

Klinika Weterynaryjna - Julia Bigaj

Generated by Doxygen 1.9.7

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 CzytaczGrafikuPracy Class Reference	7
4.1.1 Detailed Description	8
4.2 CzytaczLekarza Class Reference	8
4.2.1 Detailed Description	9
4.3 CzytaczPlikow Class Reference	9
4.3.1 Detailed Description	9
4.3.2 Member Function Documentation	10
4.3.2.1 dopiszDoPliku()	10
4.3.2.2 getNextelid()	10
4.3.2.3 getPozycjaTextu()	10
4.3.2.4 getPozycjaWiersza()	11
4.3.2.5 podajNazwePliku()	11
4.3.2.6 usunWieszPold()	11
4.3.2.7 wczytajPlik()	12
4.3.2.8 zapiszDoPliku()	12
4.3.2.9 zmaiienNaWiersze()	12
4.4 CzytaczUzytkownikow Class Reference	12
4.4.1 Detailed Description	13
4.4.2 Member Function Documentation	13
4.4.2.1 podajNazwePliku()	13
4.5 CzytaczWizyt Class Reference	14
4.5.1 Detailed Description	15
4.5.2 Member Function Documentation	15
4.5.2.1 podajNazwePliku()	15
4.6 CzytaczWlasicielaZwierzecia Class Reference	15
4.6.1 Detailed Description	16
4.7 CzytaczZwierzat Class Reference	16
4.7.1 Detailed Description	17
4.7.2 Member Function Documentation	17
4.7.2.1 podajNazwePliku()	17
4.8 DataGodzinaWizyty Class Reference	18
4.8.1 Detailed Description	18
4.8.2 Member Function Documentation	18

4.8.2.1	getDzien()	18
4.8.2.2	getGodzina()	19
4.8.2.3	getMiesiac()	19
4.8.2.4	getRok()	19
4.9	GodzinyPracy Class Reference	19
4.9.1	Detailed Description	20
4.9.2	Member Function Documentation	20
4.9.2.1	getGodzinaRozpoczecia()	20
4.9.2.2	getGodzinaZakonczenia()	20
4.9.2.3	getMinutaRozpoczecia()	20
4.9.2.4	getMinutaZakonczenia()	21
4.10	Kanarek Class Reference	21
4.10.1	Detailed Description	22
4.10.2	Member Function Documentation	23
4.10.2.1	dzwiek()	23
4.10.2.2	getWaga()	23
4.11	Kot Class Reference	23
4.11.1	Detailed Description	24
4.11.2	Member Function Documentation	25
4.11.2.1	dzwiek()	25
4.11.2.2	getWaga()	25
4.12	Lekarz Class Reference	25
4.12.1	Detailed Description	26
4.12.2	Member Function Documentation	26
4.12.2.1	czyDostepnyWGodzinach()	26
4.12.2.2	getImie()	26
4.12.2.3	getNazwisko()	27
4.12.2.4	getSpecjalizacja()	27
4.12.2.5	zwracGodzinyPracyDlaDnia()	27
4.13	MenuLekarza Class Reference	27
4.13.1	Detailed Description	28
4.13.2	Member Function Documentation	28
4.13.2.1	wyswietlMenuPocatkowe()	28
4.14	MenuLogowania Class Reference	28
4.14.1	Detailed Description	29
4.14.2	Member Function Documentation	29
4.14.2.1	getHaslo()	29
4.14.2.2	getLogin()	29
4.14.2.3	wyswietlMenuPocatkowe()	29
4.15	MenuWlasicielaZwierzecia Class Reference	29
4.15.1	Detailed Description	30
4.15.2	Member Function Documentation	30

4.15.2.1	wswietlMenuUslug()	30
4.15.2.2	wyswietlMenuDoPlanowaniaWizyty()	30
4.15.2.3	wyswietlMenuPocztkowe()	31
4.15.2.4	wyswietlMenuWyboruTerminu()	31
4.16	Pajak Class Reference	31
4.16.1	Detailed Description	32
4.16.2	Member Function Documentation	33
4.16.2.1	dzwiek()	33
4.16.2.2	getWaga()	33
4.17	Pies Class Reference	33
4.17.1	Detailed Description	34
4.17.2	Member Function Documentation	35
4.17.2.1	dzwiek()	35
4.17.2.2	getWaga()	35
4.18	PlanerWizyt Class Reference	35
4.18.1	Detailed Description	37
4.18.2	Member Function Documentation	37
4.18.2.1	czyJestDniem()	37
4.18.2.2	czyOstatniDostepnyMiesiac()	37
4.18.2.3	czyPierwszDostepnyMiesiac()	38
4.18.2.4	dodajDzien()	38
4.18.2.5	dodajGodzine()	39
4.18.2.6	dodajGodzinePlusZaokraglenie()	39
4.18.2.7	dodajMiesiac()	39
4.18.2.8	getGodzinaZamknieniaWDniu()	40
4.18.2.9	getMiesiac()	40
4.18.2.10	ostatniDzienMiesiaca()	40
4.18.2.11	ustawCzas()	41
4.18.2.12	utworzWizyte()	41
4.18.2.13	wygenerujBlokiGodzinowe()	41
4.18.2.14	wygenerujDni()	42
4.18.2.15	wygenerujIdWizyty()	42
4.18.2.16	wygenerujMiesiace()	43
4.18.2.17	wypiszKalendarz()	43
4.18.2.18	wyswietl()	43
4.18.2.19	zaladujAktualnyDzienTygodnia()	44
4.18.2.20	zaladujDostepnaGodzineNaDzienTygodnia()	44
4.18.2.21	zaladujDostepyCzas()	44
4.18.2.22	zaladujDzienMiesiaca()	46
4.18.2.23	zaladujMiesiac()	46
4.18.2.24	zaladujNastepnyDzienTygodnia()	46
4.18.2.25	zaladujOstatniaDostepnaDate()	47

4.18.2.26	załadujPierszaDostepnaDate()	47
4.18.2.27	załadujRok()	47
4.18.2.28	zamienNaDniTygonida() [1/2]	48
4.18.2.29	zamienNaDniTygonida() [2/2]	48
4.18.2.30	zamienNaMiesiac()	48
4.18.2.31	zamienZAngielskiegoNaPolski()	49
4.18.2.32	znajdzPierwszyDostepnyTermin()	49
4.19	SerwisGrafikuPracy Class Reference	49
4.19.1	Detailed Description	50
4.19.2	Member Function Documentation	50
4.19.2.1	załadujGrafik()	50
4.20	SerwisLekarza Class Reference	50
4.20.1	Detailed Description	50
4.20.2	Member Function Documentation	51
4.20.2.1	załadujLekarza()	51
4.20.2.2	załadujWszystkichLekarzy()	51
4.20.2.3	znajdzLekarzaPold()	51
4.21	SerwisLogowaniaUzytkownika Class Reference	52
4.21.1	Detailed Description	52
4.21.2	Member Function Documentation	52
4.21.2.1	zaloguj()	52
4.21.2.2	znajdz()	52
4.22	SerwisUslug Class Reference	53
4.22.1	Detailed Description	53
4.22.2	Member Function Documentation	53
4.22.2.1	stworzUsludeDlaZwierzaka()	53
4.22.2.2	znajdzPoTypie()	54
4.23	SerwisWizyt Class Reference	54
4.23.1	Detailed Description	55
4.23.2	Member Function Documentation	55
4.23.2.1	anulujWizyte()	55
4.23.2.2	anulujWizyteDlaZwierzaka()	55
4.23.2.3	getNastepneId()	55
4.23.2.4	przeglądWizytPoldLekarza()	56
4.23.2.5	przeglądWizytPoldWlasiciela()	56
4.23.2.6	załadujWizytyDlaDaty()	56
4.23.2.7	zarezerujWizyte()	57
4.24	SerwisWlasicielaZwierzecia Class Reference	57
4.24.1	Detailed Description	57
4.24.2	Member Function Documentation	58
4.24.2.1	załadujWlasicielaZwierzecia()	58
4.24.2.2	załadujWlasicielaZwierzeciaPold()	58

