

ZarządzanieFakturami

Wygenerowano przez Doxygen 1.9.1

1 Indeks hierarchiczny	1
1.1 Hierarchia klas	1
2 Indeks klas	3
2.1 Lista klas	3
3 Dokumentacja klas	5
3.1 Dokumentacja klasy Artykul	5
3.1.1 Opis szczegółowy	5
3.1.2 Dokumentacja konstruktora i destruktora	6
3.1.2.1 Artykul()	6
3.1.3 Dokumentacja funkcji składowych	6
3.1.3.1 cenaBrutto()	6
3.1.3.2 cenaNetto()	6
3.1.3.3 kwotaVAT()	7
3.1.3.4 stawkaVAT()	7
3.1.3.5 vatString()	7
3.2 Dokumentacja klasy Faktura	7
3.2.1 Opis szczegółowy	8
3.2.2 Dokumentacja konstruktora i destruktora	8
3.2.2.1 Faktura()	9
3.2.3 Dokumentacja funkcji składowych	9
3.2.3.1 dodajTowar()	9
3.2.3.2 getCzyOplacone()	9
3.2.3.3 getDataSprzedazy()	9
3.2.3.4 getFormaPlatnosci()	9
3.2.3.5 getKlient()	10
3.2.3.6 getKwotaBrutto()	10
3.2.3.7 getKwotaNetto()	10
3.2.3.8 getListTowarow()	10
3.2.3.9 getMiejsceWystawienia()	10
3.2.3.10 getNumerFaktury()	10
3.2.3.11 getSprzedawca()	10
3.2.3.12 getUwagi()	10
3.2.3.13 setCzyOplacone()	10
3.2.3.14 setDataSprzedazy()	11
3.2.3.15 setFormaPlatnosci()	11
3.2.3.16 setKlient()	11
3.2.3.17 setMiejsceWystawienia()	12
3.2.3.18 setNumerFaktury()	12
3.2.3.19 setSprzedawca()	12
3.2.3.20 setUwagi()	12
3.2.3.21 suma()	13

3.2.3.22 wypisz()	13
3.2.3.23 wypiszNumerFaktury()	13
3.3 Dokumentacja klasy Klient	13
3.3.1 Opis szczegółowy	14
3.4 Dokumentacja klasy KlientFirmowy	14
3.4.1 Opis szczegółowy	15
3.4.2 Dokumentacja konstruktora i destruktora	16
3.4.2.1 KlientFirmowy() [1/2]	16
3.4.2.2 KlientFirmowy() [2/2]	16
3.4.3 Dokumentacja funkcji składowych	16
3.4.3.1 nazwa()	16
3.4.3.2 wypisz()	17
3.5 Dokumentacja klasy KlientIndywidualny	17
3.5.1 Opis szczegółowy	17
3.5.2 Dokumentacja konstruktora i destruktora	18
3.5.2.1 KlientIndywidualny()	18
3.5.3 Dokumentacja funkcji składowych	18
3.5.3.1 nazwa()	18
3.5.3.2 wypisz()	18
3.6 Dokumentacja klasy KreatorFaktur	19
3.6.1 Opis szczegółowy	19
3.6.2 Dokumentacja konstruktora i destruktora	19
3.6.2.1 KreatorFaktur()	19
3.6.3 Dokumentacja funkcji składowych	20
3.6.3.1 dodajFakture()	20
3.6.3.2 usunFakture()	20
3.6.3.3 wyczyscPamiec()	20
3.6.3.4 wypisywanieKlientow()	20
3.6.3.5 wypisywanieTowarow()	20
3.6.3.6 wypiszFakture()	20
3.6.3.7 wypiszListeFaktur()	21
3.7 Dokumentacja klasy Towar	21
3.7.1 Opis szczegółowy	21
3.7.2 Dokumentacja funkcji składowych	22
3.7.2.1 getIlosc()	22
3.7.2.2 nazwa()	22
3.7.2.3 setIlosc()	22
3.7.2.4 wypisz()	22
3.8 Dokumentacja klasy Usługa	22
3.8.1 Opis szczegółowy	23
3.8.2 Dokumentacja konstruktora i destruktora	23
3.8.2.1 Usługa()	23

