

Pizzeria

Generated by Doxygen 1.16.1

1 Projekt 1: Objektový model pizzerie	1
1.1 Shrnutí	1
1.2 Hierarchie a třídy	1
1.3 UML Model	1
2 Hierarchical Index	2
2.1 Class Hierarchy	2
3 Data Structure Index	3
3.1 Data Structures	3
4 File Index	3
4.1 File List	3
5 Data Structure Documentation	4
5.1 AlcoholicDrink Class Reference	4
5.1.1 Detailed Description	7
5.1.2 Constructor & Destructor Documentation	8
5.1.3 Member Function Documentation	8
5.1.4 Field Documentation	8
5.2 Customer Class Reference	9
5.2.1 Detailed Description	9
5.2.2 Constructor & Destructor Documentation	10
5.2.3 Member Function Documentation	10
5.2.4 Field Documentation	11
5.3 Drink Class Reference	11
5.3.1 Detailed Description	14
5.3.2 Constructor & Destructor Documentation	15
5.3.3 Member Function Documentation	15
5.3.4 Field Documentation	15
5.4 IMenuItem Class Reference	16
5.4.1 Detailed Description	18
5.4.2 Constructor & Destructor Documentation	18
5.4.3 Member Function Documentation	18
5.5 Ingredient Class Reference	19
5.5.1 Detailed Description	20
5.5.2 Constructor & Destructor Documentation	20
5.5.3 Member Function Documentation	21
5.5.4 Field Documentation	21
5.6 MenuItem Class Reference	22
5.6.1 Detailed Description	24
5.6.2 Constructor & Destructor Documentation	24
5.6.3 Member Function Documentation	24
5.6.4 Field Documentation	25
5.7 Order Class Reference	26

5.7.1 Detailed Description	27
5.7.2 Constructor & Destructor Documentation	28
5.7.3 Member Function Documentation	28
5.7.4 Field Documentation	30
5.8 Pizza Class Reference	31
5.8.1 Detailed Description	34
5.8.2 Constructor & Destructor Documentation	34
5.8.3 Member Function Documentation	35
5.8.4 Field Documentation	36
5.9 Pizzeria Class Reference	37
5.9.1 Detailed Description	38
5.9.2 Constructor & Destructor Documentation	38
5.9.3 Member Function Documentation	39
5.9.4 Field Documentation	39
6 File Documentation	40
6.1 AlcoholicDrink.cpp File Reference	40
6.2 AlcoholicDrink.h File Reference	40
6.2.1 Detailed Description	41
6.3 AlcoholicDrink.h	42
6.4 Customer.cpp File Reference	42
6.5 Customer.h File Reference	42
6.5.1 Detailed Description	43
6.6 Customer.h	43
6.7 Drink.cpp File Reference	44
6.8 Drink.h File Reference	44
6.8.1 Detailed Description	45
6.9 Drink.h	46
6.10 IMenuItem.cpp File Reference	46
6.11 IMenuItem.h File Reference	46
6.11.1 Detailed Description	47
6.12 IMenuItem.h	47
6.13 Ingredient.cpp File Reference	48
6.14 Ingredient.h File Reference	48
6.14.1 Detailed Description	49
6.15 Ingredient.h	49
6.16 main.cpp File Reference	50
6.16.1 Function Documentation	50
6.17 MenuItem.cpp File Reference	50
6.18 MenuItem.h File Reference	51
6.18.1 Detailed Description	52
6.19 MenuItem.h	52
6.20 Order.cpp File Reference	53

6.21 Order.h File Reference	53
6.21.1 Detailed Description	54
6.22 Order.h	54
6.23 Pizza.cpp File Reference	55
6.24 Pizza.h File Reference	55
6.24.1 Detailed Description	56
6.25 Pizza.h	56
6.26 Pizzeria.cpp File Reference	57
6.27 Pizzeria.h File Reference	57
6.27.1 Detailed Description	58
6.28 Pizzeria.h	58
6.29 README.md File Reference	59
Index	61

1 Projekt 1: Objektový model pizzerie

1.1 Shrnutí

Tento projekt představuje softwarovou simulaci fungování pizzerie, pomocí OOP.

1.2 Hierarchie a třídy

Systém se skládá z následujících logických částí:

- Základní rozhraní: [IMenuItem](#)
- Specifické položky (potomci): [MenuItem](#), [Pizza](#), [Drink](#), [AlcoholicDrink](#)
- Správa podniku: Třída [Pizzeria](#) (zajišťuje hlavní logiku)
- Propojení entit: Třída [Order](#) agreguje jednotlivé položky do objednávek.

1.3 UML Model

Vzájemné vztahy, dědičnost a asociace mezi jednotlivými třídami ilustruje níže uvedený grafický návrh.

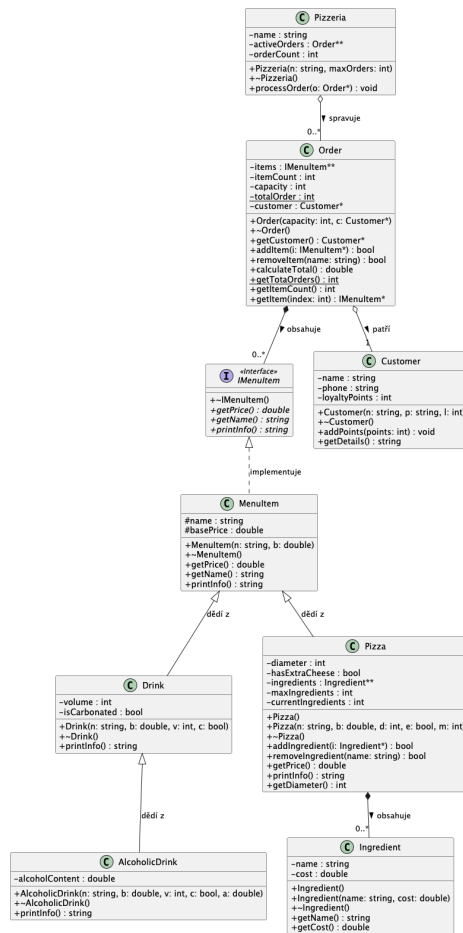


Figure 1 Schéma tříd systému

2 Hierarchical Index

2.1 Class Hierarchy

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

Customer	9
IMenuItem	16
MenuItem	22
Drink	11
AlcoholicDrink	4
Pizza	31
Ingredient	19
Order	26
Pizzeria	37

3 Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

AlcoholicDrink	
Třída reprezentující alkoholický nápoj (dědí z Drink)	4
Customer	
Třída reprezentující zákazníka pizzerie	9
Drink	
Třída reprezentující nápoj v menu (dědí z MenuItem)	11
MenuItem	
Čistě virtuální třída (rozhraní) pro všechny položky v menu	16
Ingredient	
Třída reprezentující přísadu (např. na pizzu)	19
MenuItem	
Základní třída pro konkrétní položky menu (implementuje MenuItem)	22
Order	
Třída reprezentující objednávku zákazníka	26
Pizza	
Třída reprezentující pizzu (dědí z MenuItem)	31
Pizzeria	
Hlavní třída zastupující samotnou pizzerii	37

4 File Index

4.1 File List

Here is a list of all files with brief descriptions:

AlcoholicDrink.cpp	40
AlcoholicDrink.h	
Hlavičkový soubor pro třídu AlcoholicDrink	40
Customer.cpp	42
Customer.h	
Hlavičkový soubor pro třídu Customer	42
Drink.cpp	44
Drink.h	
Hlavičkový soubor pro třídu Drink	44
MenuItem.cpp	46

IMenuItem.h	
Hlavičkový soubor pro rozhraní IMenuItem	46
Ingredient.cpp	48
Ingredient.h	
Hlavičkový soubor pro třídu Ingredient	48
main.cpp	50
MenuItem.cpp	50
MenuItem.h	
Hlavičkový soubor pro základní třídu MenuItem	51
Order.cpp	53
Order.h	
Hlavičkový soubor pro třídu Order	53
Pizza.cpp	55
Pizza.h	
Hlavičkový soubor pro třídu Pizza	55
Pizzeria.cpp	57
Pizzeria.h	
Hlavičkový soubor pro třídu Pizzeria	57

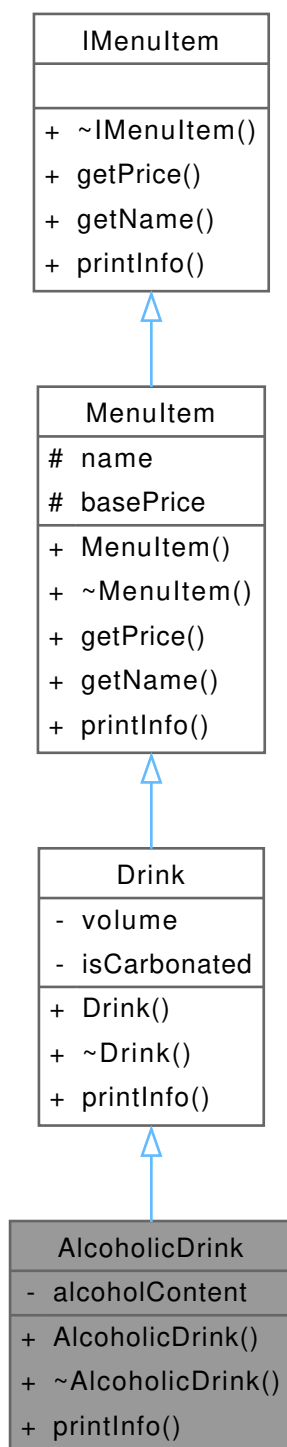
5 Data Structure Documentation

5.1 AlcoholicDrink Class Reference

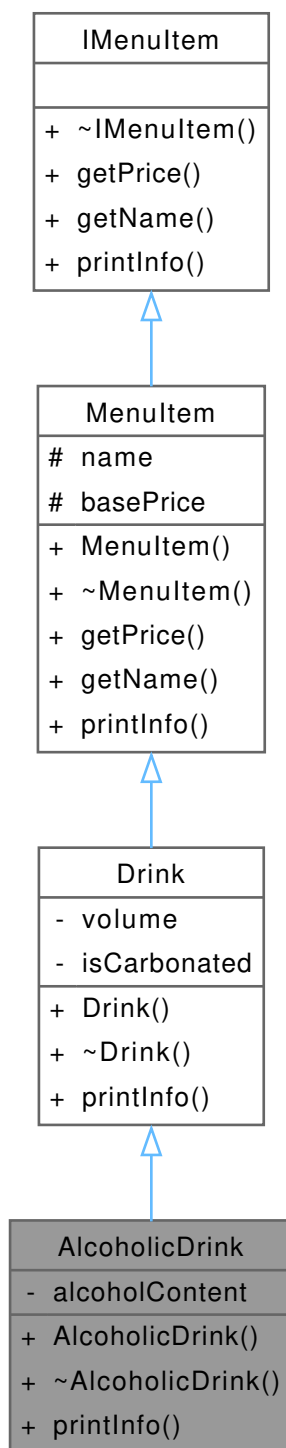
Třída reprezentující alkoholický nápoj (dědí z [Drink](#)).

```
#include <AlcoholicDrink.h>
```

Inheritance diagram for AlcoholicDrink:



Collaboration diagram for AlcoholicDrink:



Public Member Functions

- `AlcoholicDrink` (string n, double b, int v, bool c, double a)
Konstruktor alkoholického nápoje.
- `~AlcoholicDrink` () override
Destruktor.
- string `printInfo` () override
Vrátí detailní informace o alkoholickém nápoji.

Public Member Functions inherited from [Drink](#)

- [Drink](#) (string n, double b, int v, bool c)
Konstruktor nápoje.
- [~Drink](#) () override
Destruktor.
- string [printInfo](#) () override
Vrátí detailní informace o nápoji.

Public Member Functions inherited from [MenuItem](#)

- [MenuItem](#) (string n, double b)
Konstruktor položky menu.
- [~MenuItem](#) () override
Virtuální destruktork.
- double [getPrice](#) () const override
Získá cenu položky.
- string [getName](#) () const override
Získá název položky.
- string [printInfo](#) () override
Vrátí základní informace o položce.