4.25 SerwisZwierzat Class Reference	58
4.25.1 Detailed Description	59
4.25.2 Member Function Documentation	59
4.25.2.1 dodaj()	59
4.25.2.2 getNextpneId()	59
4.25.2.3 usunZwierzaka()	60
4.25.2.4 zaktualizuj()	60
4.25.2.5 zaladujZwierzepold()	60
4.25.2.6 zaladujZwierzetaDlaWlasciciela()	60
4.25.2.7 znajdzPold()	61
4.26 SesjaLekarza Class Reference	61
4.26.1 Detailed Description	62
4.26.2 Member Function Documentation	62
4.26.2.1 getFabrykaUslug()	62
4.26.2.2 getLekarz()	62
4.26.2.3 getSerwisWizyt()	62
4.26.2.4 getSerwisWlascicielaZwierzecia()	62
4.26.2.5 getSerwisZwierzat()	63
4.27 SesjaWlascicielaZwierzecia Class Reference	63
4.27.1 Detailed Description	64
4.27.2 Member Function Documentation	64
4.27.2.1 czyMoznaUtworzycWizyte()	64
4.27.2.2 getIdWybranegoLekarza()	64
4.27.2.3 getWlascielZwierzecia()	64
4.27.2.4 getWybarneZwierzecia()	65
4.27.2.5 getWybranaGodzina()	65
4.27.2.6 getWybranaUslug()	65
4.27.2.7 getWybranyDzien()	65
4.27.2.8 getWybranyMiesiac()	65
4.27.2.9 getWybranyRok()	66
4.27.2.10 getZwierzeta()	66
4.27.2.11 setIdWybranegoLekarza()	66
4.27.2.12 setWybranaGodzina()	66
4.27.2.13 setWybranaUslug()	66
4.27.2.14 setWybraneZwierzecia()	67
4.27.2.15 setWybranyDzien()	67
4.27.2.16 setWybranyMiesiac()	67
4.27.2.17 setWybranyRok()	67
4.28 SprawdzaczLiczba Class Reference	68
4.28.1 Detailed Description	68
4.28.2 Member Function Documentation	68
4.28.2.1 jestLiczba()	68

4.29 Usługa Class Reference	69
4.29.1 Detailed Description	69
4.29.2 Member Function Documentation	69
4.29.2.1 getCena()	69
4.29.2.2 getNazwaUslugi()	69
4.29.2.3 getTypUslugi()	70
4.30 Uzytkownik Class Reference	70
4.30.1 Detailed Description	70
4.30.2 Member Function Documentation	71
4.30.2.1 getId()	71
4.30.2.2 getTypUzytkownika()	71
4.30.2.3 zamienNaTekst()	71
4.31 Waz Class Reference	71
4.31.1 Detailed Description	73
4.31.2 Member Function Documentation	73
4.31.2.1 dzwiek()	73
4.31.2.2 getWaga()	73
4.32 Wizyta Class Reference	73
4.32.1 Detailed Description	74
4.32.2 Member Function Documentation	74
4.32.2.1 getData()	74
4.32.2.2 getDataGodzina()	74
4.32.2.3 getGodzina()	74
4.32.2.4 getIdLekarza()	74
4.32.2.5 getIdWizyty()	75
4.32.2.6 getIdWlasciciela()	75
4.32.2.7 getIdZwierzecia()	75
4.32.2.8 getTypUslugi()	75
4.32.2.9 zamieNaTekst()	75
4.33 WlascicielZwierzecia Class Reference	76
4.33.1 Detailed Description	77
4.33.2 Member Function Documentation	77
4.33.2.1 getId()	77
4.33.2.2 getImie()	77
4.33.2.3 getKodPoczowy()	77
4.33.2.4 getMiasto()	78
4.33.2.5 getNumerTelefonu()	78
4.33.2.6 getUlicalNumerDomu()	78
4.33.2.7 setKodPocztowy()	78
4.33.2.8 setMiasto()	78
4.33.2.9 setNumerTelefonu()	80
4.33.2.10 setUlicalNumerDomu()	80

4.34 Zwierze Class Reference	80
4.34.1 Detailed Description	81
4.34.2 Member Function Documentation	82
4.34.2.1 dzwiek()	82
4.34.2.2 getId()	82
4.34.2.3 getIdWlasciciela()	82
4.34.2.4 getImie()	82
4.34.2.5 getKolor()	83
4.34.2.6 getIczaNog()	83
4.34.2.7 getPlec()	83
4.34.2.8 getUmaszczenie()	83
4.34.2.9 getWaga()	83
4.34.2.10 getWiek()	84
4.34.2.11 setImie()	84
4.34.2.12 setWagaWGramach()	84
4.34.2.13 zamienNaTekst()	84
4.34.3 Member Data Documentation	84
4.34.3.1 wagaWGramach	84
5 File Documentation	87
5.1 CzytaczGrafikuPracy.h File Reference	87
5.2 CzytaczGrafikuPracy.h	87
5.3 CzytaczLekarza.h File Reference	87
5.4 CzytaczLekarza.h	88
5.5 CzytaczPlikow.h File Reference	88
5.6 CzytaczPlikow.h	88
5.7 CzytaczUzytkownikow.h File Reference	89
5.8 CzytaczUzytkownikow.h	89
5.9 CzytaczWizyt.h File Reference	89
5.10 CzytaczWizyt.h	89
5.11 CzytaczWlascicielaZwierzecia.h	90
5.12 CzytaczZwierzat.h File Reference	90
5.13 CzytaczZwierzat.h	90
5.14 DniTygodnia.h File Reference	90
5.15 DniTygodnia.h	90
5.16 GodzinyPracy.h File Reference	91
5.17 GodzinyPracy.h	91
5.18 Kanarek.h File Reference	91
5.19 Kanarek.h	91
5.20 Kot.h File Reference	92
5.21 Kot.h	92
5.22 Lekarz.h File Reference	92