3.8.3 Dokumentacja funkcji składowych	23
3.8.3.1 cenaBrutto()	24
3.8.3.2 cenaNetto()	24
3.8.3.3 kwotaVAT()	24
3.8.3.4 stawkaVAT()	24
3.8.3.5 vatString()	24
3.9 Dokumentacja klasy WczytywaczDanych	25
3.9.1 Opis szczegółowy	25
3.9.2 Dokumentacja funkcji składowych	25
3.9.2.1 wczytajKlientow()	25
3.9.2.2 wczytajSprzedawce()	25
3.9.2.3 wczytajTowary()	26
3.10 Dokumentacja klasy Zestaw	26
3.10.1 Opis szczegółowy	26
3.10.2 Dokumentacja konstruktora i destruktora	27
3.10.2.1 Zestaw() [1/2]	27
3.10.2.2 Zestaw() [2/2]	27
3.10.2.3 ~Zestaw()	27
3.10.3 Dokumentacja funkcji składowych	27
3.10.3.1 cenaBrutto()	27
3.10.3.2 cenaNetto()	28
3.10.3.3 dodajTowar()	28
3.10.3.4 kwotaVAT()	28
3.10.3.5 stawkaVAT()	28
3.10.3.6 vatString()	28
Indeks	29

Rozdział 1

Indeks hierarchiczny

1.1 Hierarchia klas

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

Faktura	7
Klient	13
KlientFirmowy	14
KlientIndywidualny	17
KreatorFaktur	19
Towar	21
Artykul	5
Usługa	22
Zestaw	26
WczytywaczDanych	25

Rozdział 2

Indeks klas

2.1 Lista klas

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

Artykul	5
Faktura	7
Klient	13
KlientFirmowy	14
KlientIndywidualny	17
KreatorFaktur	19
Towar	21
Usługa	22
WczytywaczDanych	25
Zestaw	26

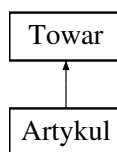
Rozdział 3

Dokumentacja klas

3.1 Dokumentacja klasy Artykul

```
#include <Artykul.h>
```

Diagram dziedziczenia dla Artykul



Metody publiczne

- [Artykul](#) (string nazwaTowaru, string jednostka, float cena, int vat)
- float [cenaNetto](#) ()
- float [cenaBrutto](#) ()
- int [stawkaVAT](#) ()
- float [kwotaVAT](#) ()
- string [vatString](#) ()

Dodatkowe Dziedziczone Składowe

3.1.1 Opis szczegółowy

Klasa jednego typu towaru - [Artykul](#). Podklasa, która dziedziczy po klasie [Towar](#). Zawiera informacje o cenie towaru i stawce VAT.

Parametry

<i>cena</i>	cena towaru;
<i>vat</i>	stawka VAT;

3.1.2 Dokumentacja konstruktora i destruktora

3.1.2.1 Artykul()

```
Artykul::Artykul (
    string nazwaTowaru,
    string jednostka,
    float cena,
    int vat )
```

Konstruktor przyjmujący dane do utworzenia faktury.

Parametry

<i>nazwaTowaru</i>	nazwa towaru;
<i>jednostka</i>	jednostka;
<i>cena</i>	cena towaru;
<i>vat</i>	stawka VAT;

3.1.3 Dokumentacja funkcji składowych

3.1.3.1 cenaBrutto()

```
float Artykul::cenaBrutto ( ) [virtual]
```

Metoda wyliczająca i zwracająca cenę brutto.

Implementuje [Towar](#).

3.1.3.2 cenaNetto()

```
float Artykul::cenaNetto ( ) [virtual]
```

Metoda wyliczająca i zwracająca cenę netto.

Implementuje [Towar](#).

3.1.3.3 kwotaVAT()

```
float Artykul::kwotaVAT ( ) [virtual]
```

Metoda wyliczająca i zwracająca kwotę VAT.

Implementuje [Towar](#).

3.1.3.4 stawkaVAT()

```
int Artykul::stawkaVAT ( ) [virtual]
```

Metoda zwracająca stawkę VAT.

Implementuje [Towar](#).

3.1.3.5 vatString()

```
string Artykul::vatString ( ) [virtual]
```

Metoda zwracająca stawkę VAT w stringu.

Implementuje [Towar](#).

Dokumentacja dla tej klasy została wygenerowana z plików:

- Artykul.h
- Artykul.cpp

3.2 Dokumentacja klasy Faktura

```
#include <Faktura.h>
```

Metody publiczne

- [Faktura](#) ()
- `template<typename T >`
`T suma (const T &lhs, const T &rhs)`
- [KlientFirmowy](#) [getSprzedawca](#) ()
- `void setSprzedawca (KlientFirmowy sprzedawca)`
- `string getNumerFaktury` ()
- `void setNumerFaktury (string numerFaktury)`
- `string getMiejsceWystawienia` ()
- `void setMiejsceWystawienia (string miejsceWystawienia)`
- `string getDataSprzedazy` ()
- `void setDataSprzedazy (string dataSprzedazy)`
- [Klient](#) * [getKlient](#) ()
- `void setKlient (Klient *klient)`
- `vector< Towar * > getListiTowarow` ()
- `void dodajTowar (int ilosc, Towar *towar)`
- `float getKwotaNetto` ()
- `float getKwotaBrutto` ()
- `string getFormaPlatnosci` ()
- `void setFormaPlatnosci (string formaPlatnosci)`
- `bool getCzyOplacone` ()
- `void setCzyOplacone (bool czyOplacone)`
- `string getUwagi` ()
- `void setUwagi (string uwagi)`
- `void wypisz` ()
- `void wypiszNumerFaktury` ()