Public Member Functions inherited from [IMenuItem](#)

- virtual [~IMenuItem](#) ()=default
Virtuální destruktork pro správné mazání odvozených tříd.

Private Attributes

- double [alcoholContent](#)
Obsah alkoholu v procentech/promile.

Additional Inherited Members

Protected Attributes inherited from [MenuItem](#)

- string [name](#)
Název položky.
- double [basePrice](#)
Základní cena položky.

5.1.1 Detailed Description

Třída reprezentující alkoholický nápoj (dědí z [Drink](#)).

5.1.2 Constructor & Destructor Documentation

AlcoholicDrink()

```
AlcoholicDrink::AlcoholicDrink (
    string n,
    double b,
    int v,
    bool c,
    double a)
```

Konstruktor alkoholického nápoje.

Parameters

n	Název nápoje.
b	Cena nápoje.
v	Objem v ml.
c	Je sycený?
a	Obsah alkoholu.

~AlcoholicDrink()

```
AlcoholicDrink::~~AlcoholicDrink () [override]
```

Destruktor.

5.1.3 Member Function Documentation

printInfo()

```
string AlcoholicDrink::printInfo () [override], [virtual]
```

Vrátí detailní informace o alkoholickém nápoji.

Returns

Formátovaný řetězec zahrnující i obsah alkoholu.

Implements [IMenuItem](#).

5.1.4 Field Documentation

alcoholContent

```
double AlcoholicDrink::alcoholContent [private]
```

Obsah alkoholu v procentech/promile.

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

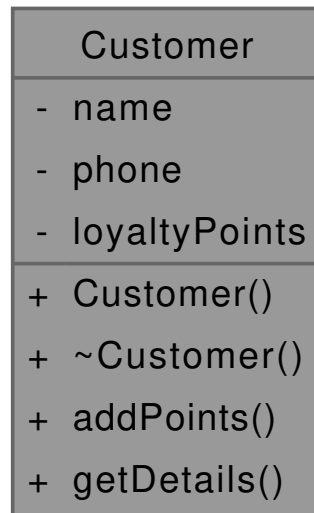
- [AlcoholicDrink.h](#)
- [AlcoholicDrink.cpp](#)

5.2 Customer Class Reference

Třída reprezentující zákazníka pizzerie.

```
#include <Customer.h>
```

Collaboration diagram for Customer:



Public Member Functions

- `Customer` (string n, string p, int l)
Konstruktor pro vytvoření nového zákazníka.
- `~Customer` ()
Destruktor zákazníka.
- void `addPoints` (int points)
Přidá zákazníkovi věrnostní body.
- string `getDetails` ()
Získá naformátované detaily o zákazníkovi.

Private Attributes

- string `name`
Jméno zákazníka.
- string `phone`
Telefonní číslo zákazníka.
- int `loyaltyPoints`
Aktuální počet věrnostních bodů

5.2.1 Detailed Description

Třída reprezentující zákazníka pizzerie.

Uchovává kontaktní údaje zákazníka a jeho věrnostní body.

5.2.2 Constructor & Destructor Documentation

Customer()

```
Customer::Customer (  
    string n,  
    string p,  
    int l)
```

Konstruktor pro vytvoření nového zákazníka.

Parameters

n	Jméno zákazníka.
p	Telefonní číslo.
l	Počáteční stav věrnostních bodů.

~Customer()

```
Customer::~~Customer ()
```

Destruktor zákazníka.

5.2.3 Member Function Documentation

addPoints()

```
void Customer::addPoints (  
    int points)
```

Přidá zákazníkovi věrnostní body.

Parameters

points	Počet bodů k přičtení.
--------	------------------------

getDetails()

```
string Customer::getDetails ()
```

Získá naformátované detaily o zákazníkovi.

Returns

Řetězec obsahující jméno, telefon a body.

5.2.4 Field Documentation

loyaltyPoints

```
int Customer::loyaltyPoints [private]
```

Aktuální počet věrnostních bodů

name

```
string Customer::name [private]
```

Jméno zákazníka.

phone

```
string Customer::phone [private]
```

Telefonní číslo zákazníka.

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

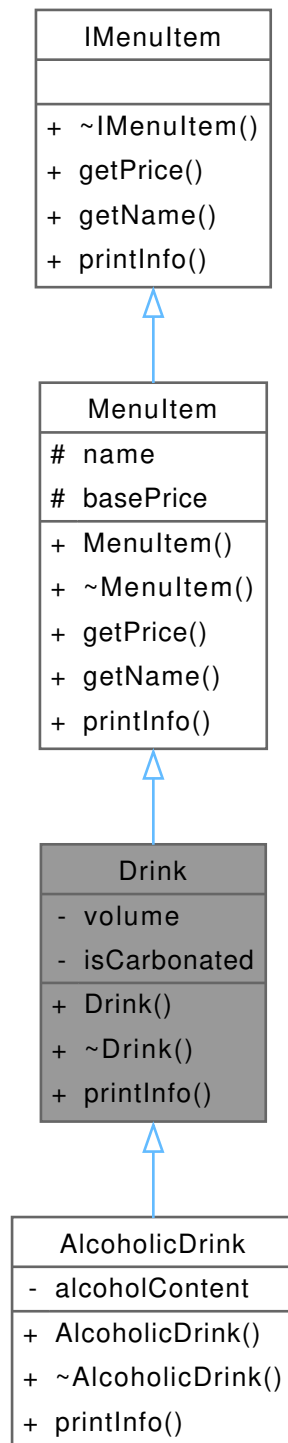
- [Customer.h](#)
- [Customer.cpp](#)

5.3 Drink Class Reference

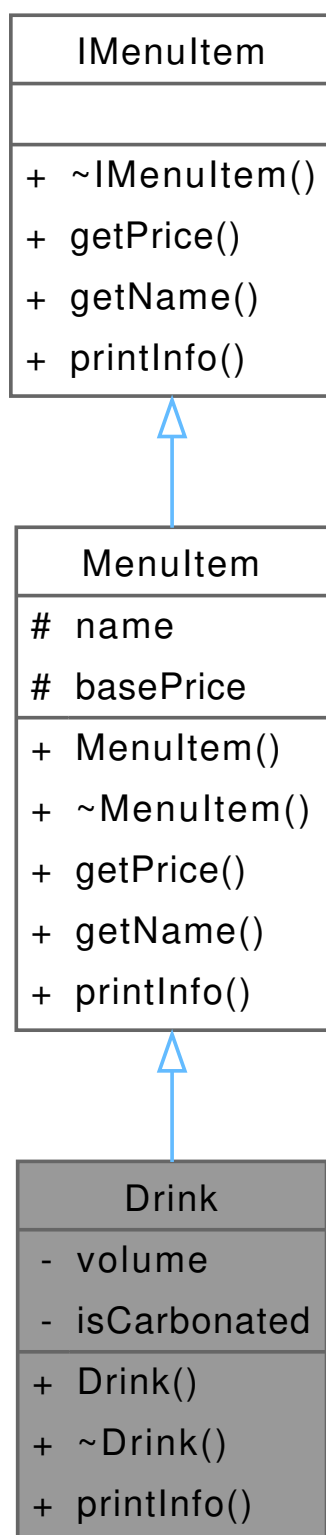
Třída reprezentující nápoj v menu (dědí z [MenuItem](#)).

```
#include <Drink.h>
```

Inheritance diagram for Drink:



Collaboration diagram for Drink:



Public Member Functions

- [Drink](#) (string n, double b, int v, bool c)
Konstruktor nápoje.

- `~Drink ()` override
Destruktor.
- string `printInfo ()` override
Vrátí detailní informace o nápoji.

Public Member Functions inherited from `MenuItem`

- `MenuItem (string n, double b)`
Konstruktor položky menu.
- `~MenuItem ()` override
Virtuální destruktork.
- double `getPrice ()` const override
Získá cenu položky.
- string `getName ()` const override
Získá název položky.
- string `printInfo ()` override
Vrátí základní informace o položce.

Public Member Functions inherited from `IMenuItem`

- virtual `~IMenuItem ()`=default
Virtuální destruktork pro správné mazání odvozených tříd.

Private Attributes

- int `volume`
Objem nápoje v ml.
- bool `isCarbonated`
Indikátor, zda je nápoj sycený

Additional Inherited Members

Protected Attributes inherited from `MenuItem`

- string `name`
Název položky.
- double `basePrice`
Základní cena položky.

5.3.1 Detailed Description

Třída reprezentující nápoj v menu (dědí z `MenuItem`).

5.3.2 Constructor & Destructor Documentation

Drink()

```
Drink::Drink (  
    string n,  
    double b,  
    int v,  
    bool c)
```

Konstruktor nápoje.

Parameters

n	Název nápoje.
b	Cena nápoje.
v	Objem v mililitrech.
c	True, pokud je nápoj sycený, jinak false.

~Drink()

```
Drink::~Drink () [override]
```

Destruktor.

5.3.3 Member Function Documentation

printInfo()

```
string Drink::printInfo () [override], [virtual]
```

Vrátí detailní informace o nápoji.

Returns

Formátovaný řetězec s vlastnostmi nápoje.

Implements [IMenuItem](#).

5.3.4 Field Documentation

isCarbonated

```
bool Drink::isCarbonated [private]
```

Indikátor, zda je nápoj sycený

volume

```
int Drink::volume [private]
```

Objem nápoje v ml.

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

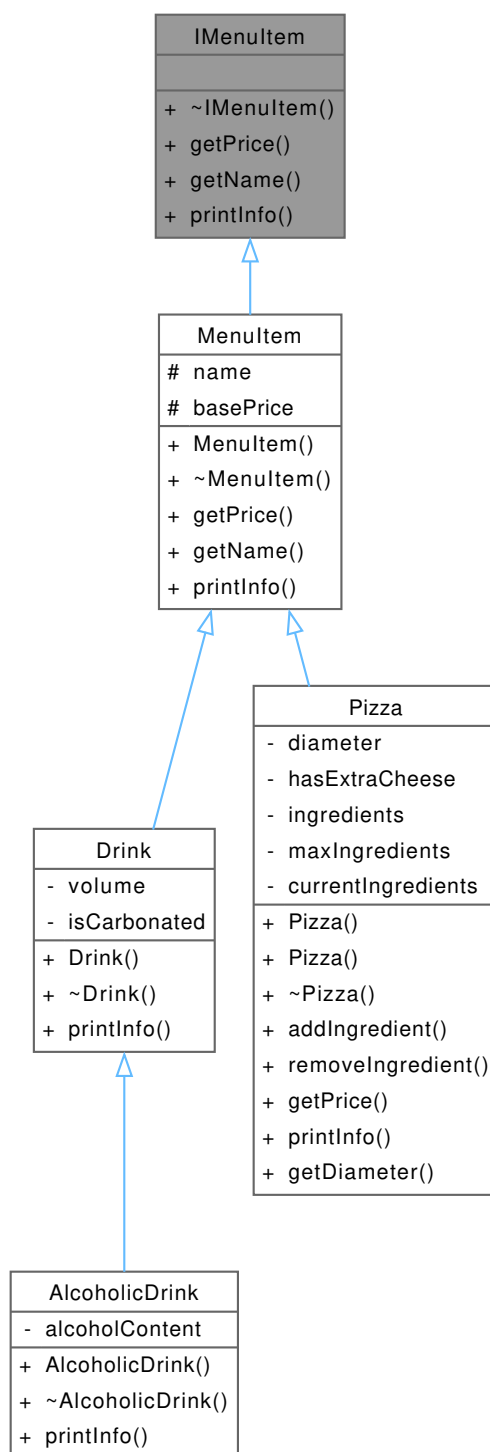
- [Drink.h](#)
- [Drink.cpp](#)

5.4 IMenuItem Class Reference

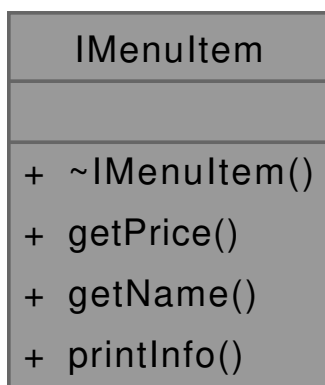
Čistě virtuální třída (rozhraní) pro všechny položky v menu.

```
#include <IMenuItem.h>
```

Inheritance diagram for IMenuItem:



Collaboration diagram for IMenuItem:



Public Member Functions

- virtual [~IMenuItem](#) ()=default
Virtuální destruktork pro správné mazání odvozených tříd.
- virtual double [getPrice](#) () const =0
Získá cenu položky.
- virtual string [getName](#) () const =0
Získá název položky.
- virtual string [printInfo](#) ()=0
Získá formátované informace o položce.