5.23 Lekarz.h	93
5.24 MenuLekarza.h File Reference	93
5.25 MenuLekarza.h	93
5.26 MenuLogowania.h File Reference	94
5.27 MenuLogowania.h	94
5.28 MenuLogowaniaStatusu.h File Reference	94
5.29 MenuLogowaniaStatusu.h	95
5.30 MenuWlasicielaZwierzecia.h	95
5.31 Pajak.h File Reference	95
5.32 Pajak.h	96
5.33 Pies.h File Reference	96
5.34 Pies.h	96
5.35 PlanerWizyt.h File Reference	96
5.36 PlanerWizyt.h	97
5.37 Plec.h File Reference	98
5.38 Plec.h	98
5.39 SerwisGrafikuPracy.h File Reference	98
5.40 SerwisGrafikuPracy.h	98
5.41 SerwisLekarza.h File Reference	99
5.42 SerwisLekarza.h	99
5.43 SerwisLogowaniaUzytkownika.h	99
5.44 SerwisUslug.h File Reference	100
5.45 SerwisUslug.h	100
5.46 SerwisWizyt.h File Reference	100
5.47 SerwisWizyt.h	101
5.48 SerwisWlasicielaZwierzecia.h	101
5.49 SerwisZwierzat.h File Reference	102
5.50 SerwisZwierzat.h	102
5.51 SesjaLekarza.h File Reference	102
5.52 SesjaLekarza.h	103
5.53 SesjaWlasicielaZwierzecia.h File Reference	103
5.54 SesjaWlasicielaZwierzecia.h	104
5.55 SprawdzaczLiczb.h File Reference	104
5.56 SprawdzaczLiczb.h	104
5.57 StatusWizyty.h	105
5.58 TypUzytkownika.h File Reference	105
5.59 TypUzytkownika.h	105
5.60 Usługa.h File Reference	105
5.61 Usługa.h	106
5.62 Uzytkownik.h File Reference	106
5.63 Uzytkownik.h	106
5.64 Waz.h File Reference	106

5.65 Waz.h	107
5.66 Wizyta.h File Reference	107
5.67 Wizyta.h	107
5.68 WlascicielZwierzecia.h File Reference	108
5.69 WlascicielZwierzecia.h	108
5.70 Zwierze.h File Reference	109
5.71 Zwierze.h	109
Index	111

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

CzytaczPlikow	9
CzytaczGrafikuPracy	7
CzytaczLekarza	8
CzytaczUzytkownikow	12
CzytaczWizyt	14
CzytaczWlasicielaZwierzecia	15
CzytaczZwierzat	16
DataGodzinaWizyty	18
GodzinyPracy	19
MenuLekarza	27
MenuLogowania	28
MenuWlasicielaZwierzecia	29
PlanerWizyt	35
SerwisGrafikuPracy	49
SerwisLekarza	50
SerwisLogowaniaUzytkownika	52
SerwisUslug	53
SerwisWizyt	54
SerwisWlasicielaZwierzecia	57
SerwisZwierzat	58
SesjaLekarza	61
SesjaWlasicielaZwierzecia	63
SprawdzaczLiczb	68
Usługa	69
Uzytkownik	70
Lekarz	25
WlasicielZwierzecia	76
Wizyta	73
Zwierze	80
Kanarek	21
Kot	23
Pajak	31
Pies	33
Waz	71

Chapter 2

Class Index

2.1 Class List

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

CzytaczGrafikuPracy	Klasa do czytania pliku, w którym jest zapisany grafik pracy kliniki	7
CzytaczLekarza	Klasa do czytania pliku, w którym są zapisani lekarze	8
CzytaczPlikow	Klasa bazowa do czytania pliku, po której dziedziczą inne klasy, które muszą zaimplementować funkcję <code>podajNazwePliku</code>	9
CzytaczUzytkownikow	Klasa bazowa do czytania użytkowników (lekarz, właściciel zwierzęcia)	12
CzytaczWizyt	Klasa bazowa do czytania wizyt	14
CzytaczWlasicielaZwierzecia	Klasa bazowa do czytania właściciela zwierzęci	15
CzytaczZwierzat	Klasa bazowa do czytania zwierząt	16
DataGodzinaWizyty	Klasa odpowiedzialna za datę i godzinę wizyty	18
GodzinyPracy	Klasa opisująca strukturę godzin pracy, gdzie mamy godzinę rozpoczęcia i zakończenia wraz z minutami	19
Kanarek	Klasa opisująca kanarkę, klasa pochodna. Dziedziczy po klasie Zwierze	21
Kot	Klasa opisująca kota, klasa pochodna. Dziedziczy po klasie Zwierze	23
Lekarz	Klasa opisująca lekarza, klasa pochodna. Dziedziczy po klasie Uzytkownik	25
MenuLekarza	Klasa opisująca całe menu lekarza	27
MenuLogowania	Klasa zajmująca się obsługą menu logowania	28
MenuWlasicielaZwierzecia	Klasa opisująca całe menu właściciela zwierzęcia	29
Pajak	Klasa opisująca pajaka, klasa pochodna. Dziedziczy po klasie Zwierze	31
Pies	Klasa opisująca psa, klasa pochodna. Dziedziczy po klasie Zwierze	33

PlanerWizyt	
Klasa odpowiedzialna za planowanie wizyt oraz wszelkie manipulacje data i czasem	35
SerwisGrafikuPracy	
Klasa odpowiedzialna za serwis grafiku pracy	49
SerwisLekarza	
Klasa odpowiedzialna za serwis lekarza	50
SerwisLogowaniaUzytkownika	
Klasa odpowiedzialna za serwis logowania uzytkownikow	52
SerwisUslug	
Klasa odpowiedzialna za serwis uslug	53
SerwisWizyt	
Klasa odpowiedzialna za serwis wizyt	54
SerwisWlascicielaZwierzecia	
Klasa odpowiedzialna za serwis wlasciciela zwierzecia	57
SerwisZwierzat	
Klasa odpowiedzialna za serwis zwierzat	58
SesjaLekarza	
Klasa odpowiedzialna za sesje lekarza (za jego wizyty)	61
SesjaWlascicielaZwierzecia	
Klasa odpowiedzialna za sesje wlasciciela zwierzecia	63
SprawdzaczLiczb	
Klasa odpowiedzialna za sprawdzenie czy to co jest wprowadzane jest liczba	68
Uslug	
Klasa odpowiedzialna za uslugi	69
Uzytkownik	
Klasa odpowiedzialna za uzytkownikow	70
Waz	
Klasa opisujaca weza, klasa pochodna. Dziedziczy po klasie Zwierze	71
Wizyta	
Klasa odpowiedzialna wizyte	73
WlascicielZwierzecia	
Klasa odpowiedzialna za wlasciciela zwierzecia	76
Zwierze	
Klasa odpowiedzialna za zwierzeta	80

Chapter 3

File Index

3.1 File List

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

CzytaczGrafikuPracy.h	87
CzytaczLekarza.h	87
CzytaczPlikow.h	88
CzytaczUzytkownikow.h	89
CzytaczWizyt.h	89
CzytaczWlasicielaZwierzecia.h	90
CzytaczZwierzat.h	90
DniTygodnia.h	90
GodzinyPracy.h	91
Kanarek.h	91
Kot.h	92
Lekarz.h	92
MenuLekarza.h	93
MenuLogowania.h	94
MenuLogowaniaStatusu.h	94
MenuWlasicielaZwierzecia.h	95
Pajak.h	95
Pies.h	96
PlanerWizyt.h	96
Plec.h	98
SerwisGrafikuPracy.h	98
SerwisLekarza.h	99
SerwisLogowaniaUzytkownika.h	99
SerwisUslug.h	100
SerwisWizyt.h	100
SerwisWlasicielaZwierzecia.h	101
SerwisZwierzat.h	102
SesjaLekarza.h	102
SesjaWlasicielaZwierzecia.h	103
SprawdzaczLiczb.h	104
StatusWizyty.h	105
TypUzytkownika.h	105
Usluga.h	105
Uzytkownik.h	106
Waz.h	106
Wizyta.h	107
WlascicielZwierzecia.h	108
Zwierze.h	109