3.2.1 Opis szczegółowy

Klasa zawiera w sobie informacje o elementach faktury;

Parametry

<i>sprzedawca</i>	zawiera informacje na temat sprzedawcy;
<i>numerFaktury</i>	zawiera informacje na temat numeru faktury;
<i>miejsceWystawienia</i>	zawiera informacje na temat miejsca wystawienia faktury;
<i>dataSprzedazy</i>	zawiera informacje na temat daty sprzedazy towaru;
<i>klient</i>	zawiera informacje na temat klienta;
<i>listaTowarow</i>	pozycje towarow na fakturze;
<i>formaPlatnosci</i>	zawiera informacje o wybranym sposobie platnosci klienta;
<i>czyOplacone</i>	zawiera informacje o tym czy faktura zostala oplacona;
<i>uwagi</i>	przechowuje uwagi dla urzedu skarbowego;

3.2.2 Dokumentacja konstruktora i destruktoru

3.2.2.1 Faktura()

```
Faktura::Faktura ( )
```

Konstruktor ustawiający wartości domyślne pola czyOpłacone oraz klient.

3.2.3 Dokumentacja funkcji składowych

3.2.3.1 dodajTowar()

```
void Faktura::dodajTowar (
    int ilosc,
    Towar * towar )
```

Metoda dodająca towar do listy towarów w zadanej ilości.

Parametry

<i>ilosc</i>	ilosc;
<i>towar</i>	dodawany towar;

3.2.3.2 getCzyOpłacone()

```
bool Faktura::getCzyOpłacone ( )
```

Metoda zwracająca wartość, która zawiera informacje czy faktura została opłacona.

3.2.3.3 getDataSprzedazy()

```
string Faktura::getDataSprzedazy ( )
```

Metoda zwracająca datę sprzedaży towaru.

3.2.3.4 getFormaPłatności()

```
string Faktura::getFormaPłatności ( )
```

Metoda zwraca formę płatności.

3.2.3.5 getKlient()

```
Klient * Faktura::getKlient ( )
```

Metoda zwracająca klienta.

3.2.3.6 getKwotaBrutto()

```
float Faktura::getKwotaBrutto ( )
```

Metoda wyliczająca i zwracająca kwotę brutto faktury.

3.2.3.7 getKwotaNetto()

```
float Faktura::getKwotaNetto ( )
```

Metoda wyliczająca i zwracająca kwotę netto faktury.

3.2.3.8 getListTowarow()

```
vector< Towar * > Faktura::getListTowarow ( )
```

Metoda zwracająca listę towarów.

3.2.3.9 getMiejsceWystawienia()

```
string Faktura::getMiejsceWystawienia ( )
```

Metoda zwracająca miejsce wystawienia faktury;

3.2.3.10 getNumerFaktury()

```
string Faktura::getNumerFaktury ( )
```

Metoda zwracająca numer faktury.

3.2.3.11 getSprzedawca()

```
KlientFirmowy Faktura::getSprzedawca ( )
```

Metoda zwracająca dane sprzedawcy.

3.2.3.12 getUwagi()

```
string Faktura::getUwagi ( )
```

Metoda zwracająca uwagi.

3.2.3.13 setCzyOpłacone()

```
void Faktura::setCzyOpłacone (
    bool czyOpłacone )
```

Metoda przypisująca nową wartość do pola czyOpłacone.

Parametry

<i>czyOpłacane;</i>

3.2.3.14 setDataSprzedazy()

```
void Faktura::setDataSprzedazy (
    string dataSprzedazy )
```

Metoda przypisująca nowa wartość do pola dataSprzedazy.

Parametry

<i>dataSprzedazy</i>	data sprzedaży towaru;
----------------------	------------------------

3.2.3.15 setFormaPlatnosci()

```
void Faktura::setFormaPlatnosci (
    string formaPlatnosci )
```

Metoda przypisująca nowa wartość do pola formaPlatnosci.

Parametry

<i>formaPlatnosci</i>	forma płatności;
-----------------------	------------------

3.2.3.16 setKlient()

```
void Faktura::setKlient (
    Klient * klient )
```

Metoda przypisująca nowa wartość do pola klient;

Parametry

<i>klient</i>	klient;
---------------	---------

3.2.3.17 setMiejsceWystawienia()

```
void Faktura::setMiejsceWystawienia (
    string miejsceWystawienia )
```

Metoda przypisująca nowa wartosc do pola `miejsceWystawienia`.