5.4.1 Detailed Description

Čistě virtuální třída (rozhraní) pro všechny položky v menu.

5.4.2 Constructor & Destructor Documentation

[~IMenuItem\(\)](#)

virtual IMenuItem::~~IMenuItem () [virtual], [default]

Virtuální destruktork pro správné mazání odvozených tříd.

5.4.3 Member Function Documentation

[getName\(\)](#)

virtual string IMenuItem::getName () const [pure virtual]

Získá název položky.

Returns

Název položky.

Implemented in [MenuItem](#).

getPrice()

virtual double IMenuItem::getPrice () const [pure virtual]

Získá cenu položky.

Returns

Cena položky typu double.

Implemented in [MenuItem](#), and [Pizza](#).

printInfo()

virtual string IMenuItem::printInfo () [pure virtual]

Získá formátované informace o položce.

Returns

Detailní informace o položce.

Implemented in [AlcoholicDrink](#), [Drink](#), [MenuItem](#), and [Pizza](#).

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

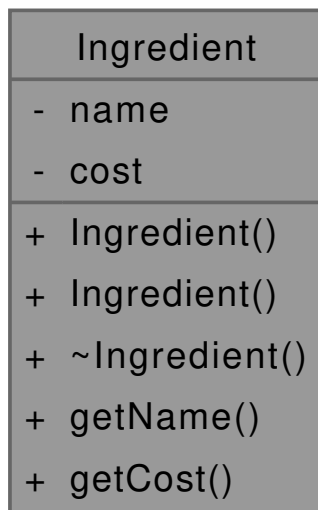
- [IMenuItem.h](#)

5.5 Ingredient Class Reference

Třída reprezentující přísadu (např. na pizzu).

```
#include <Ingredient.h>
```

Collaboration diagram for Ingredient:



Public Member Functions

- `Ingredient ()`
Výchozí konstruktor.
- `Ingredient (string name, double cost)`
Konstruktor s parametry.
- `~Ingredient ()`
Destruktor.
- `string getName ()`
Získá název přísady.
- `double getCost ()`
Získá cenu přísady.

Private Attributes

- `string name`
Název přísady.
- `double cost`
Cena přísady.

5.5.1 Detailed Description

Třída reprezentující přísadu (např. na pizzu).

5.5.2 Constructor & Destructor Documentation

`Ingredient()` [1/2]

`Ingredient::Ingredient ()`

Výchozí konstruktor.

`Ingredient()` [2/2]

`Ingredient::Ingredient (`
 `string name,`
 `double cost)`

Konstruktor s parametry.

Parameters

<code>name</code>	Název přísady.
<code>cost</code>	Cena přísady.

~Ingredient()

Ingredient::~~Ingredient ()

Destruktor.

5.5.3 Member Function Documentation

getCost()

double Ingredient::getCost ()

Získá cenu přísady.

Returns

Cena přísady.

getName()

string Ingredient::getName ()

Získá název přísady.

Returns

Název přísady.

5.5.4 Field Documentation

cost

double Ingredient::cost [private]

Cena přísady.

name

string Ingredient::name [private]

Název přísady.

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

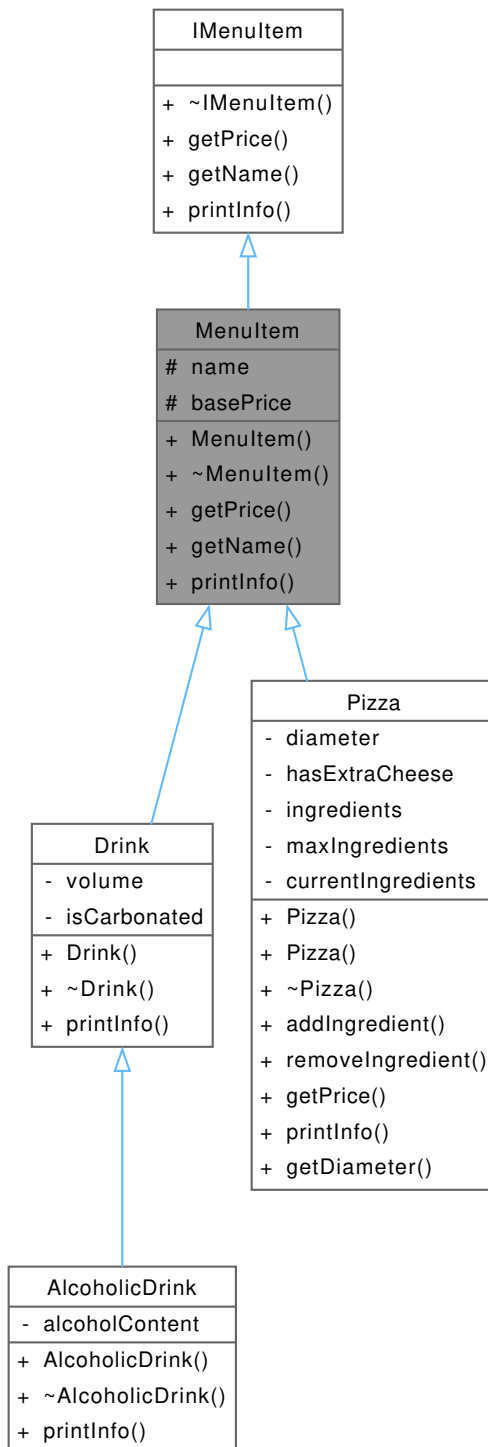
- [Ingredient.h](#)
- [Ingredient.cpp](#)

5.6 MenuItem Class Reference

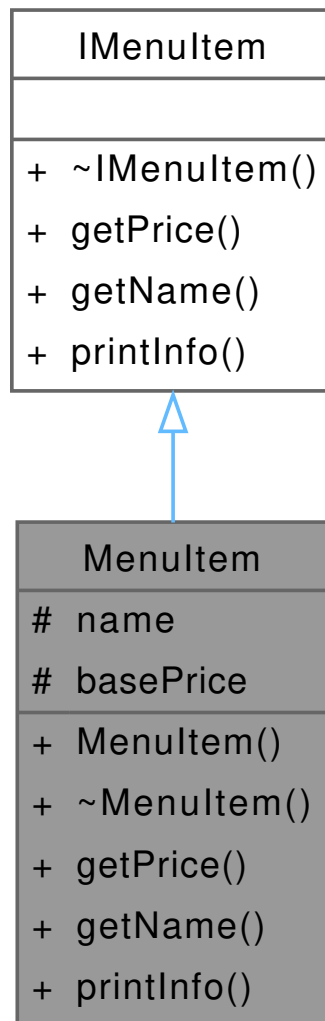
Základní třída pro konkrétní položky menu (implementuje [IMenuItem](#)).

```
#include <MenuItem.h>
```

Inheritance diagram for MenuItem:



Collaboration diagram for MenuItem:



Public Member Functions

- [MenuItem](#) (string n, double b)
Konstruktor položky menu.
- [~MenuItem](#) () override
Virtuální destruktork.
- double [getPrice](#) () const override
Získá cenu položky.
- string [getName](#) () const override
Získá název položky.
- string [printInfo](#) () override
Vrátí základní informace o položce.

Public Member Functions inherited from [IMenuItem](#)

- virtual [~IMenuItem](#) ()=default
Virtuální destruktork pro správné mazání odvozených tříd.

Protected Attributes

- string [name](#)
Název položky.
- double [basePrice](#)
Základní cena položky.

5.6.1 Detailed Description

Základní třída pro konkrétní položky menu (implementuje [IMenuItem](#)).

5.6.2 Constructor & Destructor Documentation

MenuItem()

```
MenuItem::MenuItem (  
    string n,  
    double b)
```

Konstruktor položky menu.

Parameters

n	Název položky.
b	Základní cena.

~MenuItem()

```
MenuItem::~MenuItem () [override]
```

Virtuální destruktork.

5.6.3 Member Function Documentation

getName()

```
string MenuItem::getName () const [override], [virtual]
```

Získá název položky.

Returns

Název položky.

Implements [IMenuItem](#).

getPrice()

```
double MenuItem::getPrice () const [override], [virtual]
```

Získá cenu položky.

Returns

Cena položky.

Implements [IMenuItem](#).

Reimplemented in [Pizza](#).

printInfo()

```
string MenuItem::printInfo () [override], [virtual]
```

Vrátí základní informace o položce.

Returns

Řetězec s názvem a cenou.

Implements [IMenuItem](#).

Reimplemented in [Pizza](#).

5.6.4 Field Documentation

basePrice

```
double MenuItem::basePrice [protected]
```

Základní cena položky.

name

```
string MenuItem::name [protected]
```

Název položky.

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

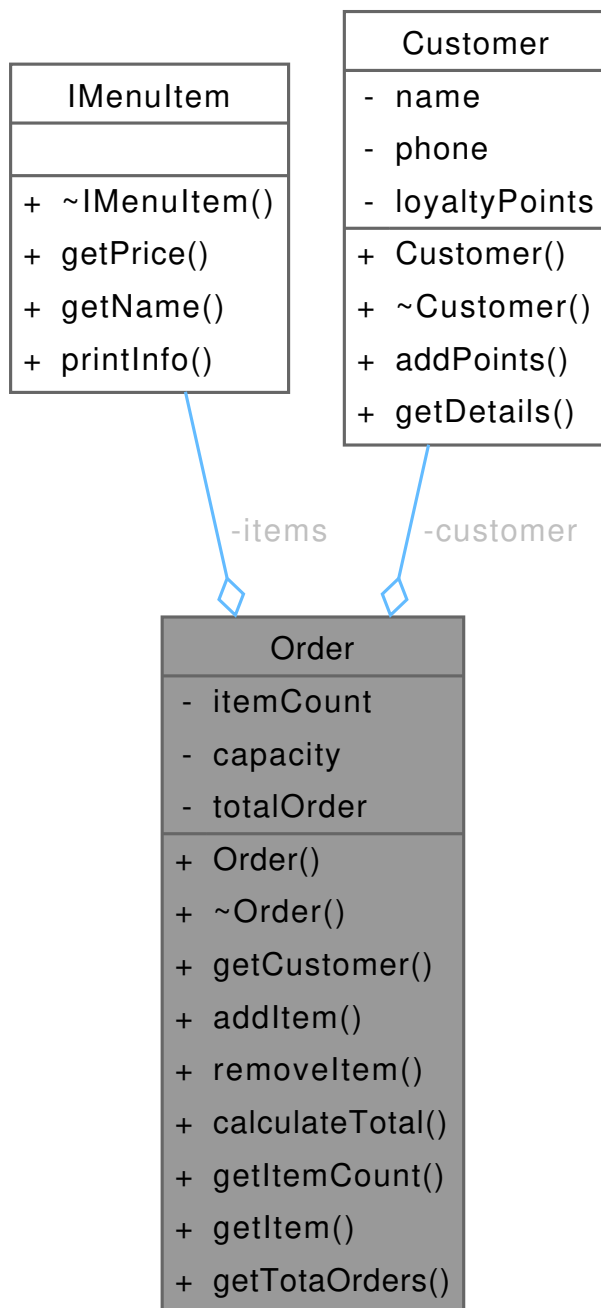
- [MenuItem.h](#)
- [MenuItem.cpp](#)

5.7 Order Class Reference

Třída reprezentující objednávku zákazníka.

```
#include <Order.h>
```

Collaboration diagram for Order:



Public Member Functions

- `Order (int capacity, Customer *c)`
Konstruktor objednávky.
- `~Order ()`

- Destruktor. Uvolní všechny položky objednávky z paměti.
- `Customer * getCustomer ()`
Vrátí ukazatel na zákazníka.
- `bool addItem (IMenuItem *i)`
Přidá položku do objednávky.
- `bool removeItem (string name)`
Odebere položku z objednávky podle jména.
- `double calculateTotal ()`
Vypočítá celkovou cenu všech položek.
- `int getItemCount () const`
Získá aktuální počet položek v objednávce.
- `IMenuItem * getItem (int index) const`
Získá konkrétní položku podle indexu.

