

Zarządzanie przepisami

Wygenerowano przez Doxygen 1.9.6

1 Indeks hierarchiczny	1
1.1 Hierarchia klas	1
2 Indeks klas	3
2.1 Lista klas	3
3 Indeks plików	5
3.1 Lista plików	5
4 Dokumentacja klas	7
4.1 Dokumentacja klasy Dodatki	7
4.1.1 Opis szczegółowy	7
4.1.2 Dokumentacja konstruktora i destruktora	8
4.1.2.1 Dodatki()	8
4.1.3 Dokumentacja funkcji składowych	8
4.1.3.1 pokaz()	8
4.2 Dokumentacja klasy Krok	8
4.2.1 Opis szczegółowy	8
4.2.2 Dokumentacja konstruktora i destruktora	9
4.2.2.1 Krok()	9
4.2.3 Dokumentacja funkcji składowych	9
4.2.3.1 wykonanie_kroku()	9
4.3 Dokumentacja klasy Przepis	9
4.3.1 Dokumentacja funkcji składowych	9
4.3.1.1 czytaj_przepis()	9
4.3.1.2 dodaj_krok()	10
4.3.1.3 dodaj_skladnik()	10
4.3.1.4 pokaz()	10
4.4 Dokumentacja klasy Przyprawy	11
4.4.1 Opis szczegółowy	11
4.4.2 Dokumentacja konstruktora i destruktora	11
4.4.2.1 Przyprawy()	11
4.4.3 Dokumentacja funkcji składowych	12
4.4.3.1 pokaz()	12
4.5 Dokumentacja klasy Skladnik	12
4.5.1 Opis szczegółowy	12
4.5.2 Dokumentacja konstruktora i destruktora	13
4.5.2.1 Skladnik()	13
4.5.3 Dokumentacja funkcji składowych	13
4.5.3.1 pokaz()	13
4.6 Dokumentacja klasy Skladniki_glowne	13
4.6.1 Opis szczegółowy	14
4.6.2 Dokumentacja konstruktora i destruktora	14

4.6.2.1 Składniki_glowne()	14
4.6.3 Dokumentacja funkcji składowych	14
4.6.3.1 pokaz()	14
5 Dokumentacja plików	17
5.1 Dokumentacja pliku przepis.h	17
5.1.1 Opis szczegółowy	17
5.2 przepis.h	18
5.3 składnik.h	18
Skorowidz	21

Rozdział 1

Indeks hierarchiczny

1.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

Krok	8
Przepis	9
Składnik	12
Dodatki	7
Przyprawy	11
Składniki_główne	13

Rozdział 2

Indeks klas

2.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

Dodatki	7
Krok	8
Przepis	9
Przyprawy	11
Składnik	12
Składniki_glowne	13

Rozdział 3

Indeks plików

3.1 Lista plików

Tutaj znajduje się lista wszystkich udokumentowanych plików z ich krótkimi opisami:

przepis.h	Plik zawiera definicje klasy przepis	17
skladnik.h	??

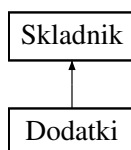
Rozdział 4

Dokumentacja klas

4.1 Dokumentacja klasy Dodatki

```
#include <skladnik.h>
```

Diagram dziedziczenia dla Dodatki



Metody publiczne

- [Dodatki](#) (std::string n)
- void [pokaz](#) () override

Metody publiczne dziedziczone z [Skladnik](#)

- [Skladnik](#) (std::string n)
- virtual void [pokaz](#) ()=0

Dodatkowe Dziedziczone Składowe

Atrybuty chronione dziedziczone z [Skladnik](#)

- std::string **nazwa**

4.1.1 Opis szczegółowy

Klasa pochodna przedstawiająca dodatki

4.1.2 Dokumentacja konstruktora i destruktora

4.1.2.1 Dodatki()

```
Dodatki::Dodatki (
    std::string n ) [inline]
```

Konstruktor klasy [Dodatki](#)

Parametry

<i>n</i>	- nazwa dodatku
----------	-----------------

4.1.3 Dokumentacja funkcji składowych

4.1.3.1 pokaz()

```
void Dodatki::pokaz ( ) [inline], [override], [virtual]
```

Metoda pokazania na ekran dodatków

Implementuje [Skladnik](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- skladnik.h

4.2 Dokumentacja klasy Krok

```
#include <skladnik.h>
```

Metody publiczne

- void [wykonanie_kroku](#) ()
- [Krok](#) (std::string k)

4.2.1 Opis szczegółowy

Klasa przedstawiająca kroki potrzebne do wykonania dania kroki - opis kroku

4.2.2 Dokumentacja konstruktora i destruktora

4.2.2.1 Krok()

```
Krok::Krok (
    std::string k ) [inline]
```

Konstruktor klasy [Krok](#)

Parametry

<i>k</i>	- opis kroku
----------	--------------

4.2.3 Dokumentacja funkcji składowych

4.2.3.1 wykonanie_kroku()

```
void Krok::wykonanie_kroku ( )
```

Metoda wykonania kroku

Dokumentacja dla tej klasy została wygenerowana z plików:

- skladnik.h
- skladnik.cpp

4.3 Dokumentacja klasy Przepis

Metody publiczne

- void [czytaj_przepis](#) (std::ifstream &sciezka_pliku, [Przepis](#) &obiekt)
- void [dodaj_skladnik](#) ([Skladnik](#) *s)
- void [dodaj_krok](#) ([Krok](#) *k)
- void [pokaz](#) ([Przepis](#) &obiekt)

4.3.1 Dokumentacja funkcji składowych

4.3.1.1 czytaj_przepis()

```
void Przepis::czytaj_przepis (
    std::ifstream & sciezka_pliku,
    Przepis & obiekt )
```

Metoda czytania pliku z przepisem

Parametry

<i>sciezka_pliku</i>	- czytany plik
<i>obiekt</i>	- tworzony przepis

4.3.1.2 dodaj_krok()

```
void Przepis::dodaj_krok (
    Krok * k ) [inline]
```

Metoda dodania kroku do wektora składników

Parametry

<i>k</i>	- dodawany krok
----------	-----------------

4.3.1.3 dodaj_skladnik()

```
void Przepis::dodaj_skladnik (
    Skladnik * s ) [inline]
```

Metoda dodania składnika do wektora składników

Parametry

<i>s</i>	- dodawany składnik
----------	---------------------

4.3.1.4 pokaz()

```
void Przepis::pokaz (
    Przepis & obiekt )
```

Metoda wypisania przepisu na ekran

Parametry

<i>obiekt</i>	- tworzony przepis
---------------	--------------------

Dokumentacja dla tej klasy została wygenerowana z plików:

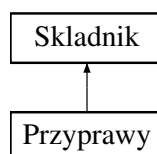
- [przepis.h](#)

- `przepis.cpp`

4.4 Dokumentacja klasy Przyprawy

```
#include <skladnik.h>
```

Diagram dziedziczenia dla Przyprawy



Metody publiczne

- `Przyprawy` (`std::string n`)
- `void pokaz ()` override

Metody publiczne dziedziczone z `Skladnik`

- `Skladnik` (`std::string n`)
- `virtual void pokaz ()=0`

Dodatkowe Dziedziczone Składowe

Atrybuty chronione dziedziczone z `Skladnik`

- `std::string nazwa`

4.4.1 Opis szczegółowy

Klasa pochodna przedstawiająca przyprawy

4.4.2 Dokumentacja konstruktora i destruktora

4.4.2.1 `Przyprawy()`

```
Przyprawy::Przyprawy (
    std::string n ) [inline]
```

Konstruktor klasy `Przyprawy`

Parametry

<i>n</i>	- nazwa przyprawy
----------	-------------------

4.4.3 Dokumentacja funkcji składowych

4.4.3.1 pokaz()

```
void Przyprawy::pokaz ( ) [inline], [override], [virtual]
```

Metoda pokazania na ekran przypraw

Implementuje [Skladnik](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- skladnik.h

4.5 Dokumentacja klasy Skladnik

```
#include <skladnik.h>
```

Diagram dziedziczenia dla Skladnik



Metody publiczne

- [Skladnik](#) (std::string n)
- virtual void [pokaz](#) ()=0

Atrybuty chronione

- std::string **nazwa**

4.5.1 Opis szczegółowy

Wirtualna klasa przedstawiająca składnik

Parametry

<i>nazwa</i>	- nazwa składnika
--------------	-------------------

4.5.2 Dokumentacja konstruktora i destruktora

4.5.2.1 Skladnik()

```
Skladnik::Skladnik (
    std::string n ) [inline]
```

Konstruktor klasy [Skladnik](#)

Parametry

<i>n</i>	- nazwa składnika
----------	-------------------

4.5.3 Dokumentacja funkcji składowych

4.5.3.1 pokaz()

```
virtual void Skladnik::pokaz ( ) [pure virtual]
```

Wirtualna metoda pokazania składników

Implementowany w [Przyprawy](#), [Skladniki_glowne](#) i [Dodatki](#).

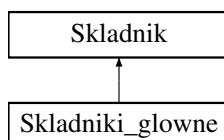
Dokumentacja dla tej klasy została wygenerowana z pliku:

- skladnik.h

4.6 Dokumentacja klasy Skladniki_glowne

```
#include <skladnik.h>
```

Diagram dziedziczenia dla Skladniki_glowne



Metody publiczne

- [Skladniki_glowne](#) (std::string n)
- void [pokaz](#) () override

Metody publiczne dziedziczone z [Skladnik](#)

- [Skladnik](#) (std::string n)
- virtual void [pokaz](#) ()=0

Dodatkowe Dziedziczone Składowe

Atrybuty chronione dziedziczone z [Skladnik](#)

- std::string [nazwa](#)

4.6.1 Opis szczegółowy

Klasa pochodna przedstawiająca składniki główne

4.6.2 Dokumentacja konstruktora i destruktora

4.6.2.1 Skladniki_glowne()

```
Skladniki_glowne::Skladniki_glowne (  
    std::string n ) [inline]
```

Konstruktor klasy [Skladniki_glowne](#)

Parametry

<i>n</i>	- nazwa składnika głównego
----------	----------------------------

4.6.3 Dokumentacja funkcji składowych

4.6.3.1 pokaz()

```
void Skladniki_glowne::pokaz ( ) [inline], [override], [virtual]
```

Metoda pokazania na ekran składników głównych

Implementuje [Skladnik](#).