Parametry

<i>miejsceWystawienia</i>	miejsce wystawienia faktury;
---------------------------	------------------------------

3.2.3.18 setNumerFaktury()

```
void Faktura::setNumerFaktury (
    string numerFaktury )
```

Metoda przypisująca nowa wartosc do pola `numerFaktury`;

Parametry

<i>numerFaktury</i>	numer faktury;
---------------------	----------------

3.2.3.19 setSprzedawca()

```
void Faktura::setSprzedawca (
    KlientFirmowy sprzedawca )
```

Metoda przypisująca nowa wartosc do pola `sprzedawca`.

Parametry

<i>sprzedawca</i>	obiekt reprezentujący sprzedawcę;
-------------------	-----------------------------------

3.2.3.20 setUwagi()

```
void Faktura::setUwagi (
    string uwagi )
```

Metoda przypisująca nowa wartosc do pola `uwagi`.

Parametry

<i>uwagi</i>	uwagi dla urzedu skarbowego;
--------------	------------------------------

3.2.3.21 suma()

```
template<typename T >
T Faktura::suma (
    const T & a,
    const T & b )
```

Szablon. Metoda zwracajaca sume podanych argumentow.

Parametry

<i>a</i>	skladnik;
<i>b</i>	skladnik;

3.2.3.22 wypisz()

```
void Faktura::wypisz ( )
```

Metoda wypisujaca szczegoly faktury.

3.2.3.23 wypiszNumerFaktury()

```
void Faktura::wypiszNumerFaktury ( )
```

Metoda wypisujaca numer faktury.

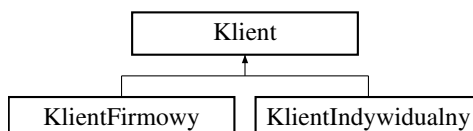
Dokumentacja dla tej klasy zostala wygenerowana z plikow:

- Faktura.h
- Faktura.cpp

3.3 Dokumentacja klasy Klient

```
#include <Klient.h>
```

Diagram dziedziczenia dla Klient



Metody publiczne

- virtual string **nazwa** ()=0
- virtual void **wypisz** ()=0

Atrybuty chronione

- string **adres**
- string **kodPocztowy**
- string **miescowosc**

3.3.1 Opis szczegółowy

Klasa zawiera informacje dotyczące klienta oraz wirtualna metode nazwa i wypisz;

Parametry

<i>adres</i>	adres klienta;
<i>kodPocztowy</i>	kod pocztowy klienta;
<i>miescowosc</i>	miescowosc klienta;

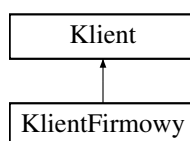
Dokumentacja dla tej klasy została wygenerowana z pliku:

- Klient.h

3.4 Dokumentacja klasy KlientFirmowy

```
#include <KlientFirmowy.h>
```

Diagram dziedziczenia dla KlientFirmowy



Metody publiczne

- [KlientFirmowy](#) ()
- [KlientFirmowy](#) (string nazwaFirmy, string nip, string adres, string kodPocztowy, string miescowosc, string zwolnionyZVat)
- string [nazwa](#) ()
- void [wypisz](#) ()

Dodatkowe Dziedziczone Składowe

3.4.1 Opis szczegółowy

Klasa jednego typu klienta - [Klient](#) Firmowy. Podklasa, która dziedziczy po klasie [Klient](#). Zawiera nazwę firmy, NIP oraz informacje czy klient jest zwolniony z podatku VAT.

Parametry

<i>nazwaFirmy</i>	nazwa firmy;
<i>NIP</i>	NIP klienta;
<i>zwolnionyZVat</i>	zawiera informacje czy klient jest zwolniony z podatku VAT;

3.4.2 Dokumentacja konstruktora i destruktor

3.4.2.1 KlientFirmowy() [1/2]

```
KlientFirmowy::KlientFirmowy ( )
```

Konstruktor ustawiający wartości domyślne pola *zwolnionyZVat*.

3.4.2.2 KlientFirmowy() [2/2]

```
KlientFirmowy::KlientFirmowy (
    string nazwaFirmy,
    string nip,
    string adres,
    string kodPocztowy,
    string miejscowosc,
    string zwolnionyZVat )
```

Konstruktor przyjmujący dane klienta firmowego.