Static Public Member Functions

- `static int getTotaOrders ()`
Získá celkový historický počet vytvořených objednávek napříč programem.

Private Attributes

- `IMenuItem ** items`
Dynamické pole ukazatelů na položky objednávky.
- `int itemCount`
Aktuální počet položek.
- `int capacity`
Maximální kapacita objednávky.
- `Customer * customer`
Ukazatel na zákazníka, kterému objednávka patří

Static Private Attributes

- `static int totalOrder =0`
Statické počítadlo všech vytvořených objednávek.

5.7.1 Detailed Description

Třída reprezentující objednávku zákazníka.

Obsahuje dynamické pole položek a odkaz na zákazníka.

5.7.2 Constructor & Destructor Documentation

Order()

```
Order::Order (  
    int capacity,  
    Customer * c)
```

Konstruktor objednávky.

Parameters

capacity	Maximální počet položek v objednávce.
c	Ukazatel na zákazníka.

~Order()

```
Order::~Order ()
```

Destruktor. Uvolní všechny položky objednávky z paměti.

5.7.3 Member Function Documentation

addItem()

```
bool Order::addItem (  
    IMenuItem * i)
```

Přidá položku do objednávky.

Parameters

i	Ukazatel na položku (IMenuItem).
---	----------------------------------

Returns

True, pokud se přidání podařilo, jinak false (plná kapacita).

calculateTotal()

```
double Order::calculateTotal ()
```

Vypočítá celkovou cenu všech položek.

Returns

Součet cen položek v objednávce.

getCustomer()

[Customer](#) * Order::getCustomer ()

Vrátí ukazatel na zákazníka.

Returns

Ukazatel na objekt zákazníka.

getItem()

[MenuItem](#) * Order::getItem (
int index) const

Získá konkrétní položku podle indexu.

Parameters

index	Pozice položky v poli.
-------	------------------------

Returns

Ukazatel na položku, nebo nullptr pokud je index neplatný.

getItemCount()

int Order::getItemCount () const

Získá aktuální počet položek v objednávce.

Returns

Počet položek.

getTotaOrders()

int Order::getTotaOrders () [static]

Získá celkový historický počet vytvořených objednávek napříč programem.

Returns

Počet objednávek (statická proměnná).

removeItem()

```
bool Order::removeItem (  
    string name)
```

Odebere položku z objednávky podle jména.

Parameters

name	Název položky k odebrání.
------	---------------------------

Returns

True, pokud byla položka smazána, jinak false.

5.7.4 Field Documentation

capacity

```
int Order::capacity [private]
```

Maximální kapacita objednávky.

customer

```
Customer\* Order::customer [private]
```

Ukazatel na zákazníka, kterému objednávka patří

itemCount

```
int Order::itemCount [private]
```

Aktuální počet položek.

items

```
Item\* Order::items [private]
```

Dynamické pole ukazatelů na položky objednávky.

totalOrder

```
int Order::totalOrder =0 [static], [private]
```

Statické počítadlo všech vytvořených objednávek.

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

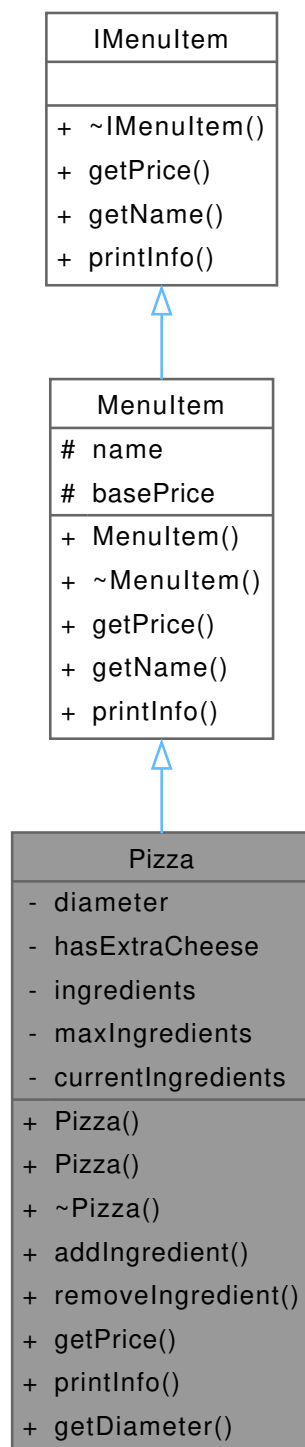
- [Order.h](#)
- [Order.cpp](#)

5.8 Pizza Class Reference

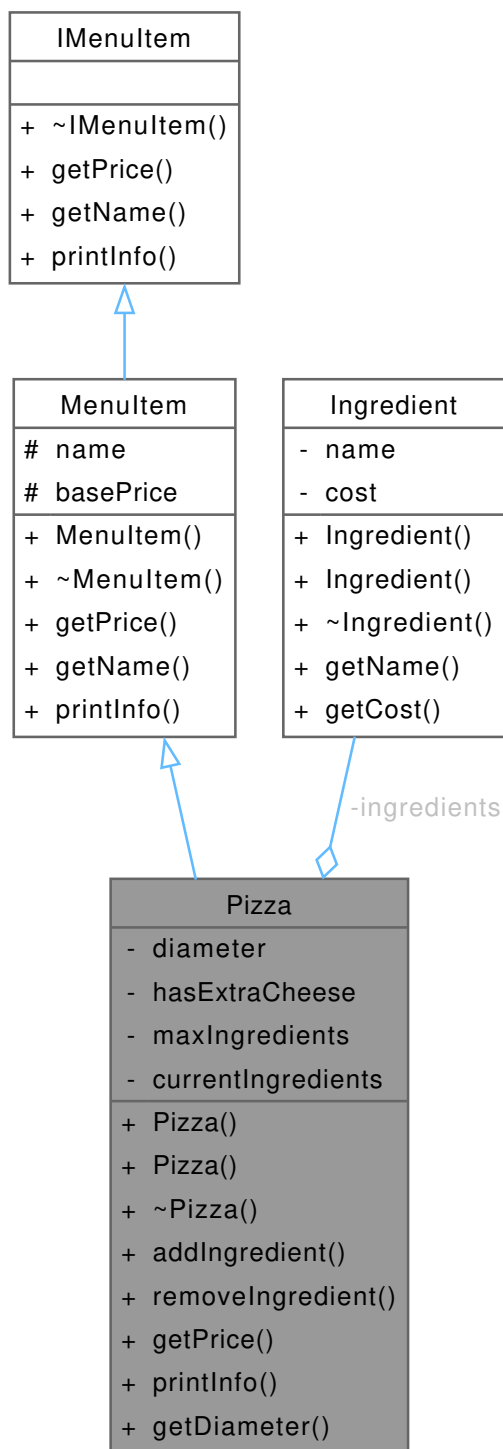
Třída reprezentující pizzu (dědí z [MenuItem](#)).

```
#include <Pizza.h>
```

Inheritance diagram for Pizza:



Collaboration diagram for Pizza:



Public Member Functions

- [Pizza](#) ()
Výchozí konstruktor (nastaví základní hodnoty).
- [Pizza](#) (string n, double b, int d, bool e, int m)
Konstruktor s parametry.
- [~Pizza](#) ()

- Destruktor. Uvolňuje paměť všech alokovaných ingrediencí.
- bool `addIngredient` (`Ingredient *i`)
Přidá ingredienci na pizzu.
- bool `removeIngredient` (string `name`)
Odebere ingredienci z pizzy podle názvu.
- double `getPrice` () const override
Získá celkovou cenu pizzy.
- string `printInfo` () override
Vypíše informace o pizze.
- int `getDiameter` ()
Vrátí průměr pizzy.

Public Member Functions inherited from `MenuItem`

- `MenuItem` (string `n`, double `b`)
Konstruktor položky menu.
- `~MenuItem` () override
Virtuální destruktork.
- string `getName` () const override
Získá název položky.

Public Member Functions inherited from `IMenuItem`

- virtual `~IMenuItem` ()=default
Virtuální destruktork pro správné mazání odvozených tříd.

Private Attributes

- int `diameter`
Průměr pizzy v cm.
- bool `hasExtraCheese`
Indikátor extra sýra.
- `Ingredient ** ingredients`
Dynamické pole ukazatelů na ingredience.
- int `maxIngredients`
Maximální kapacita pole ingrediencí
- int `currentIngredients` = 0
Aktuální počet přidanych ingrediencí

Additional Inherited Members

Protected Attributes inherited from `MenuItem`

- string `name`
Název položky.
- double `basePrice`
Základní cena položky.

5.8.1 Detailed Description

Třída reprezentující pizzu (dědí z [MenuItem](#)).

Umožňuje přidávat a odebírat dynamicky alokované ingredience.

5.8.2 Constructor & Destructor Documentation

Pizza() [1/2]

Pizza::Pizza ()

Výchozí konstruktor (nastaví základní hodnoty).

Pizza() [2/2]

```
Pizza::Pizza (  
    string n,  
    double b,  
    int d,  
    bool e,  
    int m)
```

Konstruktor s parametry.

Parameters

n	Název pizzy.
b	Základní cena.
d	Průměr v cm.
e	True, pokud má extra sýr.
m	Maximální počet ingrediencí.

~Pizza()

Pizza::~Pizza ()

Destruktor. Uvolňuje paměť všech alokovaných ingrediencí.

5.8.3 Member Function Documentation

addIngredient()

```
bool Pizza::addIngredient (  
    Ingredient * i)
```

Přidá ingredienci na pizzu.

Parameters

i	Ukazatel na novou ingredienci.
---	--------------------------------

Returns

True, pokud se přidání povedlo, jinak false (při překročení kapacity).

getDiameter()

```
int Pizza::getDiameter ()
```

Vrátí průměr pizzy.

Returns

Průměr v cm.

getPrice()

```
double Pizza::getPrice () const  [override], [virtual]
```

Získá celkovou cenu pizzy.

Returns

Cena pizzy.

Reimplemented from [MenuItem](#).

printInfo()

```
string Pizza::printInfo ()  [override], [virtual]
```

Vypíše informace o pizze.

Returns

Detailní textový popis pizzy.

Reimplemented from [MenuItem](#).

removeIngredient()

```
bool Pizza::removeIngredient (  
    string name)
```

Odebere ingredienci z pizzy podle názvu.

Parameters

name	Název ingredience k odebrání.
----------------------	-------------------------------

Returns

True, pokud byla nalezena a odebrána, jinak false.