Dokumentacja dla tej klasy została wygenerowana z pliku:

- skladnik.h

Rozdział 5

Dokumentacja plików

5.1 Dokumentacja pliku przepis.h

plik zawiera definicje klasy przepis

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

Komponenty

- class [Przepis](#)

5.1.1 Opis szczegółowy

plik zawiera definicje klasy przepis

Autor

Victoria Morawiec

Data

July 2023

5.2 przepis.h

[Idź do dokumentacji tego pliku.](#)

```
00001 /*****
00009 #pragma once
00010
00011 #include <iostream>
00012 #include <fstream>
00013 #include <string>
00014 #include <vector>
00015 #include "skladnik.h"
00016
00017
00024 class Skladnik;
00025 class Krok;
00026
00027
00028 class Przepis {
00029
00030     std::string nazwa;
00031     std::vector<Skladnik*> skladniki;    //wektor składników
00032     std::vector<Krok*> kroki;           //wektor przechowujący kroki
00033
00034 public:
00035
00040     void czytaj_przepis(std::ifstream &sciezka_pliku, Przepis& obiekt); //Pierwsza linia to nazwa
        przepisu, druga to "Składniki główne", czyta aż do natrafienia "Przyprawy", a potem robi to samo, aż
        natrafi na "dodatki, potem czyta składniki aż do znalezienia słowa "Kroki:", kończy na znalezieniu "*"
00041
00045     void dodaj_skladnik(Skladnik* s) {
00046         skladniki.push_back(s);
00047     }
00048
00052     void dodaj_krok(Krok* k) {
00053         kroki.push_back(k);
00054     }
00055
00059     void pokaz(Przepis& obiekt);    //pokazuje całość przepisu
00060 };
00061
00062
00063
```

5.3 skladnik.h

```
00001 #pragma once
00002
00003 #include <iostream>
00004 #include <fstream>
00005 #include <string>
00006 #include <vector>
00007 #include "przepis.h"
00008
00012 class Skladnik {
00013
00014 protected:
00015     std::string nazwa;
00016 public:
00017
00021     Skladnik(std::string n) {
00022         nazwa = n;
00023     }
00024
00027     virtual void pokaz() = 0;
00028 };
00029
00032 class Przyprawy : public Skladnik {
00033
00034 public:
00035
00039     Przyprawy(std::string n) : Skladnik(n) {
00040     }
00041
00044     void pokaz() override {
00045         std::cout << "- przyprawa: " << nazwa << std::endl;
00046     }
00047 };
00048
00052 class Skladniki_glowne : public Skladnik {
00053 public:
00054
```

```
00058     Skladniki_glowne(std::string n) : Skladnik(n) {
00059     }
00060
00063     void pokaz() override {
00064         std::cout << "- składnik główny: " << nazwa << std::endl;
00065     }
00066 };
00070 class Dodatki : public Skladnik {
00071 public:
00072
00076     Dodatki(std::string n) : Skladnik(n) {
00077     }
00078
00081     void pokaz() override {
00082         std::cout << "- dodatek: " << nazwa << std::endl;
00083     }
00084 };
00085
00089 class Krok {
00090     std::string kroki;
00091
00092 public:
00093
00096     void wykonanie_kroku();
00097
00102     Krok(std::string k) : kroki(k) {
00103     }
00104 };
```


Skorowidz

czytaj_przepis
Przepis, [9](#)

dodaj_krok
Przepis, [10](#)

dodaj_skladnik
Przepis, [10](#)

Dodatki, [7](#)
Dodatki, [8](#)
pokaz, [8](#)

Krok, [8](#)
Krok, [9](#)
wykonanie_kroku, [9](#)

pokaz
Dodatki, [8](#)
Przepis, [10](#)
Przyprawy, [12](#)
Skladnik, [13](#)
Skladniki_glowne, [14](#)

Przepis, [9](#)
czytaj_przepis, [9](#)
dodaj_krok, [10](#)
dodaj_skladnik, [10](#)
pokaz, [10](#)

przepis.h, [17](#)
Przyprawy, [11](#)
pokaz, [12](#)
Przyprawy, [11](#)

Skladnik, [12](#)
pokaz, [13](#)
Skladnik, [13](#)
Skladniki_glowne, [13](#)
pokaz, [14](#)
Skladniki_glowne, [14](#)

wykonanie_kroku
Krok, [9](#)