Parametry

<i>nazwaFirmy</i>	nazwa firmy;
<i>nip</i>	NIP klienta firmowego;
<i>adres</i>	adres klienta firmowego;
<i>kodPocztowy</i>	kod pocztowy klienta firmowego;
<i>miejscowosc</i>	miejscowosc klienta firmowego;
<i>zwolnionyZVat</i>	zawiera informacje czy klient firmowy jest zwolniony z podatku VAT;

3.4.3 Dokumentacja funkcji składowych

3.4.3.1 nazwa()

```
string KlientFirmowy::nazwa ( ) [virtual]
```

Metoda zwracająca nazwę firmy.

Implementuje [Klient](#).

3.4.3.2 wypisz()

```
void KlientFirmowy::wypisz ( ) [virtual]
```

Metoda wypisująca dane klienta firmowego.

Implementuje [Klient](#).

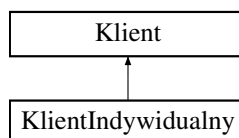
Dokumentacja dla tej klasy została wygenerowana z plików:

- KlientFirmowy.h
- KlientFirmowy.cpp

3.5 Dokumentacja klasy KlientIndywidualny

```
#include <KlientIndywidualny.h>
```

Diagram dziedziczenia dla KlientIndywidualny



Metody publiczne

- [KlientIndywidualny](#) (string imie, string nazwisko, string adres, string kodPocztowy, string miejscowosc)
- string [nazwa](#) ()
- void [wypisz](#) ()

Dodatkowe Dziedziczone Składowe

3.5.1 Opis szczegółowy

Klasa jednego typu klienta - [Klient](#) Indywidualny. Podklasa, która dziedziczy po klasie [Klient](#). Zawiera informacje na temat imienia oraz nazwiska klienta.

Parametry

<i>imie</i>	imie klienta;
<i>nazwisko</i>	nazwisko klienta;

3.5.2 Dokumentacja konstruktora i destruktora

3.5.2.1 KlientIndywidualny()

```
KlientIndywidualny::KlientIndywidualny (
    string imie,
    string nazwisko,
    string adres,
    string kodPocztowy,
    string miejscowosc )
```

Konstruktor przyjmujący dane klienta indywidualnego.

Parametry

<i>imie</i>	imie klienta indywidualnego;
<i>nazwisko</i>	nazwisko klienta indywidualnego;
<i>adres</i>	adres klienta indywidualnego;
<i>kodPocztowy</i>	kod pocztowy klienta indywidualnego;
<i>miejscowosc</i>	miejscowosc klienta indywidualnego;

3.5.3 Dokumentacja funkcji składowych

3.5.3.1 nazwa()

```
string KlientIndywidualny::nazwa ( ) [virtual]
```

Metoda zwracająca imię i nazwisko klienta indywidualnego.

Implementuje [Klient](#).

3.5.3.2 wypisz()

```
void KlientIndywidualny::wypisz ( ) [virtual]
```

Metoda wypisująca nazwę oraz adres klienta indywidualnego.

Implementuje [Klient](#).

Dokumentacja dla tej klasy została wygenerowana z plików:

- KlientIndywidualny.h
- KlientIndywidualny.cpp

3.6 Dokumentacja klasy KreatorFaktur

```
#include <KreatorFaktur.h>
```

Metody publiczne

- `KreatorFaktur` (`KlientFirmowy` sprzedawca, `vector< Klient * >` k, `vector< Towar * >` t)
- `void wypisywanieKlientow` ()
- `void wypisywanieTowarow` ()
- `void dodajFakture` ()
- `void wypiszListeFaktur` ()
- `void wypiszFakture` (int i)
- `void usunFakture` (int i)
- `void wyczyscPamiec` ()

3.6.1 Opis szczegółowy

Klasa zajmuje siorzeniem i przechowywaniem utworzonych faktur, a także wypisywaniem ich i usuwaniem.

Parametry

<i>sprzedawca</i>	dane sprzedającego;
<i>klienci</i>	baza klientow;
<i>towary</i>	baza towarow;
<i>faktury</i>	utworzone faktury;

3.6.2 Dokumentacja konstruktora i destruktora

3.6.2.1 KreatorFaktur()

```
KreatorFaktur::KreatorFaktur (  
    KlientFirmowy sprzedawca,  
    vector< Klient * > k,  
    vector< Towar * > t )
```

Konstruktor.

Parametry

<i>sprzedawca</i>	sprzedawca;
<i>k</i>	klienci;
<i>t</i>	towary;

3.6.3 Dokumentacja funkcji składowych

3.6.3.1 dodajFaktur()

```
void KreatorFaktur::dodajFaktur ( )
```

Metoda tworząca i dodająca fakturę.

3.6.3.2 usunFaktur()

```
void KreatorFaktur::usunFaktur (
    int i )
```

Metoda usuwająca fakturę.