5.8.4 Field Documentation

currentIngredients

```
int Pizza::currentIngredients = 0 [private]
```

Aktuální počet přidanych ingrediencí

diameter

```
int Pizza::diameter [private]
```

Průměr pizzy v cm.

hasExtraCheese

```
bool Pizza::hasExtraCheese [private]
```

Indikátor extra sýra.

ingredients

```
Ingredient** Pizza::ingredients [private]
```

Dynamické pole ukazatelů na ingredience.

maxIngredients

```
int Pizza::maxIngredients [private]
```

Maximální kapacita pole ingrediencí

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

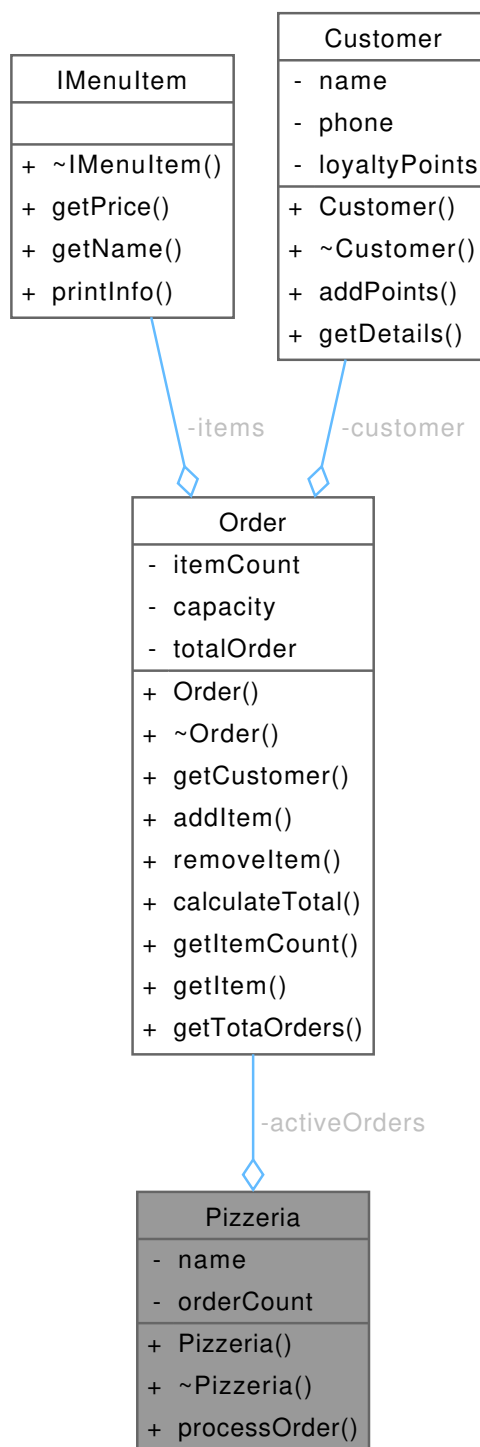
- [Pizza.h](#)
- [Pizza.cpp](#)

5.9 Pizzeria Class Reference

Hlavní třída zastupující samotnou pizzerii.

```
#include <Pizzeria.h>
```

Collaboration diagram for Pizzeria:



Public Member Functions

- [Pizzeria](#) (string n, int maxOrders)
Konstruktor pizzerie.
- [~Pizzeria](#) ()
Destruktor. Uvolní paměť pole objednávek.
- void [processOrder](#) ([Order](#) *o)
Zpracuje přijatou objednávku.

Private Attributes

- string [name](#)
Název pizzerie.
- [Order](#) ** [activeOrders](#)
Pole aktivních objednávek (v této verzi nepoužito naplno).
- int [orderCount](#)
Počet aktivních objednávek.

5.9.1 Detailed Description

Hlavní třída zastupující samotnou pizzerii.

Slouží ke správě a zpracování objednávek zákazníků.

5.9.2 Constructor & Destructor Documentation

Pizzeria()

```
Pizzeria::Pizzeria (  
    string n,  
    int maxOrders)
```

Konstruktor pizzerie.

Parameters

n	Název pizzerie.
maxOrders	Maximální kapacita aktivních objednávek.

~Pizzeria()

```
Pizzeria::~Pizzeria ()
```

Destruktor. Uvolní paměť pole objednávek.

5.9.3 Member Function Documentation

processOrder()

```
void Pizzeria::processOrder (  
    Order * o)
```

Zpracuje přijatou objednávku.

Provede výpis informací o objednávce, celkové ceně a postará se o její smazání z paměti (dokončení životního cyklu).

Parameters

o	Ukazatel na zpracovávanou objednávku.
---	---------------------------------------

5.9.4 Field Documentation

activeOrders

```
Order** Pizzeria::activeOrders [private]
```

Pole aktivních objednávek (v této verzi nepoužito naplno).

name

```
string Pizzeria::name [private]
```

Název pizzerie.

orderCount

```
int Pizzeria::orderCount [private]
```

Počet aktivních objednávek.

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

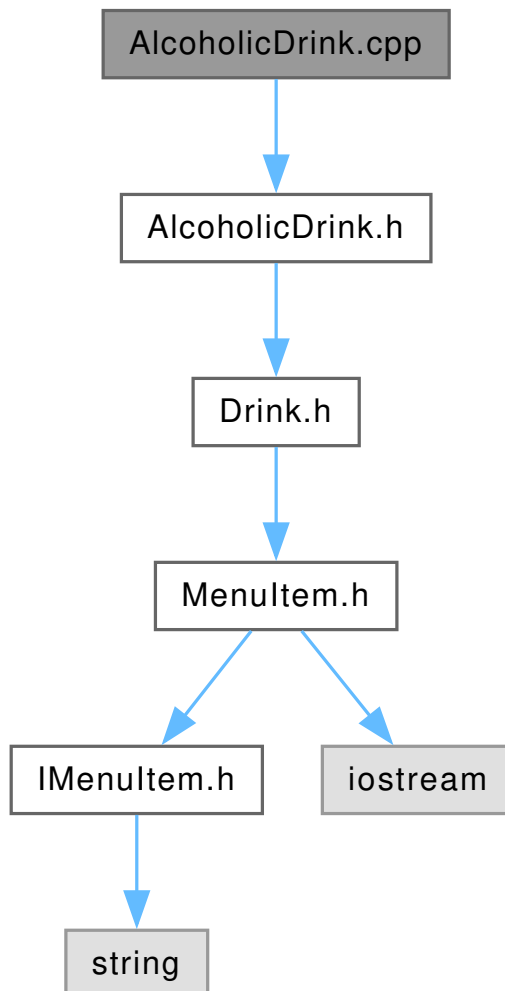
- [Pizzeria.h](#)
- [Pizzeria.cpp](#)

6 File Documentation

6.1 AlcoholicDrink.cpp File Reference

```
#include "AlcoholicDrink.h"
```

Include dependency graph for AlcoholicDrink.cpp:

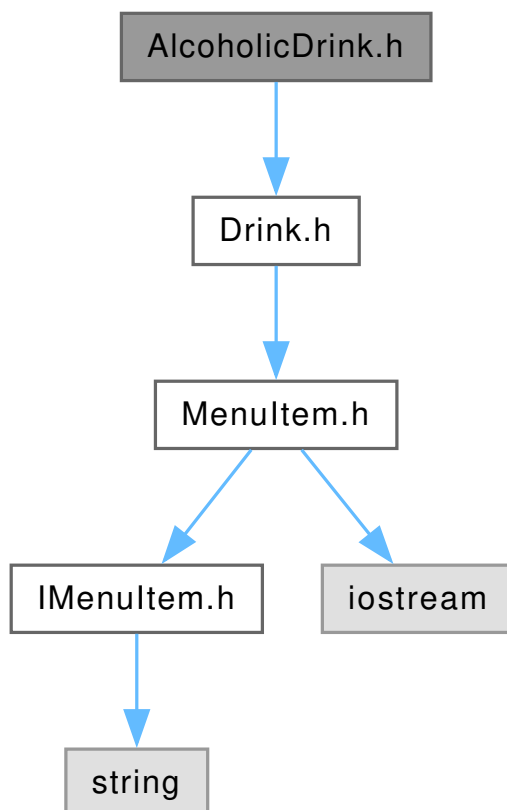


6.2 AlcoholicDrink.h File Reference

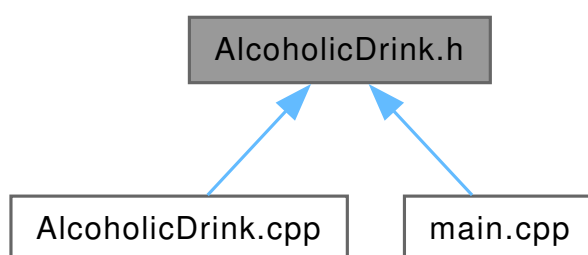
Hlavičkový soubor pro třídu [AlcoholicDrink](#).

```
#include "Drink.h"
```

Include dependency graph for AlcoholicDrink.h:



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



Data Structures

- class [AlcoholicDrink](#)
Třída reprezentující alkoholický nápoj (dědí z [Drink](#)).

6.2.1 Detailed Description

Hlavičkový soubor pro třídu [AlcoholicDrink](#).

6.3 AlcoholicDrink.h

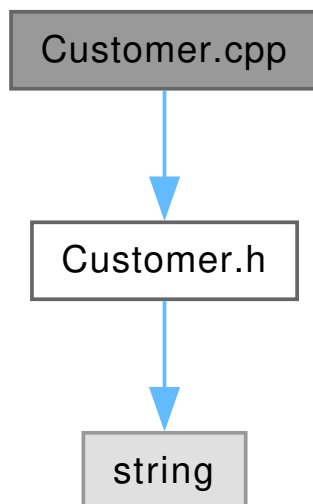
[Go to the documentation of this file.](#)

```
00001
00005
00006 #ifndef PROOP_ALCOHOLICDRINK_H
00007 #define PROOP_ALCOHOLICDRINK_H
00008 #include "Drink.h"
00009
00014 class AlcoholicDrink : public Drink {
00015 private:
00016     double alcoholContent;
00017
00018 public:
00027     AlcoholicDrink(string n, double b, int v, bool c, double a);
00028
00032     ~AlcoholicDrink() override;
00033
00038     string printInfo() override;
00039 };
00040
00041 #endif //PROOP_ALCOHOLICDRINK_H
```

6.4 Customer.cpp File Reference

#include "Customer.h"

Include dependency graph for Customer.cpp:

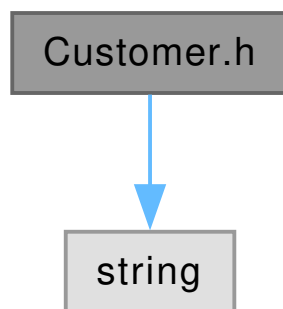


6.5 Customer.h File Reference

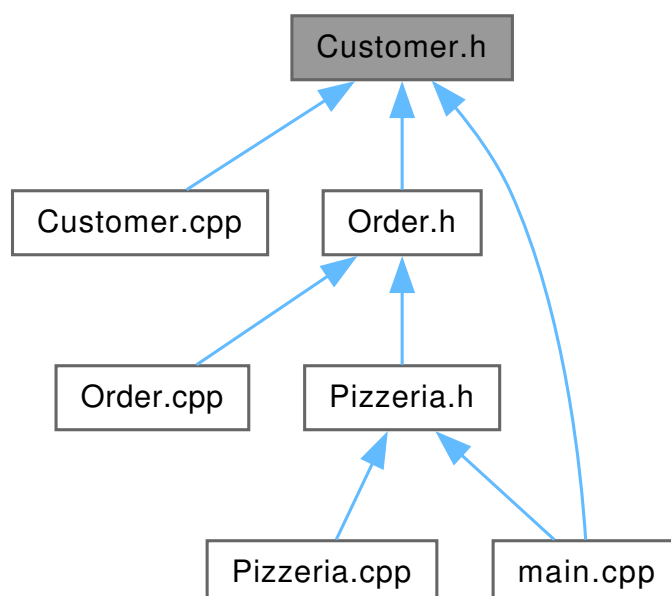
Hlavičkový soubor pro třídu [Customer](#).

```
#include <string>
```

Include dependency graph for Customer.h:



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



Data Structures

- class [Customer](#)
Třída reprezentující zákazníka pizzerie.

6.5.1 Detailed Description

Hlavičkový soubor pro třídu [Customer](#).