Chapter 4

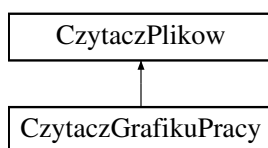
Class Documentation

4.1 CzytaczGrafikuPracy Class Reference

Klasa do czytania pliku, w którym jest zapisany grafik pracy kliniki.

```
#include <CzytaczGrafikuPracy.h>
```

Inheritance diagram for CzytaczGrafikuPracy:



Additional Inherited Members

Public Member Functions inherited from **CzytaczPlikow**

- string [wczytajPlik](#) ()
Funkcja zwracająca całą zawartość pliku.
- void [dopiszDoPliku](#) (string &nowyWiersz)
Funkcja dopisująca do pliku nowy wiersz.
- virtual string [podajNazwePliku](#) ()=0
Funkcja wirtualna, która musi zaimplementować klasy, które po niej dziedziczą. Funkcja ta musi zwracać ścieżkę do pliku.
- vector< string > [zmaiennaWiersze](#) (string &trescPliku)
Funkcja zamieniająca całą zawartość pliku na wektor typu string.
- string [getPozycjaTextu](#) (string całyTekst, int pozycja)
*Funkcja zwracająca tekst z danej kolumny. Np. string tekst = "abc|def" getPozycjaTextu(tekst, 0) -> zwróci abc
getPozycjaTextu(tekst, 1) -> zwróci def.*
- void [usunWierszPold](#) (int indeksId, int id, bool czyJestUnikalny)
Funkcja, która usuwa dany wiersz po danym id.
- int [getPozycjaWiersza](#) (vector< string > wiersze, string wiersz)
Funkcja zwracająca pozycję danego wiersza z podanego wektora stringów.
- void [zapiszDoPliku](#) (const vector< string > &wektor, const string &nazwaPliku)
Funkcja zapisująca do pliku w dany wektor stringów.
- int [getNextId](#) (int indeksId)
Funkcja zwracająca kolejne id, które możemy użyć przy zapisaniu do pliku. Np. przykładowy plik 1|a 2|b 3|c.

4.1.1 Detailed Description

Klasa do czytania pliku, w którym jest zapisany grafik pracy kliniki.

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

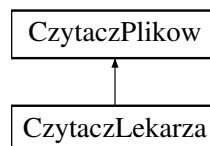
- [CzytaczGrafikuPracy.h](#)
- [CzytaczGrafikuPracy.cpp](#)

4.2 CzytaczLekarza Class Reference

Klasa do czytania pliku, w którym są zapisani lekarze.

```
#include <CzytaczLekarza.h>
```

Inheritance diagram for CzytaczLekarza:



Additional Inherited Members

Public Member Functions inherited from [CzytaczPlikow](#)

- string [wczytajPlik](#) ()
Funkcja zwracająca całą zawartość pliku.
- void [dopiszDoPliku](#) (string &nowyWiersz)
Funkcja dopisująca do pliku nowy wiersz.
- virtual string [podajNazwePliku](#) ()=0
Funkcja wirtualna, która musi być zaimplementowana w klasach, które po niej dziedziczą. Funkcja ta musi zwracać ścieżkę do pliku.
- vector< string > [zmaiennaWiersze](#) (string &trescPliku)
Funkcja zamieniająca całą zawartość pliku na wektor typu string.
- string [getPozycjaTekstu](#) (string całyTekst, int pozycja)
*Funkcja zwracająca tekst z danej kolumny. Np. string tekst = "abc|def" getPozycjaTekstu(tekst, 0) -> zwróci abc
getPozycjaTekstu(tekst, 1) -> zwróci def.*
- void [usunWierszPold](#) (int indeksId, int id, bool czyJestUnikalny)
Funkcja, która usuwa dany wiersz po danym id.
- int [getPozycjaWiersza](#) (vector< string > wiersze, string wiersz)
Funkcja zwracająca pozycję danego wiersza z podanego wektora stringów.
- void [zapiszDoPliku](#) (const vector< string > &wektor, const string &nazwaPliku)
Funkcja zapisująca do pliku w dany wektor stringów.
- int [getNextId](#) (int indeksId)
Funkcja zwracająca kolejne id, które możemy użyć przy zapisaniu do pliku. Np. przykładowy plik 1|a 2|b 3|c.

4.2.1 Detailed Description

Klasa do czytania pliku, w którym są zapisani lekarze.

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

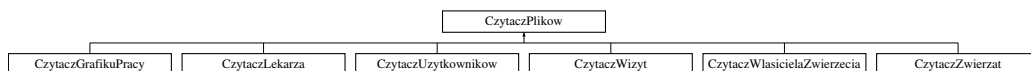
- [CzytaczLekarza.h](#)
- [CzytaczLekarza.cpp](#)

4.3 CzytaczPlikow Class Reference

Klasa bazowa do czytania pliku, po której dziedziczą inne klasy, które muszą zaimplementować funkcję `podajNazwePliku`.

```
#include <CzytaczPlikow.h>
```

Inheritance diagram for CzytaczPlikow:



Public Member Functions

- `string` [wczytajPlik](#) ()
Funkcja zwracająca całą zawartość pliku.
- `void` [dopiszDoPliku](#) (string &nowyWiersz)
Funkcja dopisująca do pliku nowy wiersz.
- `virtual string` [podajNazwePliku](#) ()=0
Funkcja wirtualna, którą muszą zaimplementować klasy, które po niej dziedziczą. Funkcja ta musi zwracać ścieżkę do pliku.
- `vector< string >` [zmaienNaWiersze](#) (string &trecPliku)
Funkcja zamieniająca całą zawartość pliku na wektor typu string.
- `string` [getPozycjaTekstu](#) (string całyTekst, int pozycja)
*Funkcja zwracająca tekst z danej kolumny. Np. string tekst = "abc|def" getPozycjaTekstu(tekst, 0) -> zwróci abc
getPozycjaTekstu(tekst, 1) -> zwróci def.*
- `void` [usunWieszPold](#) (int indeksId, int id, bool czyJestUnikalny)
Funkcja, która usuwa dany wiersz po danym id.
- `int` [getPozycjaWiersza](#) (vector< string > wiersze, string wiersz)
Funkcja zwracająca pozycję danego wiersza z podanego wektora stringów.
- `void` [zapiszDoPliku](#) (const vector< string > &wektor, const string &nazwaPliku)
Funkcja zapisująca do pliku w dany wektor stringów.
- `int` [getNextId](#) (int indeksId)
Funkcja zwracająca kolejne id, które możemy użyć przy zapisaniu do pliku. Np. przykładowy plik 1|a 2|b 3|c.

4.3.1 Detailed Description

Klasa bazowa do czytania pliku, po której dziedziczą inne klasy, które muszą zaimplementować funkcję `podajNazwePliku`.

4.3.2 Member Function Documentation

4.3.2.1 dopiszDoPliku()

```
void CzytaczPlikow::dopiszDoPliku (
    string & nowyWiersz )
```

Funkcja dopisująca do pliku nowy wiersz.