3.6.3.3 wyczyszczeniePamiec()

```
void KreatorFaktur::wyczyszczeniePamiec ( )
```

Metoda czyszcząca pamięć pol.

3.6.3.4 wypisywanieKlientow()

```
void KreatorFaktur::wypisywanieKlientow ( )
```

Metoda wypisująca listę klientów.

3.6.3.5 wypisywanieTowarow()

```
void KreatorFaktur::wypisywanieTowarow ( )
```

Metoda wypisująca listę towarów.

3.6.3.6 wypiszFaktur()

```
void KreatorFaktur::wypiszFaktur (
    int i )
```

Metoda wypisująca fakturę.

3.6.3.7 wypiszListeFaktur()

```
void KreatorFaktur::wypiszListeFaktur ( )
```

Metoda wypisująca liste faktur.

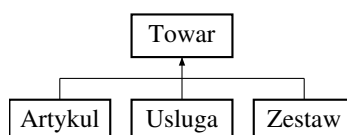
Dokumentacja dla tej klasy została wygenerowana z plików:

- KreatorFaktur.h
- KreatorFaktur.cpp

3.7 Dokumentacja klasy Towar

```
#include <Towar.h>
```

Diagram dziedziczenia dla Towar



Metody publiczne

- virtual float **cenaNetto** ()=0
- virtual float **cenaBrutto** ()=0
- virtual int **stawkaVAT** ()=0
- virtual float **kwotaVAT** ()=0
- virtual string **vatString** ()=0
- void **wypisz** ()
- string **nazwa** ()
- void **setIlosc** (int ilosc)
- int **getIlosc** ()

Atrybuty chronione

- string **nazwaTowaru**
- string **jednostka**
- int **ilosc** = 1

3.7.1 Opis szczegółowy

Klasa zawierająca informacje na temat nazwy towaru, jednostki, ilości oraz wirtualne metody.

Parametry

<i>nazwaTowaru</i>	nazwa towaru;
<i>jednostka</i>	jednostka;
<i>ilosc</i>	ilosc towaru z domyslna wartoscia 1;

3.7.2 Dokumentacja funkcji składowych

3.7.2.1 `getIlosc()`

```
int Towar::getIlosc ( )
```

Metoda zwracająca ilość towaru.

3.7.2.2 `nazwa()`

```
string Towar::nazwa ( )
```

Metoda zwracająca nazwę towaru.

3.7.2.3 `setIlosc()`

```
void Towar::setIlosc (
    int ilosc )
```

Metoda przypisująca nową wartość do pola `ilosc`.

3.7.2.4 `wypisz()`

```
void Towar::wypisz ( )
```

Metoda wypisująca nazwę towaru, ilość, jednostkę, cenę netto, cenę brutto, stawkę VAT oraz wartość VAT.

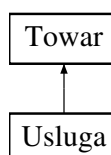
Dokumentacja dla tej klasy została wygenerowana z plików:

- `Towar.h`
- `Towar.cpp`

3.8 Dokumentacja klasy `Usługa`

```
#include <Usługa.h>
```

Diagram dziedziczenia dla `Usługa`



Metody publiczne

- `Usługa` (string nazwaTowaru, float cena, int vat)
- float `cenaNetto` ()
- float `cenaBrutto` ()
- int `stawkaVAT` ()
- float `kwotaVAT` ()
- string `vatString` ()

Dodatkowe Dziedziczone Składowe

3.8.1 Opis szczegółowy

Klasa jednego typu towaru - `Usługa`. Podklasa, która dziedziczy po klasie `Towar`. Zawiera informacje o cenie towaru i stawce VAT.

Parametry

<i>cena</i>	cena towaru;
<i>vat</i>	stawka VAT;

3.8.2 Dokumentacja konstruktora i destruktora

3.8.2.1 Usługa()

```
Usługa::Usługa (
    string nazwaTowaru,
    float cena,
    int vat )
```

Konstruktor przyjmujący dane do utworzenia faktury.

Parametry

<i>nazwaTowaru</i>	nazwa towaru;
<i>cena</i>	cena towaru;
<i>vat</i>	stawka VAT;

3.8.3 Dokumentacja funkcji składowych

3.8.3.1 cenaBrutto()

```
float Usluga::cenaBrutto ( ) [virtual]
```

Metoda wyliczająca i zwracająca cenę brutto.

Implementuje [Towar](#).

3.8.3.2 cenaNetto()

```
float Usluga::cenaNetto ( ) [virtual]
```

Metoda wyliczająca i zwracająca cenę netto.

Implementuje [Towar](#).

3.8.3.3 kwotaVAT()

```
float Usluga::kwotaVAT ( ) [virtual]
```

Metoda wyliczająca i zwracająca kwotę VAT.

Implementuje [Towar](#).