6.6 Customer.h

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

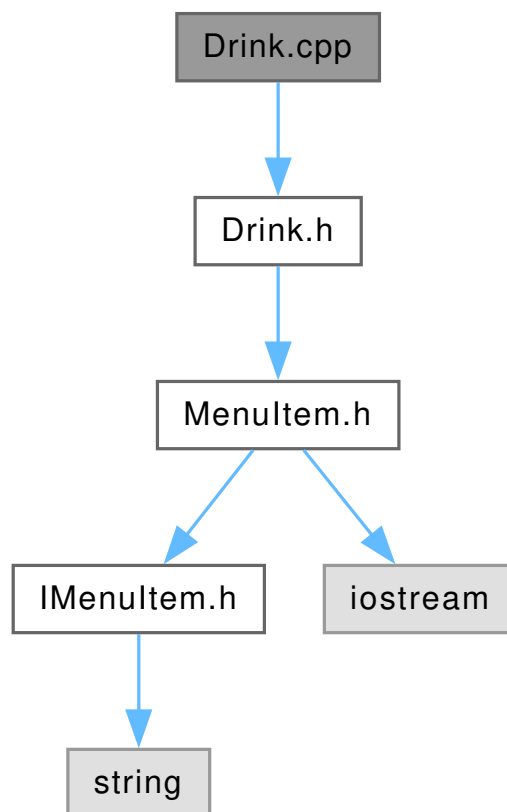
```
00001
00005
00006 #ifndef PROOP_CUSTOMER_H
```

```
00007 #define PROOP_CUSTOMER_H
00008 #include <string>
00009 using namespace std;
00010
00017 class Customer {
00018     string name;
00019     string phone;
00020     int loyaltyPoints;
00021
00022 public:
00029     Customer(string n, string p, int l);
00030
00034     ~Customer();
00035
00040     void addPoints(int points);
00041
00046     string getDetails();
00047 };
00048
00049 #endif //PROOP_CUSTOMER_H
```

6.7 Drink.cpp File Reference

#include "Drink.h"

Include dependency graph for Drink.cpp:

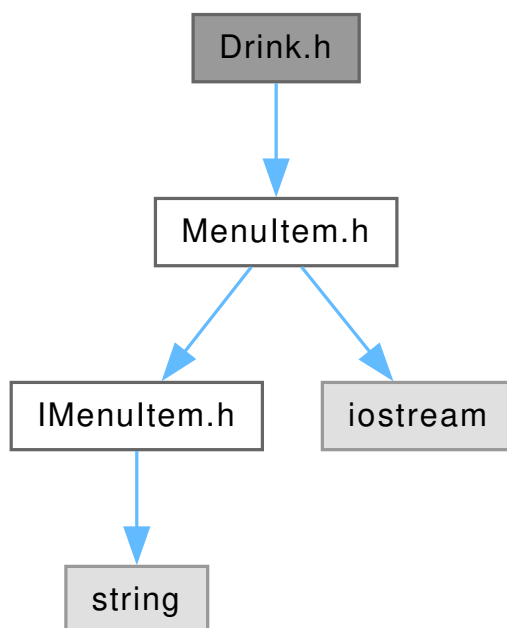


6.8 Drink.h File Reference

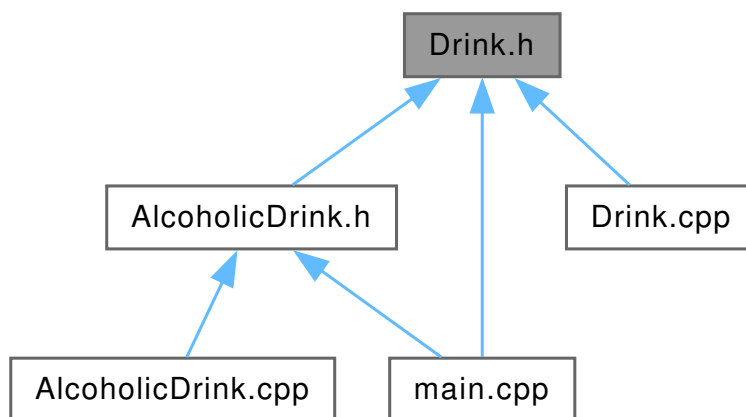
Hlavičkový soubor pro třídu [Drink](#).

```
#include "MenuItem.h"
```

Include dependency graph for Drink.h:



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



Data Structures

- class [Drink](#)
Třída reprezentující nápoj v menu (dědí z [MenuItem](#)).

6.8.1 Detailed Description

Hlavičkový soubor pro třídu [Drink](#).

6.9 Drink.h

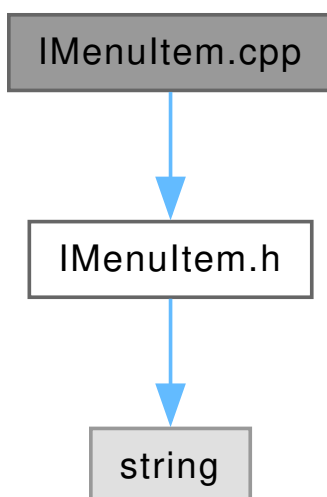
[Go to the documentation of this file.](#)

```
00001
00005
00006 #ifndef PROOP_DRINK_H
00007 #define PROOP_DRINK_H
00008 #include "MenuItem.h"
00009
00014 class Drink : public MenuItem {
00015 private:
00016     int volume;
00017     bool isCarbonated;
00018
00019 public:
00027     Drink(string n, double b, int v, bool c);
00028
00032     ~Drink() override;
00033
00038     string printInfo() override;
00039 };
00040
00041 #endif //PROOP_DRINK_H
```

6.10 IMenuItem.cpp File Reference

```
#include "IMenuItem.h"
```

Include dependency graph for IMenuItem.cpp:

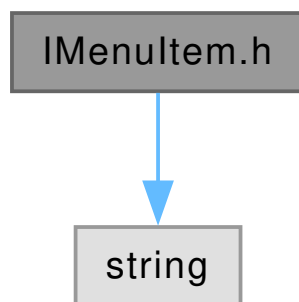


6.11 IMenuItem.h File Reference

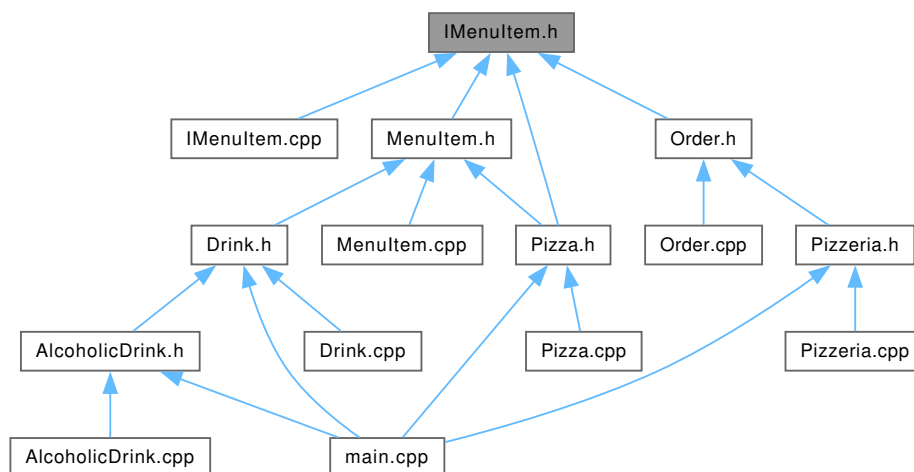
Hlavičkový soubor pro rozhraní [IMenuItem](#).

```
#include <string>
```

Include dependency graph for IMenuItem.h:



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



Data Structures

- class [IMenuItem](#)

Čistě virtuální třída (rozhraní) pro všechny položky v menu.

6.11.1 Detailed Description

Hlavičkový soubor pro rozhraní [IMenuItem](#).

6.12 IMenuItem.h

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

```

00001
00002
00003 #ifndef PROOP_IMENUIITEM_H
00004 #define PROOP_IMENUIITEM_H
00005 #include <string>
00006 using namespace std;
00007
00008 class IMenuItem {
00009 public:

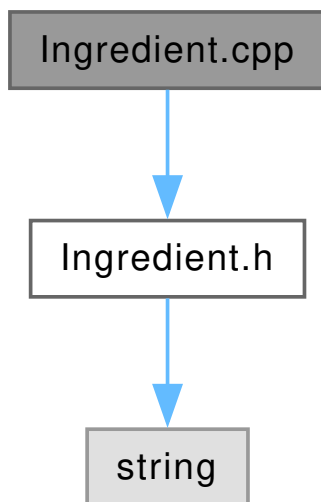
```

```
00020     virtual ~MenuItem() = default;
00021
00026     virtual double getPrice() const = 0;
00027
00032     virtual string getName() const = 0;
00033
00038     virtual string printInfo() = 0;
00039 };
00040
00041 #endif //PROOP_IMENUITEM_H
```

6.13 Ingredient.cpp File Reference

```
#include "Ingredient.h"
```

Include dependency graph for Ingredient.cpp:

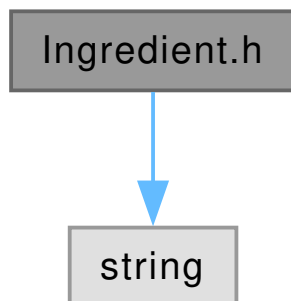


6.14 Ingredient.h File Reference

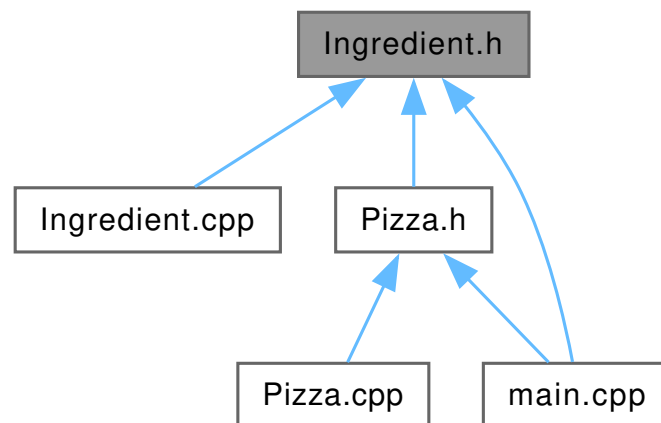
Hlavičkový soubor pro třídu [Ingredient](#).

```
#include <string>
```

Include dependency graph for Ingredient.h:



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



Data Structures

- class [Ingredient](#)
Třída reprezentující přísadu (např. na pizzu).

6.14.1 Detailed Description

Hlavičkový soubor pro třídu [Ingredient](#).

6.15 Ingredient.h

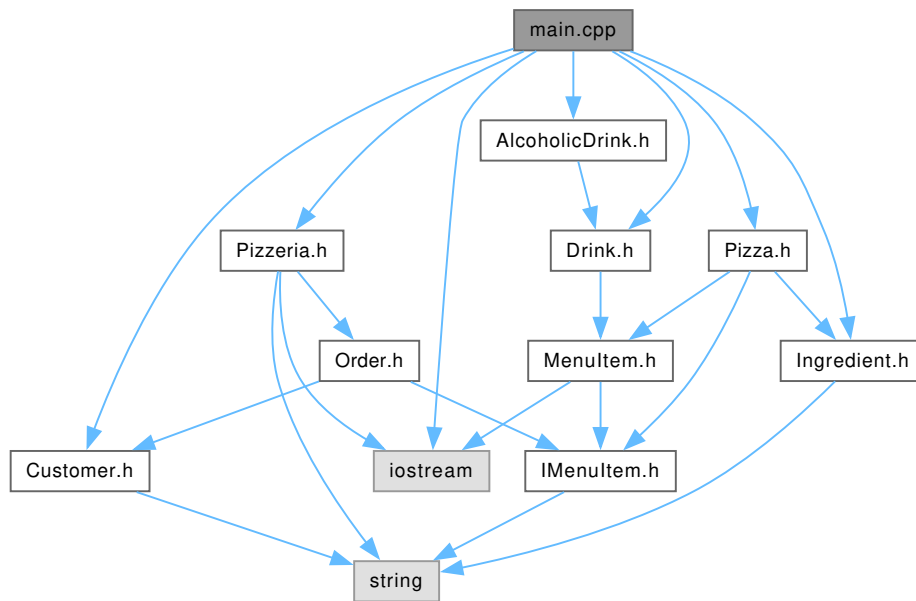
[Go to the documentation of this file.](#)

```

00001
00002
00003 #ifndef PROOP_INGREDIENT_H
00004 #define PROOP_INGREDIENT_H
00005 #include <string>
00006 using namespace std;
00007
00008 class Ingredient {
00009 private:
00010     string name;
00011     double cost;
00012
00013 public:
00014     Ingredient();
00015
00016     Ingredient(string name, double cost);
00017
00018     ~Ingredient();
00019
00020     string getName();
00021
00022     double getCost();
00023 };
00024
00025 #endif //PROOP_INGREDIENT_H
  
