - Employee, [16](#)
- GetWorkedYears
  - Employee, [16](#)
- IntroducePizzeria
  - Pizzeria, [36](#)
- MakeSpace
  - Pizzeria, [36](#)
- Menu, [18](#)
  - AddDrink, [19](#)
  - AddPizza, [19](#)
  - DeletionDrinks, [19](#)
  - DrinkExist, [19](#)
  - GetDrink, [20](#)
  - GetDrinkIndex, [20](#)
  - GetDrinksCount, [20](#)
  - GetPizza, [21](#)
  - GetPizzaIndex, [21](#)
  - GetPizzasCount, [21](#)
  - PizzaExist, [21](#)
  - PrintWholeMenu, [22](#)
  - RemoveDrink, [22](#)
  - RemovePizza, [22](#)
  - SeekDrinkName, [23](#)
  - SeekPizzaName, [23](#)
- Order, [24](#)
  - AddDrink, [24](#)
  - AddPizza, [24](#)
  - GetDrink, [24](#)
  - GetDrinksInOrder, [25](#)
  - GetOrderID, [25](#)
  - GetPizza, [25](#)
  - GetPizzasInOrder, [26](#)
- OrderForCustomer
  - Pizzeria, [36](#)
- Pizza, [26](#)
  - GetPizzaID, [27](#)
  - GetPizzaName, [27](#)
  - GetPizzaPrice, [27](#)
  - GetPizzaProfit, [27](#)
  - GetStaticID, [27](#)
  - Pizza, [26](#)
  - SetPizzaID, [28](#)
- PizzaExist
  - Menu, [21](#)
- Pizzeria, [28](#)
  - AddAsCook, [31](#)
  - AddAsDelivery, [31](#)
  - AddCustomer, [32](#)
  - AddEmployee, [32](#)
  - DeleteEmployee, [32](#)
  - EmplExist, [32](#)
  - GetCook, [33](#)
  - GetCookIndex, [33](#)
  - GetCustomer, [33](#)
  - GetDelivery, [34](#)
  - GetDeliveryIndex, [34](#)
  - GetEmployee, [34](#)
  - GetEmployeeIndex, [35](#)
  - GetEmployeesCount, [35](#)
  - GetPizzeriaName, [35](#)

- GetPizzeriaTel, [35](#)
- GetPizzeriaWeb, [36](#)
- IntroducePizzeria, [36](#)
- MakeSpace, [36](#)
- OrderForCustomer, [36](#)
- Pizzeria, [30](#)
- PrintCook, [37](#)
- PrintCustomers, [37](#)
- PrintDelivery, [37](#)
- PrintEmployees, [37](#)
- ShiftOrderCook, [37](#)
- ShiftOrderDeliv, [38](#)
- ShiftOrderEmplo, [38](#)
- StopProgram, [38](#)
- PrintCook
  - Pizzeria, [37](#)
- PrintCustomers
  - Pizzeria, [37](#)
- PrintDelivery
  - Pizzeria, [37](#)
- PrintEmployees
  - Pizzeria, [37](#)
- PrintWholeMenu
  - Menu, [22](#)
- RemoveDrink
  - Menu, [22](#)
- RemovePizza
  - Menu, [22](#)
- SeekDrinkName
  - Menu, [23](#)
- SeekPizzaName
  - Menu, [23](#)
- SetDrinkID
  - Drink, [13](#)
- SetID
  - Employee, [17](#)
- SetPizzaID
  - Pizza, [28](#)
- ShiftOrderCook
  - Pizzeria, [37](#)
- ShiftOrderDeliv
  - Pizzeria, [38](#)
- ShiftOrderEmplo
  - Pizzeria, [38](#)
- StopProgram
  - Pizzeria, [38](#)