3.8.3.4 stawkaVAT()

```
int Usluga::stawkaVAT ( ) [virtual]
```

Metoda zwracająca stawkę VAT.

Implementuje [Towar](#).

3.8.3.5 vatString()

```
string Usluga::vatString ( ) [virtual]
```

Metoda zwracająca stawkę VAT w stringu.

Implementuje [Towar](#).

Dokumentacja dla tej klasy została wygenerowana z plików:

- Usluga.h
- Usluga.cpp

3.9 Dokumentacja klasy WczytywaczDanych

```
#include <WczytywaczDanych.h>
```

Metody publiczne

- `vector< Klient * > wczytajKlientow` (string sciezka)
- `vector< Towar * > wczytajTowary` (string sciezka)
- `KlientFirmowy wczytajSprzedawce` (string sciezka)

3.9.1 Opis szczegółowy

Klasa zajmuje sisytywaniem klientow, towarow i danych sprzedawcy.

3.9.2 Dokumentacja funkcji składowych

3.9.2.1 wczytajKlientow()

```
vector< Klient * > WczytywaczDanych::wczytajKlientow (
    string sciezka )
```

Metoda wczytująca klientow z pliku.

Parametry

<i>sciezka</i>	sciezka do pliku;
----------------	-------------------

3.9.2.2 wczytajSprzedawce()

```
KlientFirmowy WczytywaczDanych::wczytajSprzedawce (
    string sciezka )
```

Metoda wczytująca dane sprzedawcy z pliku.

Parametry

<i>sciezka</i>	sciezka do pliku;
----------------	-------------------

3.9.2.3 wczytajTowary()

```
vector< Towar * > WczytywaczDanych::wczytajTowary (
    string sciezka )
```

Metoda wczytująca towary z pliku.

Parametry

<i>sciezka</i>	sciezka do pliku;
----------------	-------------------

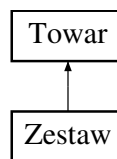
Dokumentacja dla tej klasy została wygenerowana z plików:

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

3.10 Dokumentacja klasy Zestaw

```
#include <Zestaw.h>
```

Diagram dziedziczenia dla Zestaw



Metody publiczne

- [Zestaw](#) ()
- [Zestaw](#) (string nazwaTowaru)
- [~Zestaw](#) ()
- void [dodajTowar](#) ([Artykul](#) *artykul)
- float [cenaNetto](#) ()
- float [cenaBrutto](#) ()
- float [kwotaVAT](#) ()
- int [stawkaVAT](#) ()
- string [vatString](#) ()

Dodatkowe Dziedziczone Składowe

3.10.1 Opis szczegółowy

Klasa jednego typu towaru - [Zestaw](#). Podklasa, która dziedziczy po klasie [Towar](#). Zawiera w sobie wektor towarów.

Parametry

<i>listaTowarow</i>	baza towarow;
---------------------	---------------

3.10.2 Dokumentacja konstruktora i destruktora

3.10.2.1 Zestaw() [1/2]

```
Zestaw::Zestaw ( )
```

Konstruktor bez parametrow.

3.10.2.2 Zestaw() [2/2]

```
Zestaw::Zestaw (
    string nazwaTowaru )
```

Konstruktor ustawiający nazwę towaru.

Parametry

<i>nazwaTowaru</i>	nazwa towaru;
--------------------	---------------

3.10.2.3 ~Zestaw()

```
Zestaw::~~Zestaw ( )
```

Destruktor zwalniający pamięć.

3.10.3 Dokumentacja funkcji składowych

3.10.3.1 cenaBrutto()

```
float Zestaw::cenaBrutto ( ) [virtual]
```

Metoda wyliczająca i zwracająca kwotę brutto.

Implementuje [Towar](#).

3.10.3.2 cenaNetto()

```
float Zestaw::cenaNetto ( ) [virtual]
```

Metoda wyliczająca i zwracająca kwotę netto.

Implementuje [Towar](#).

3.10.3.3 dodajTowar()

```
void Zestaw::dodajTowar (
    Artykul * artykul )
```

Metoda dodająca artykuł do listy towarów.

Parametry

<i>artykul</i>	typ towaru;
----------------	-------------

3.10.3.4 kwotaVAT()

```
float Zestaw::kwotaVAT ( ) [virtual]
```

Metoda wyliczająca i zwracająca kwotę VAT.

Implementuje [Towar](#).

3.10.3.5 stawkaVAT()

```
int Zestaw::stawkaVAT ( ) [virtual]
```

Metoda zwracająca pustą stawkę VAT.

Implementuje [Towar](#).

3.10.3.6 vatString()

```
string Zestaw::vatString ( ) [virtual]
```

Metoda zwracająca pusty VAT;

Implementuje [Towar](#).