```

6.16 main.cpp File Reference

```
#include <iostream>
#include "Pizzeria.h"
#include "Customer.h"
#include "Pizza.h"
#include "Drink.h"
#include "Ingredient.h"
#include "AlcoholicDrink.h"
Include dependency graph for main.cpp:
```



Functions

- `int main ()`

6.16.1 Function Documentation

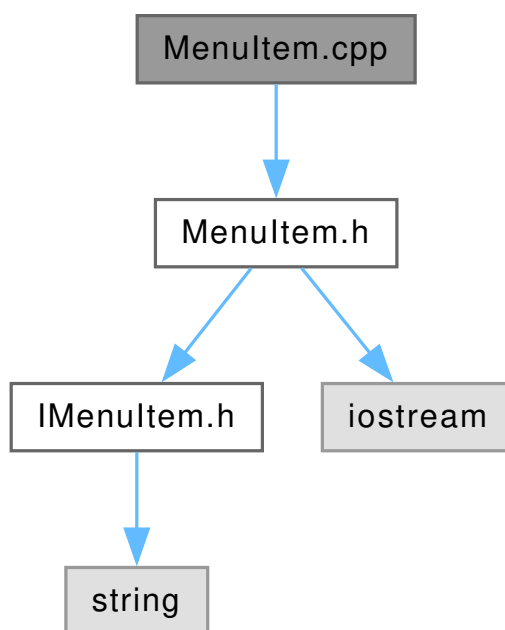
`main()`

`int main ()`

6.17 MenuItem.cpp File Reference

```
#include "MenuItem.h"
```

Include dependency graph for MenuItem.cpp:

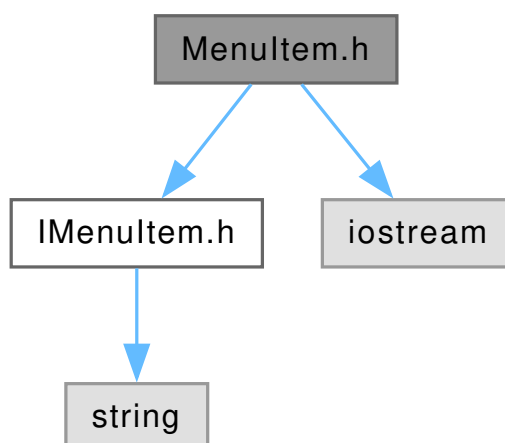


6.18 MenuItem.h File Reference

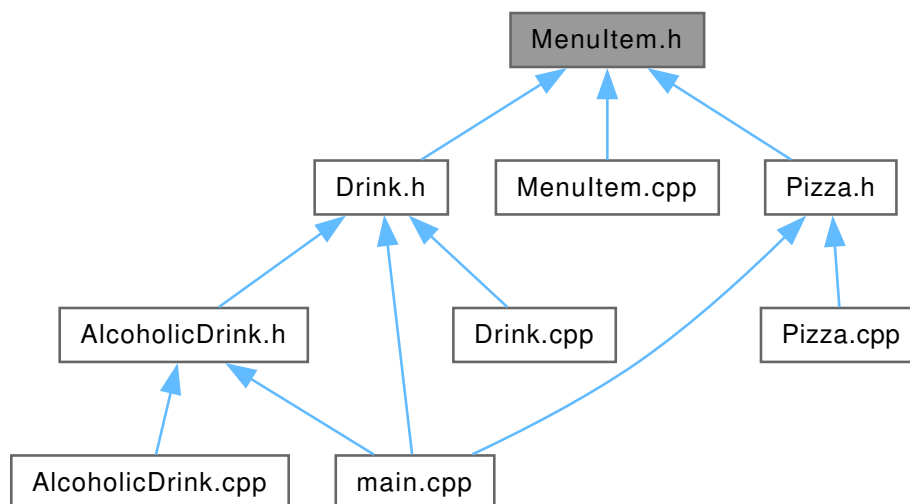
Hlavičkový soubor pro základní třídu [MenuItem](#).

```
#include "IMenuItem.h"  
#include <iostream>
```

Include dependency graph for MenuItem.h:



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



Data Structures

- class [MenuItem](#)

Základní třída pro konkrétní položky menu (implementuje [IMenuItem](#)).

6.18.1 Detailed Description

Hlavičkový soubor pro základní třídu [MenuItem](#).

6.19 MenuItem.h

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

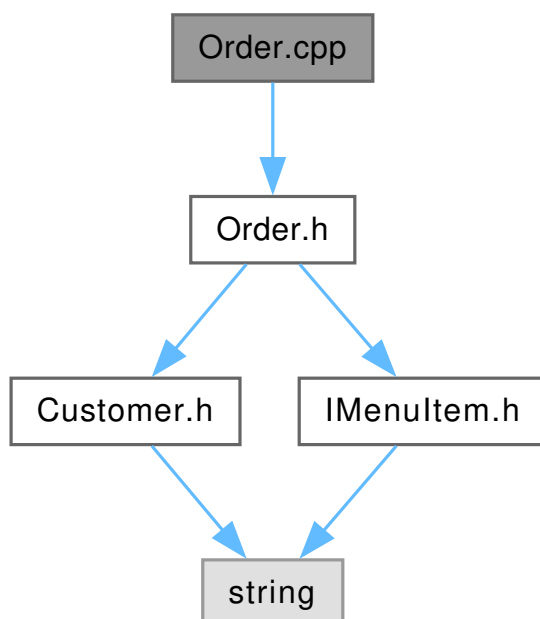
```

00001
00005
00006 #ifndef PROOP_MENUITEM_H
00007 #define PROOP_MENUITEM_H
00008 #include "IMenuItem.h"
00009 #include <iostream>
00010
00015 class MenuItem : public IMenuItem {
00016 protected:
00017     string name;
00018     double basePrice;
00019
00020 public:
00026     MenuItem(string n, double b);
00027
00031     ~MenuItem() override;
00032
00037     double getPrice() const override;
00038
00043     string getName() const override;
00044
00049     string printInfo() override;
00050 };
00051
00052 #endif //PROOP_MENUITEM_H
  
```

6.20 Order.cpp File Reference

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

Include dependency graph for Order.cpp:



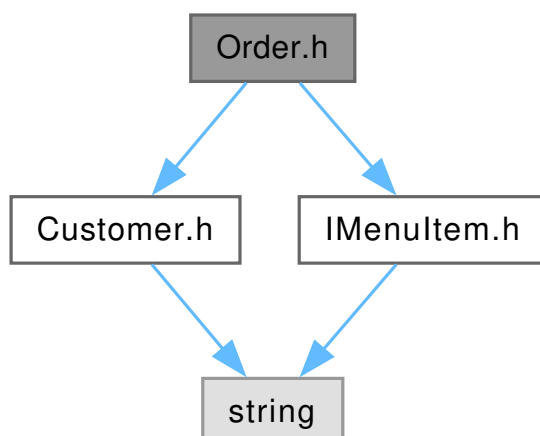
6.21 Order.h File Reference

Hlavičkový soubor pro třídu [Order](#).

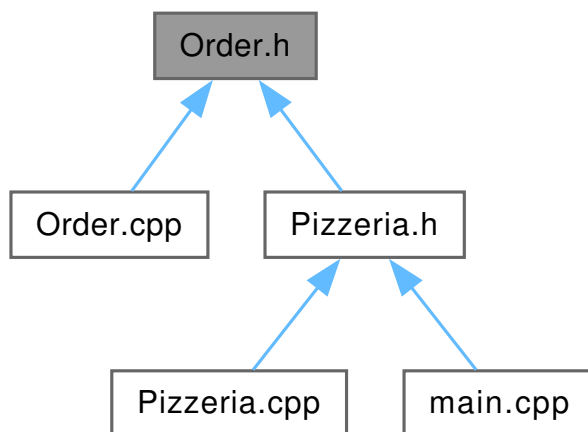
```
#include "Customer.h"
```

```
#include "IMenuItem.h"
```

Include dependency graph for Order.h:



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



Data Structures

- class [Order](#)
Třída reprezentující objednávku zákazníka.

6.21.1 Detailed Description

Hlavičkový soubor pro třídu [Order](#).

6.22 Order.h

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

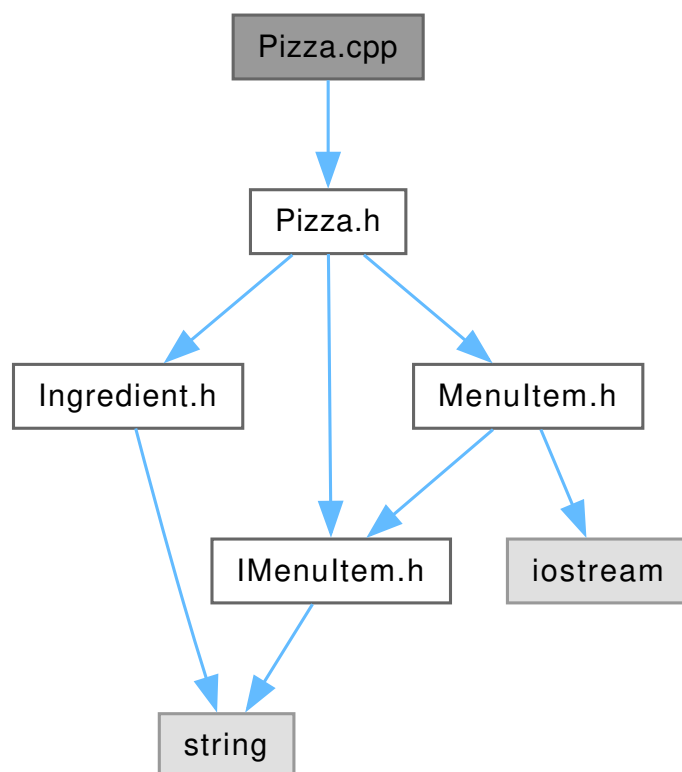
```

00001
00005
00006 #ifndef PROOP_ORDER_H
00007 #define PROOP_ORDER_H
00008 #include "Customer.h"
00009 #include "MenuItem.h"
00010
00017 class Order {
00018 private:
00019     MenuItem** items;
00020     int itemCount;
00021     int capacity;
00022     static int totalOrder;
00023     Customer* customer;
00024
00025 public:
00031     Order(int capacity, Customer * c);
00032
00036     ~Order();
00037
00042     Customer* getCustomer();
00043
00049     bool addItem(MenuItem * i);
00050
00056     bool removeItem(string name);
00057
00062     double calculateTotal();
00063
00068     static int getTotalOrders();
00069
00074     int getItemCount() const;
00075
00081     MenuItem* getItem(int index) const;
00082 };
00083
00084 #endif //PROOP_ORDER_H
  
```

6.23 Pizza.cpp File Reference

```
#include "Pizza.h"
```

Include dependency graph for Pizza.cpp:

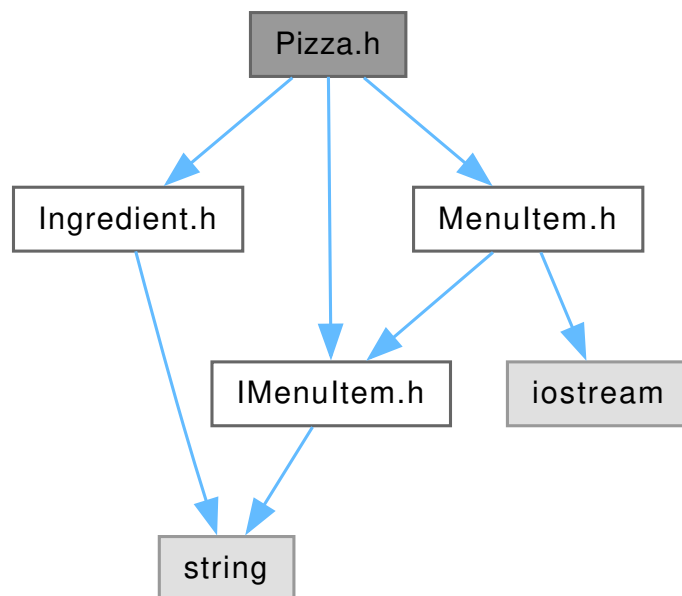


6.24 Pizza.h File Reference

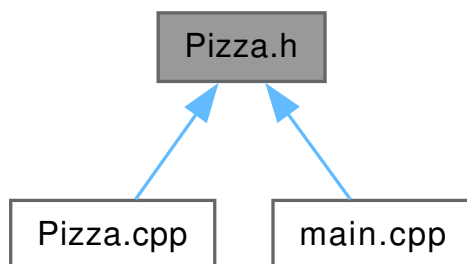
Hlavičkový soubor pro třídu [Pizza](#).

```
#include "IMenuItem.h"  
#include "Ingredient.h"  
#include "MenuItem.h"
```

Include dependency graph for Pizza.h:



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



Data Structures

- class [Pizza](#)
Třída reprezentující pizzu (dědí z [MenuItem](#)).