Parameters

<i>nowy</i>	wiersz, który zosatnie dopisany do istniejacego pliku
-------------	---

4.3.2.2 getNextId()

```
int CzytaczPlikow::getNextId (
    int indeksId )
```

Funkcja zwracająca kolejne id, które możemy użyć przy zapisie do pliku. Np. przykładowy plik 1|a 2|b 3|c.

int getNextId(0); -> zwróci wartość 4 - będzie to kolejne ID, które można użyć przy zapisie kolejnego wiersza

Returns

Kolejne dostępne ID

Parameters

<i>indeksId</i>	- indeks kolumny, od którego chcemy znaleźć kolejne ID
-----------------	--

4.3.2.3 getPozycjaTekstu()

```
string CzytaczPlikow::getPozycjaTekstu (
    string calyTekst,
    int pozycja )
```

Funkcja zwracająca tekst z danej kolumny. Np. string tekst = "abc|def" getPozycjaTekstu(tekst, 0) -> zwróci abc
getPozycjaTekstu(tekst, 1) -> zwróci def.

Returns

Tekst z danej kolumny

Parameters

<i>calyTekst</i>	- cały wiersz z pliku
<i>pozycja</i>	- indeks pozycji, której chcemy wyciągnąć dany tekst

4.3.2.4 getPozycjaWiersza()

```
int CzytaczPlikow::getPozycjaWiersza (
    vector< string > wiersze,
    string wiersz )
```

Funkcja zwracająca pozycje danego wiersza z podanego wektora stringow.

Returns

Indeks wiersza

Parameters

<i>wiersze</i>	- wektor wierszy
<i>wiersz</i>	- tekst, którego pozycje chcemy znalezc w wektorze wierszy

4.3.2.5 podajNazwePliku()

```
virtual string CzytaczPlikow::podajNazwePliku ( ) [pure virtual]
```

Funkcja wirtualna, która musi być zaimplementowana przez klasy, które po niej dziedziczą. Funkcja ta musi zwracać ścieżkę do pliku.

Returns

ścieżka do pliku

Implemented in [CzytaczUzytkownikow](#), [CzytaczWizyt](#), and [CzytaczZwierzat](#).

4.3.2.6 usunWieszPoid()

```
void CzytaczPlikow::usunWieszPoId (
    int indeksId,
    int id,
    bool czyJestUnikalny )
```

Funkcja, która usuwa dany wiersz po danym id.

Parameters

<i>indeksID</i>	- indeks kolumny, z której chcemy usunąć daną wartość
<i>id</i>	- numer który ma zostać usunięty
<i>czyJestUnikalny</i>	- false jeśli jest szansa, że id o takiej samej wartości występuje więcej niż raz w pliku. True jeśli to id jest unikalne w ramach jednego pliku

4.3.2.7 wczytajPlik()

```
string CzytaczPlikow::wczytajPlik ( )
```

Funkcja zwracająca całą zawartość pliku.

Returns

Zawartość całego pliku

4.3.2.8 zapiszDoPliku()

```
void CzytaczPlikow::zapiszDoPliku (
    const vector< string > & wektor,
    const string & nazwaPliku )
```

Funkcja zapisująca do pliku w dany wektor stringów.

Parameters

<i>wektor</i>	- wektor stringów, w którym znajduje się treść do zapisania do pliku
<i>nazwaPliku</i>	- nazwa pliku, do którego chcemy zapisać cały tekst

4.3.2.9 zmaienNaWiersze()

```
vector< string > CzytaczPlikow::zmaienNaWiersze (
    string & trescPliku )
```

Funkcja zamieniająca całą zawartość pliku na wektor typu string.

Returns

Wektor typu string. Każdy element kolekcji wektora reprezentuje dany wiersz. Np. pierwszy element będzie pierwszym wierszem w pliku.

Parameters

<i>tresc</i>	całego pliku
--------------	--------------

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

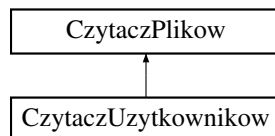
- [CzytaczPlikow.h](#)
- [CzytaczPlikow.cpp](#)

4.4 CzytaczUzytkownikow Class Reference

Klasa bazowa do czytania użytkowników (lekarz, właściciel zwierzęcia)


```
#include <CzytaczUzytkownikow.h>
```

Inheritance diagram for CzytaczUzytkownikow:



Public Member Functions

- string [podajNazwePliku](#) ()
Funkcja zwracajace sciezke do pliku, w ktorym znajduja sie uzytkownicy.

Public Member Functions inherited from [CzytaczPlikow](#)

- string [wczytajPlik](#) ()
Funkcja zwracajaca cala zawartosc pliku.
- void [dopiszDoPliku](#) (string &nowyWiersz)
Funkcja dopisujaca do pliku nowy wiersz.
- virtual string [podajNazwePliku](#) ()=0
Funkcja wirtualna, ktora musza zaimplementowac klasy, ktore po niej dziedzicza Funkcja ta musi zwracac sciezke do pliku.
- vector< string > [zmaienNaWiersze](#) (string &trescPliku)
Funkcja zamienijaca cala zawartosc pliku na wektor typu string.
- string [getPozycjaTextu](#) (string calyTekst, int pozycja)
*Funkcja zwracajaca tekst z danej kolumny Np. string tekst = "abc|def" getPozycjaTextu(tekst, 0) -> zwroci abc
getPozycjaTextu(tekst, 1) -> zwroci def.*
- void [usunWieszPold](#) (int indeksId, int id, bool czyJestUnikalny)
Funkcja, kora usuwa dany wiersz po danym id.
- int [getPozycjaWiersza](#) (vector< string > wiersze, string wiersz)
Funkcja zwracajaca pozycje danego wiersza z podanego wektora stringow.
- void [zapiszDoPliku](#) (const vector< string > &wektor, const string &nazwaPliku)
Funkcja zapisujaca do pliku w dany wektor stringow.
- int [getNextId](#) (int indeksId)
Funkcja zwracajaca kolejne id, ktore mozemy uzyc przy zapisie do pliku Np. przykladowy plik 1|a 2|b 3|c.

4.4.1 Detailed Description

Klasa bazowa do czytania uzytkownikow (lekarz, wlasciciel zwierzecia)

4.4.2 Member Function Documentation

4.4.2.1 [podajNazwePliku\(\)](#)

```
std::string CzytaczUzytkownikow::podajNazwePliku ( ) [virtual]
```

Funkcja zwracajace sciezke do pliku, w ktorym znajduja sie uzytkownicy.

Returns

sciezka do pliku zaspiana w typie tekstowym (String)

Parameters

<i>brak</i>	parametrow
-------------	------------

Implements [CzytaczPlikow](#).

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

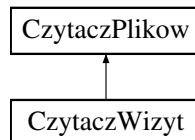
- [CzytaczUzytkownikow.h](#)
- [CzytaczUzytkownikow.cpp](#)

4.5 CzytaczWizyt Class Reference

Klasa bazowa do czytania wizyt.

```
#include <CzytaczWizyt.h>
```

Inheritance diagram for CzytaczWizyt:



Public Member Functions

- string [podajNazwePliku](#) ()
Funkcja zwracajace sciezke do pliku, w ktorym znajduja sie wszystkie wizyty.