Dokumentacja dla tej klasy została wygenerowana z plików:

- Zestaw.h
- Zestaw.cpp

Indeks

~Zestaw

Zestaw, [27](#)

Artykul, [5](#)

Artykul, [6](#)

cenaBrutto, [6](#)

cenaNetto, [6](#)

kwotaVAT, [6](#)

stawkaVAT, [7](#)

vatString, [7](#)

cenaBrutto

Artykul, [6](#)

Usluga, [23](#)

Zestaw, [27](#)

cenaNetto

Artykul, [6](#)

Usluga, [24](#)

Zestaw, [27](#)

dodajFakture

KreatorFaktur, [20](#)

dodajTowar

Faktura, [9](#)

Zestaw, [28](#)

Faktura, [7](#)

dodajTowar, [9](#)

Faktura, [8](#)

getCzyOplacone, [9](#)

getDataSprzedazy, [9](#)

getFormaPlatnosci, [9](#)

getKlient, [9](#)

getKwotaBrutto, [10](#)

getKwotaNetto, [10](#)

getListaTowarow, [10](#)

getMiejsceWystawienia, [10](#)

getNumerFaktury, [10](#)

getSprzedawca, [10](#)

getUwagi, [10](#)

setCzyOplacone, [10](#)

setDataSprzedazy, [11](#)

setFormaPlatnosci, [11](#)

setKlient, [11](#)

setMiejsceWystawienia, [11](#)

setNumerFaktury, [12](#)

setSprzedawca, [12](#)

setUwagi, [12](#)

suma, [13](#)

wypisz, [13](#)

wypiszNumerFaktury, [13](#)

getCzyOplacone

Faktura, [9](#)

getDataSprzedazy

Faktura, [9](#)

getFormaPlatnosci

Faktura, [9](#)

getIlosc

Towar, [22](#)

getKlient

Faktura, [9](#)

getKwotaBrutto

Faktura, [10](#)

getKwotaNetto

Faktura, [10](#)

getListaTowarow

Faktura, [10](#)

getMiejsceWystawienia

Faktura, [10](#)

getNumerFaktury

Faktura, [10](#)

getSprzedawca

Faktura, [10](#)

getUwagi

Faktura, [10](#)

Klient, [13](#)

KlientFirmowy, [14](#)

KlientFirmowy, [16](#)

nazwa, [16](#)

wypisz, [17](#)

KlientIndywidualny, [17](#)

KlientIndywidualny, [18](#)

nazwa, [18](#)

wypisz, [18](#)

KreatorFaktur, [19](#)

dodajFakture, [20](#)

KreatorFaktur, [19](#)

usunFakture, [20](#)

wyczyscPamiec, [20](#)

wypisywanieKlientow, [20](#)

wypisywanieTowarow, [20](#)

wypiszFakture, [20](#)

wypiszListeFaktur, [20](#)

kwotaVAT

Artykul, [6](#)

Usluga, [24](#)

Zestaw, [28](#)

nazwa
 KlientFirmowy, 16
 KlientIndywidualny, 18
 Towar, 22

setCzyOpłacone
 Faktura, 10

setDataSprzedazy
 Faktura, 11

setFormaPlatnosci
 Faktura, 11

setIlosc
 Towar, 22

setKlient
 Faktura, 11

setMiejsceWystawienia
 Faktura, 11

setNumerFaktury
 Faktura, 12

setSprzedawca
 Faktura, 12

setUwagi
 Faktura, 12

stawkaVAT
 Artykul, 7
 Usługa, 24
 Zestaw, 28

suma
 Faktura, 13

Towar, 21
 getIlosc, 22
 nazwa, 22
 setIlosc, 22
 wypisz, 22

Usługa, 22
 cenaBrutto, 23
 cenaNetto, 24
 kwotaVAT, 24
 stawkaVAT, 24
 Usługa, 23
 vatString, 24

usunFakture
 KreatorFaktur, 20

vatString
 Artykul, 7
 Usługa, 24
 Zestaw, 28

wczytajKlientow
 WczytywaczDanych, 25

wczytajSprzedawce
 WczytywaczDanych, 25

wczytajTowary
 WczytywaczDanych, 25

WczytywaczDanych, 25
 wczytajKlientow, 25
 wczytajSprzedawce, 25
 wczytajTowary, 25

wypiszFakture
 KreatorFaktur, 20

wypiszListeFaktur
 KreatorFaktur, 20

wypiszNumerFaktury
 Faktura, 13

Zestaw, 26
 ~Zestaw, 27
 cenaBrutto, 27
 cenaNetto, 27
 dodajTowar, 28
 kwotaVAT, 28
 stawkaVAT, 28
 vatString, 28
 Zestaw, 27