6.24.1 Detailed Description

Hlavičkový soubor pro třídu [Pizza](#).

6.25 Pizza.h

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

```

00001
00002
00003 #ifndef PROOP_PIZZA_H
00004 #define PROOP_PIZZA_H
00005 #include "IMenuItem.h"
00006 #include "Ingredient.h"

```

```

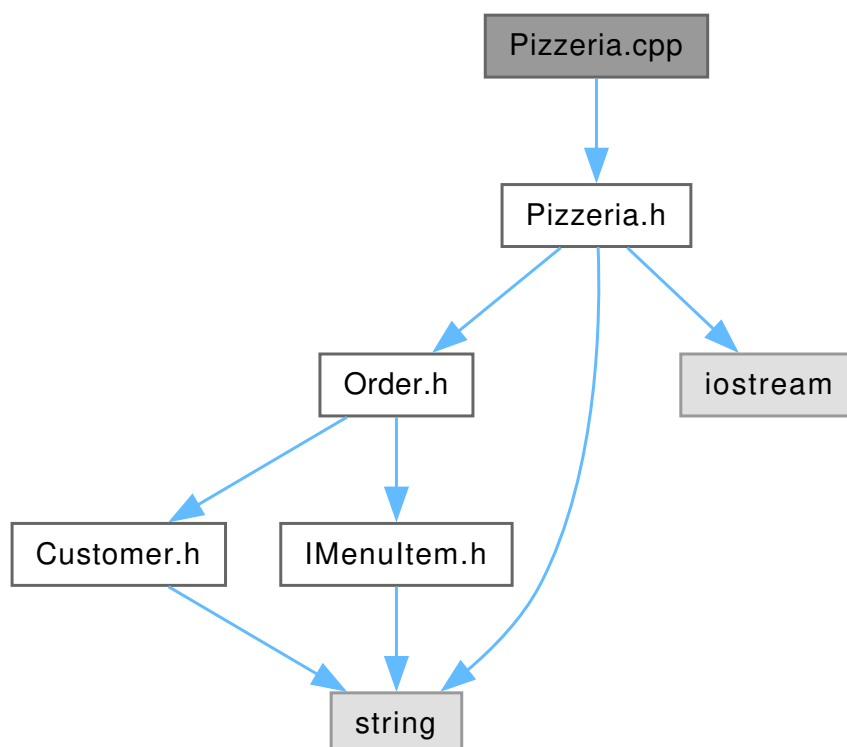
00010 #include "MenuItem.h"
00011
00018 class Pizza : public MenuItem {
00019 private:
00020     int diameter;
00021     bool hasExtraCheese;
00022     Ingredient ** ingredients;
00023     int maxIngredients;
00024     int currentIngredients = 0;
00025
00026 public:
00030     Pizza();
00031
00040     Pizza(string n, double b, int d, bool e, int m);
00041
00045     ~Pizza();
00046
00052     bool addIngredient(Ingredient * i);
00053
00059     bool removeIngredient(string name);
00060
00065     double getPrice() const override;
00066
00071     string printInfo() override;
00072
00077     int getDiameter();
00078 };
00079
00080 #endif //PROOP_PIZZA_H

```

6.26 Pizzeria.cpp File Reference

#include "Pizzeria.h"

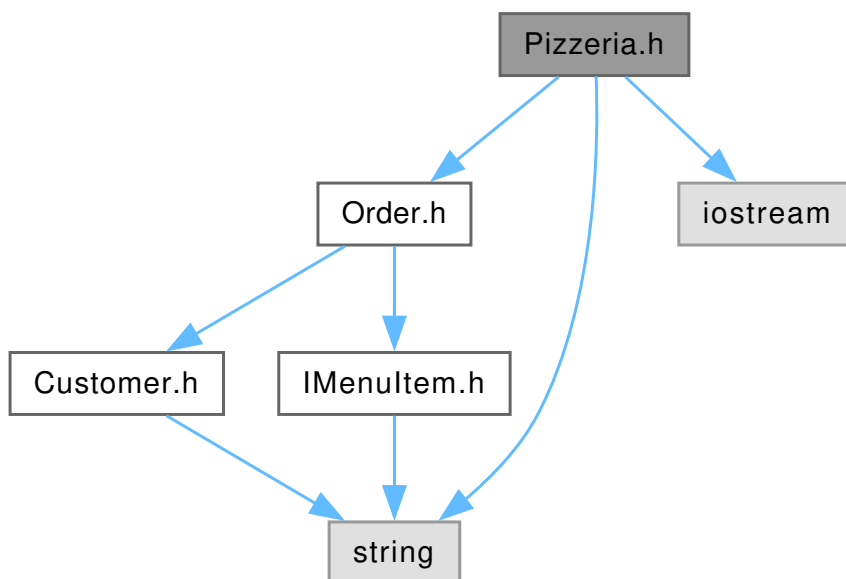
Include dependency graph for Pizzeria.cpp:



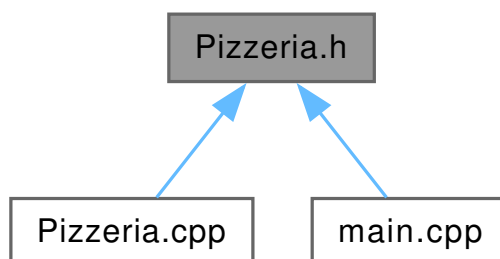
6.27 Pizzeria.h File Reference

Hlavičkový soubor pro třídu [Pizzaeria](#).

```
#include "Order.h"
#include <string>
#include <iostream>
Include dependency graph for Pizzeria.h:
```



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



Data Structures

- class `Pizzeria`
Hlavní třída zastupující samotnou pizzerii.

6.27.1 Detailed Description

Hlavičkový soubor pro třídu `Pizzeria`.

6.28 Pizzeria.h

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

```
00001
00005
00006 #ifndef PROOP_PIZZERIA_H
```

```
00007 #define PROOP_PIZZERIA_H
00008 #include "Order.h"
00009 #include <string>
00010 #include <iostream>
00011 using namespace std;
00012
00019 class Pizzeria {
00020     string name;
00021     Order ** activeOrders;
00022     int orderCount;
00023
00024 public:
00030     Pizzeria(string n, int maxOrders);
00031
00035     ~Pizzeria();
00036
00044     void processOrder(Order * o);
00045 };
00046
00047 #endif //PROOP_PIZZERIA_H
```

6.29 README.md File Reference

Index

- ~AlcoholicDrink
 - AlcoholicDrink, 8
- ~Customer
 - Customer, 10
- ~Drink
 - Drink, 15
- ~MenuItem
 - MenuItem, 18
- ~Ingredient
 - Ingredient, 20
- ~MenuItem
 - MenuItem, 24
- ~Order
 - Order, 28
- ~Pizza
 - Pizza, 34
- ~Pizzeria
 - Pizzeria, 38
- activeOrders
 - Pizzeria, 39
- addIngredient
 - Pizza, 35
- addItem
 - Order, 28
- addPoints
 - Customer, 10
- alcoholContent
 - AlcoholicDrink, 8
- AlcoholicDrink, 4
 - ~AlcoholicDrink, 8
 - alcoholContent, 8
 - AlcoholicDrink, 8
 - printInfo, 8
- AlcoholicDrink.cpp, 40
- AlcoholicDrink.h, 40
- basePrice
 - MenuItem, 25
- calculateTotal
 - Order, 28
- capacity
 - Order, 30
- cost
 - Ingredient, 21
- currentIngredients
 - Pizza, 36
- Customer, 9
 - ~Customer, 10
 - addPoints, 10
 - Customer, 10
 - getDetails, 10
 - loyaltyPoints, 11
 - name, 11
 - phone, 11
- customer
 - Order, 30
- Customer.cpp, 42
- Customer.h, 42
- diameter
 - Pizza, 36
- Drink, 11
 - ~Drink, 15
 - Drink, 15
 - isCarbonated, 15
 - printInfo, 15
 - volume, 15
- Drink.cpp, 44
- Drink.h, 44
- getCost
 - Ingredient, 21
- getCustomer
 - Order, 28
- getDetails
 - Customer, 10
- getDiameter
 - Pizza, 35
- getItem
 - Order, 29
- getItemCount
 - Order, 29
- getName
 - MenuItem, 18
 - Ingredient, 21
 - MenuItem, 24
- getPrice
 - MenuItem, 18
 - MenuItem, 24
 - Pizza, 35
- getTotalOrders
 - Order, 29
- hasExtraCheese
 - Pizza, 36
- MenuItem, 16
 - ~MenuItem, 18
 - getName, 18
 - getPrice, 18
 - printInfo, 19
- MenuItem.cpp, 46
- MenuItem.h, 46
- Ingredient, 19
 - ~Ingredient, 20
 - cost, 21
 - getCost, 21
 - getName, 21
 - Ingredient, 20
 - name, 21

- Ingredient.cpp, 48
- Ingredient.h, 48
- ingredients
 - Pizza, 36
- isCarbonated
 - Drink, 15
- itemCount
 - Order, 30
- items
 - Order, 30
- loyaltyPoints
 - Customer, 11
- main
 - main.cpp, 50
- main.cpp, 50
- main, 50
- maxIngredients
 - Pizza, 36
- MenuItem, 22
 - ~MenuItem, 24
 - basePrice, 25
 - getName, 24
 - getPrice, 24
 - MenuItem, 24
 - name, 25
 - printInfo, 25
- MenuItem.cpp, 50
- MenuItem.h, 51
- name
 - Customer, 11
 - Ingredient, 21
 - MenuItem, 25
 - Pizzeria, 39
- Order, 26
 - ~Order, 28
 - addItem, 28
 - calculateTotal, 28
 - capacity, 30
 - customer, 30
 - getCustomer, 28
 - getItem, 29
 - getItemCount, 29
 - getTotalOrders, 29
 - itemCount, 30
 - items, 30
 - Order, 28
 - removeItem, 29
 - totalOrder, 30
- Order.cpp, 53
- Order.h, 53
- orderCount
 - Pizzeria, 39
- phone
 - Customer, 11
- Pizza, 31
 - ~Pizza, 34
 - addIngredient, 35
 - currentIngredients, 36
 - diameter, 36
 - getDiameter, 35
 - getPrice, 35
 - hasExtraCheese, 36
 - ingredients, 36
 - maxIngredients, 36
 - Pizza, 34
 - printInfo, 35
 - removeIngredient, 35
- Pizza.cpp, 55
- Pizza.h, 55
- Pizzeria, 37
 - ~Pizzeria, 38
 - activeOrders, 39
 - name, 39
 - orderCount, 39
 - Pizzeria, 38
 - processOrder, 39
- Pizzeria.cpp, 57
- Pizzeria.h, 57
- printInfo
 - AlcoholicDrink, 8
 - Drink, 15
 - IMenuItem, 19
 - MenuItem, 25
 - Pizza, 35
- processOrder
 - Pizzeria, 39
- Projekt 1: Objektový model pizzerie, 1
- README.md, 59
- removeIngredient
 - Pizza, 35
- removeItem
 - Order, 29
- totalOrder
 - Order, 30
- volume
 - Drink, 15