Public Member Functions inherited from [CzytaczPlikow](#)

- string [wczytajPlik](#) ()
Funkcja zwracajajaca cala zawartosc pliku.
- void [dopiszDoPliku](#) (string &nowyWiersz)
Funkcja dopisujaca do pliku nowy wiersz.
- virtual string [podajNazwePliku](#) ()=0
Funkcja wirtualna, ktora musza zaimplementowac klasy, ktore po niej dziedzicza Funkcja ta musi zwracac sciezke do pliku.
- vector< string > [zmaiennaWiersze](#) (string &trescPliku)
Funkcja zamienijaca cala zawartosc pliku na wektor typu string.
- string [getPozycjaTekstu](#) (string calyTekst, int pozycja)
*Funkcja zwracajajaca tekst z danej kolumny Np. string tekst = "abc|def" getPozycjaTekstu(tekst, 0) -> zwroci abc
getPozycjaTekstu(tekst, 1) -> zwroci def.*
- void [usunWieszPold](#) (int indeksId, int id, bool czyJestUnikalny)
Funkcja, kora usuwa dany wiersz po danym id.
- int [getPozycjaWiersza](#) (vector< string > wiersze, string wiersz)
Funkcja zwracajaca pozycje danego wiersza z podanego wektora stringow.
- void [zapiszDoPliku](#) (const vector< string > &wektor, const string &nazwaPliku)
Funkcja zapisujaca do pliku w dany wektor stringow.
- int [getNextPole](#) (int indeksId)
Funkcja zwracajaca kolejne id, ktore mozemy uzyc przy zapisnie do pliku Np. przykladowy plik 1|a 2|b 3|c.

4.5.1 Detailed Description

Klasa bazowa do czytania wizyt.

4.5.2 Member Function Documentation

4.5.2.1 podajNazwePliku()

```
std::string CzytaczWizyt::podajNazwePliku ( ) [virtual]
```

Funkcja zwracajace sciezke do pliku, w ktorym znajduja sie wszystkie wizyty.

Returns

sciezka do pliku zaspiana w typie tekstowym (String)

Parameters

<i>brak</i>	parametrow
-------------	------------

Implements [CzytaczPlikow](#).

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

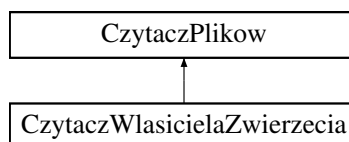
- [CzytaczWizyt.h](#)
- CzytaczWizyt.cpp

4.6 CzytaczWlasicielaZwierzecia Class Reference

Klasa bazowa do czytania wlasciciela zwierzeci.

```
#include <CzytaczWlasicielaZwierzecia.h>
```

Inheritance diagram for CzytaczWlasicielaZwierzecia:



Additional Inherited Members

Public Member Functions inherited from [CzytaczPlikow](#)

- string [wczytajPlik](#) ()
Funkcja zwracająca całą zawartość pliku.
- void [dopiszDoPliku](#) (string &nowyWiersz)
Funkcja dopisująca do pliku nowy wiersz.
- virtual string [podajNazwePliku](#) ()=0
Funkcja wirtualna, która musi zaimplementować klasy, które po niej dziedziczą. Funkcja ta musi zwracać ścieżkę do pliku.
- vector< string > [zmaiennaWiersze](#) (string &trecPliku)
Funkcja zamieniająca całą zawartość pliku na wektor typu string.
- string [getPozycjaTekstu](#) (string całyTekst, int pozycja)
*Funkcja zwracająca tekst z danej kolumny. Np. string tekst = "abc|def" getPozycjaTekstu(tekst, 0) -> zwróci abc
getPozycjaTekstu(tekst, 1) -> zwróci def.*
- void [usunWierszPold](#) (int indeksId, int id, bool czyJestUnikalny)
Funkcja, która usuwa dany wiersz po danym id.
- int [getPozycjaWiersza](#) (vector< string > wiersze, string wiersz)
Funkcja zwracająca pozycję danego wiersza z podanego wektora stringów.
- void [zapiszDoPliku](#) (const vector< string > &wektor, const string &nazwaPliku)
Funkcja zapisująca do pliku w dany wektor stringów.
- int [getNextId](#) (int indeksId)
Funkcja zwracająca kolejne id, które możemy użyć przy zapisaniu do pliku. Np. przykładowy plik 1|a 2|b 3|c.

4.6.1 Detailed Description

Klasa bazowa do czytania właściciela zwierzęci.

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

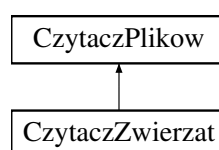
- CzytaczWlasicielaZwierzecia.h
- CzytaczWlasicielaZwierzecia.cpp

4.7 CzytaczZwierzat Class Reference

Klasa bazowa do czytania zwierząt.

```
#include <CzytaczZwierzat.h>
```

Inheritance diagram for CzytaczZwierzat:



Public Member Functions

- string [podajNazwePliku](#) ()

Funkcja zwracajace sciezke do pliku, w ktorym znajduja sie zwierzeta.

Public Member Functions inherited from [CzytaczPlikow](#)

- string [wczytajPlik](#) ()

Funkcja zwracajajaca cala zawartosc pliku.

- void [dopiszDoPliku](#) (string &nowyWiersz)

Funkcja dopisujaca do pliku nowy wiersz.

- virtual string [podajNazwePliku](#) ()=0

Funkcja wirtualna, ktora musza zaimplementowac klasy, ktore po niej dziedzicza Funkcja ta musi zwracac sciezke do pliku.

- vector< string > [zmaienNaWiersze](#) (string &trecPliku)

Funkcja zamienijaca cala zawartosc pliku na wektor typu string.

- string [getPozycjaTextu](#) (string calyTekst, int pozycja)

*Funkcja zwracajajaca tekst z danej kolumny Np. string tekst = "abc|def" getPozycjaTextu(tekst, 0) -> zwroci abc
getPozycjaTextu(tekst, 1) -> zwroci def.*

- void [usunWieszPold](#) (int indeksId, int id, bool czyJestUnikalny)

Funkcja, kora usuwa dany wiersz po danym id.

- int [getPozycjaWiersza](#) (vector< string > wiersze, string wiersz)

Funkcja zwracajaca pozycje danego wiersza z podanego wektora stringow.

- void [zapiszDoPliku](#) (const vector< string > &wektor, const string &nazwaPliku)

Funkcja zapisujaca do pliku w dany wektor stringow.

- int [getNastepnelid](#) (int indeksId)

Funkcja zwracajaca kolejne id, ktore mozemy uzyc przy zapisnie do pliku Np. przykladowy plik 1|a 2|b 3|c.

4.7.1 Detailed Description

Klasa bazowa do czytania zwierzat.

4.7.2 Member Function Documentation

4.7.2.1 [podajNazwePliku\(\)](#)

```
std::string CzytaczZwierzat::podajNazwePliku ( ) [virtual]
```

Funkcja zwracajace sciezke do pliku, w ktorym znajduja sie zwierzeta.

Returns

sciezka do pliku zaspiana w typie tekstowym (String)

Parameters

<i>brak</i>	parametrow
-------------	------------

Implements [CzytaczPlikow](#).

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

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

4.8 DataGodzinaWizyty Class Reference

Klasa odpowiedzialna za date i godzinę wizyty.

```
#include <Wizyta.h>
```

Public Member Functions

- **DataGodzinaWizyty** (int godzina, int dzien, int miesiac, int rok)
Konstruktor argumentowy, ustawiający godzinę, dzień, miesiąc, rok.
- **DataGodzinaWizyty** ()
Konstruktor bezargumentowy.
- int [getDzien](#) ()
Funkcja zwracająca dzień, ustawiamy w konstruktorze.
- int [getMiesiac](#) ()
Funkcja zwracająca miesiąc, ustawiamy w konstruktorze.
- int [getRok](#) ()
Funkcja zwracająca rok, ustawiamy w konstruktorze.
- int [getGodzina](#) ()
Funkcja zwracająca godzinę, ustawiamy w konstruktorze.

4.8.1 Detailed Description

Klasa odpowiedzialna za date i godzinę wizyty.

4.8.2 Member Function Documentation

4.8.2.1 getDzien()

```
int DataGodzinaWizyty::getDzien ( )
```

Funkcja zwracająca dzień, ustawiamy w konstruktorze.

Returns

dzien

4.8.2.2 getGodzina()

```
int DataGodzinaWizyty::getGodzina ( )
```

Funkcja zwracająca godzinę, ustawiamy w konstruktorze.

Returns

godzina

4.8.2.3 getMiesiac()

```
int DataGodzinaWizyty::getMiesiac ( )
```

Funkcja zwracająca miesiąc, ustawiamy w konstruktorze.

Returns

miesiac

4.8.2.4 getRok()

```
int DataGodzinaWizyty::getRok ( )
```

Funkcja zwracająca rok, ustawiamy w konstruktorze.

Returns

rok

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

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

4.9 GodzinyPracy Class Reference

Klasa opisująca strukturę godzin pracy, gdzie mamy godzinę rozpoczęcia i zakończenia wraz z minutami.

```
#include <GodzinyPracy.h>
```

Public Member Functions

- **GodzinyPracy** ()
Konstruktor domyslny, bezargumentowy.
- **GodzinyPracy** (int godzinaRozpoczecia, int minutaRozpoczecia, int godzinaZakonczenia, int minutaZakonczenia)
Konstruktor argumentowy ustawia godziny, minuty zakonczenia i rozpoczecia.
- int [getGodzinaRozpoczecia](#) ()
Funkcja zwracajaca godziny rozpoczecia.
- int [getGodzinaZakonczenia](#) ()
Funkcja zwracajaca godziny zakonczenia.
- int [getMinutaRozpoczecia](#) ()
Funkcja zwracajaca minuty rozpoczecia.
- int [getMinutaZakonczenia](#) ()
Funkcja zwracajaca minuty zakonczenia.

4.9.1 Detailed Description

Klasa opisujaca strukture godzin pracy, gdzie mamy godzinie rozpoczecia i zakonczenia wraz z minutami.

4.9.2 Member Function Documentation

4.9.2.1 [getGodzinaRozpoczecia\(\)](#)

```
int GodzinyPracy::getGodzinaRozpoczecia ( )
```

Funkcja zwracajaca godziny rozpoczecia.

Returns

Godziny rozpoczecia

4.9.2.2 [getGodzinaZakonczenia\(\)](#)

```
int GodzinyPracy::getGodzinaZakonczenia ( )
```

Funkcja zwracajaca godziny zakonczenia.

Returns

Godziny Zakonczenia

4.9.2.3 [getMinutaRozpoczecia\(\)](#)

```
int GodzinyPracy::getMinutaRozpoczecia ( )
```

Funkcja zwracajaca minuty rozpoczecia.

Returns

Minuty rozpoczecia

4.9.2.4 getMinutaZakonczenia()

```
int GodzinyPracy::getMinutaZakonczenia ( )
```

Funkcja zwracająca minuty zakończenia.

Returns

Minuty zakończenia

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

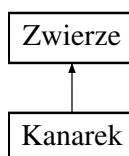
- [GodzinyPracy.h](#)
- [GodzinyPracy.cpp](#)

4.10 Kanarek Class Reference

Klasa opisująca kanarka, klasa pochodna. Dziedziczy po klasie [Zwierze](#).

```
#include <Kanarek.h>
```

Inheritance diagram for Kanarek:



Public Member Functions

- **Kanarek** (int idZwierzecia, int wiek, [Plec](#) plec, int liczbaNog, string imie, int idWlasciciela, string umaszczenie, string kolor, int [wagaWGramach](#))
Konstruktor argumentowy, ustawiający pola ID zwierzecia, wiek, plec, liczbe nog, imie, ID wlasciciela, umaszczenie, kolor, wage w gramach.
- string [dzwiek](#) ()
Metoda, która zwróci typ string, w którym będzie zawarty dzwiek, który wydaje odpowiedni typ zwierzaka. Np. Dla kanarka cwr cwir.
- string [getWaga](#) ()
Metoda zwracająca sformatowaną wagę, w zależności od wagi zwierzecia. Np. [Kanarek](#) 300g.

Public Member Functions inherited from [Zwierze](#)

- **Zwierze** (int id, int wiek, [Plec](#) plec, int liczbaNog, string imie, int idWlasciciela, string umaszczenie, string kolor, int [wagaWGramach](#))
Konstruktor argumentowy, ustawiający id, wiek, plec, liczbę nog, imię, id właściciela, umaszczenie, kolor, wagę w gramach.
- int [getId](#) ()
Funkcja zwracająca id zwierzęcia.
- int [getWiek](#) ()
Funkcja zwracająca wiek zwierzęcia.
- [Plec](#) [getPlec](#) ()
Zwraca enum plec, określający czy jest to samiec czy samica.
- int [getLiczbaNog](#) ()
Funkcja zwracająca liczbę nog zwierzęcia.
- string [getImie](#) ()
Funkcja zwracająca imię zwierzęcia.
- int [getIdWlasciciela](#) ()
Funkcja zwracająca id właściciela zwierzęcia.
- string [getUmaszczenie](#) ()
Funkcja zwracająca umaszczenie zwierzęcia.
- string [getKolor](#) ()
Funkcja zwracająca kolor zwierzęcia.
- virtual string [dzwiek](#) ()=0
Funkcja wirtualna, zwracająca dźwięk jaki wydaje konkretne zwierzę. Klasy dziedziczące po klasie zwierze muszą zaimplementować metodę [dzwiek\(\)](#). Np. kot ma wydawać dźwięk miau.
- virtual string [getWaga](#) ()=0
Funkcja wirtualna, zwracająca wagę dla konkretnego zwierzęcia. Klasy dziedziczące po klasie zwierze muszą zaimplementować metodę [getWaga\(\)](#). Np. kot ma mieć wyświetlaną wagę w kg, a kanarek w gramach.
- void [setWagaWGramach](#) (int [wagaWGramach](#))
Funkcja ustawiająca wagę, która ma być w gramach.
- void [setImie](#) (string imie)
Funkcja ustawiająca imię.
- string [zamienNaTekst](#) ()
Funkcja zamieniająca obiekt zwierze na tekst, aby można było go zapisać do pliku.

Additional Inherited Members

Protected Attributes inherited from [Zwierze](#)

- int [wagaWGramach](#)
Pole klasy, przechowujące wagę zwierzęcia w gramach.

4.10.1 Detailed Description

Klasa opisująca kanarkę, klasa pochodna. Dziedziczy po klasie [Zwierze](#).

4.10.2 Member Function Documentation

4.10.2.1 dzwiek()

```
string Kanarek::dzwiek ( ) [virtual]
```

Metoda, która zwróci typ string, w którym będzie zawarty dzwiek, który wydaje odpowiedni typ zwierzaka. Np. Dla kanarka cwr cwir.

Implements [Zwierze](#).

4.10.2.2 getWaga()

```
string Kanarek::getWaga ( ) [virtual]
```

Metoda zwracająca sformatowaną wagę, w zależności od wagi zwierzęcia. Np. [Kanarek](#) 300g.

Implements [Zwierze](#).

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

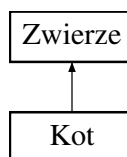
- [Kanarek.h](#)
- Kanarek.cpp

4.11 Kot Class Reference

Klasa opisująca kota, klasa pochodna. Dziedziczy po klasie [Zwierze](#).

```
#include <Kot.h>
```

Inheritance diagram for Kot:



Public Member Functions

- **Kot** (int id, int wiek, [Plec](#) plec, int liczbaNog, string imie, int idWlasciciela, string umaszczenie, string kolor, int [wagaWGramach](#))
Konstruktor argumentowy, ustawiający pola ID zwierzęcia, wiek, plec, liczbę nog, imię, ID właściciela, umaszczenie, kolor, wagę w gramach.
- string [dzwiek](#) ()
Metoda, która zwróci typ string, w którym będzie zawarty dzwiek, który wydaje odpowiedni typ zwierzaka. Np. Dla kota Miua miau.
- string [getWaga](#) ()
Metoda zwracająca sformatowaną wagę, w zależności od wagi zwierzęcia. Np. [Kot](#) 4000g.

Public Member Functions inherited from [Zwierze](#)

- **Zwierze** (int id, int wiek, [Plec](#) plec, int liczbaNog, string imie, int idWlasciciela, string umaszczenie, string kolor, int [wagaWGramach](#))
Konstruktor argumentowy, ustawiający id, wiek, plec, liczbę nog, imię, id właściciela, umaszczenie, kolor, wagę w gramach.
- int [getId](#) ()
Funkcja zwracająca id zwierzęcia.
- int [getWiek](#) ()
Funkcja zwracająca wiek zwierzęcia.
- [Plec](#) [getPlec](#) ()
Zwraca enum plec, określający czy jest to samiec czy samica.
- int [getLiczbaNog](#) ()
Funkcja zwracająca liczbę nog zwierzęcia.
- string [getImie](#) ()
Funkcja zwracająca imię zwierzęcia.
- int [getIdWlasciciela](#) ()
Funkcja zwracająca id właściciela zwierzęcia.
- string [getUmaszczenie](#) ()
Funkcja zwracająca umaszczenie zwierzęcia.
- string [getKolor](#) ()
Funkcja zwracająca kolor zwierzęcia.
- virtual string [dzwiek](#) ()=0
Funkcja wirtualna, zwracająca dźwięk jaki wydaje konkretne zwierzę. Klasy dziedziczące po klasie zwierze muszą zaimplementować metodę [dzwiek\(\)](#). Np. kot ma wydawać dźwięk miau.
- virtual string [getWaga](#) ()=0
Funkcja wirtualna, zwracająca wagę dla konkretnego zwierzęcia. Klasy dziedziczące po klasie zwierze muszą zaimplementować metodę [getWaga\(\)](#). Np. kot ma mieć wyświetlaną wagę w kg, a kanarek w gramach.
- void [setWagaWGramach](#) (int [wagaWGramach](#))
Funkcja ustawiająca wagę, która ma być w gramach.
- void [setImie](#) (string imie)
Funkcja ustawiająca imię.
- string [zamienNaTekst](#) ()
Funkcja zamieniająca obiekt zwierze na tekst, aby można było go zapisać do pliku.

Additional Inherited Members

Protected Attributes inherited from [Zwierze](#)

- int [wagaWGramach](#)
Pole klasy, przechowujące wagę zwierzęcia w gramach.

4.11.1 Detailed Description

Klasa opisująca kota, klasa pochodna. Dziedziczy po klasie [Zwierze](#).

4.11.2 Member Function Documentation

4.11.2.1 dzwiek()

```
string Kot::dzwiek ( ) [virtual]
```

Metoda, która zwróci typ string, w którym będzie zawarty dzwiek, który wydaje odpowiedni typ zwierzaka. Np. Dla kota Miua miau.

Implements [Zwierze](#).

4.11.2.2 getWaga()

```
string Kot::getWaga ( ) [virtual]
```

Metoda zwracająca sformatowaną wagę, w zależności od wagi zwierzęcia. Np. [Kot](#) 4000g.

Implements [Zwierze](#).

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

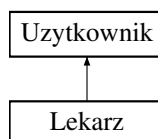
- [Kot.h](#)
- [Kot.cpp](#)

4.12 Lekarz Class Reference

Klasa opisująca lekarza, klasa pochodna. Dziedziczy po klasie [Uzytkownik](#).

```
#include <Lekarz.h>
```

Inheritance diagram for Lekarz:



Public Member Functions

- **Lekarz** (string imie, string nazwisko, string specjalizacja, [Uzytkownik](#) &uzytkownik)
Konstruktor argumentowy, ustawiający pola imie, nazwisko, specjalizację oraz typ użytkownika lekarza.
- string [getImie](#) ()
Funkcja zwracająca imię danego lekarza.
- string [getNazwisko](#) ()
Funkcja zwracająca nazwisko danego lekarza.
- string [getSpecjalizacja](#) ()
Funkcja zwracająca specjalizację danego lekarza.
- [GodzinyPracy zwrocGodzinyPracyDlaDnia](#) ([DniTygodnia](#) dzien)
Funkcja zwracająca godziny pracy lekarza na dany dzień.
- bool [czyDostepnyWGodzinach](#) (int godzina, [DniTygodnia](#) dzienTygodnia)
Funkcja zwracająca czy konkretny lekarz jest dostępny w godzinach.

Public Member Functions inherited from [Uzytkownik](#)

- **Uzytkownik** (string login, string haslo, int id, [TypUzytkownika](#) typUzytkownika)
Konstruktor argumentowy, ustawiajacy login, haslo, typ uzytkownika.
- string [zamienNaTekst](#) ()
Funkcja zamieniajaca obiekt uzytkownika na tekst, aby mozna bylo go zapisac do pliku.
- int [getId](#) ()
Funkcja zwracajaca id uzytkownika.
- [TypUzytkownika](#) [getTypUzytkownika](#) ()
Funkcja zwracajaca typ uzytkownika (lekarz, wlasciciela zwierzecia)

Additional Inherited Members

Static Public Attributes inherited from [Uzytkownik](#)

- static const int [ID_DLA_NIEZNANEGO_UZYTEKOWNIKA](#) = -1
Stala okreslajaca id nieznanego uzytkownika.

4.12.1 Detailed Description

Klasa opisujaca lekarza, klasa pochodna. Dziedziczy po klasie [Uzytkownik](#).

4.12.2 Member Function Documentation

4.12.2.1 czyDostepnyWGodzinach()

```
bool Lekarz::czyDostepnyWGodzinach (
    int godzina,
    DniTygodnia dzienTygodnia )
```

Funkcja zwracajaca czy konkretny lekarz jest dostepny w godzinach.

Parameters

<i>godzina</i>	- godzina, ktora chcemy sprawdzic czy jest dostepny lekarz
<i>dzienTygodnia</i>	- dzien, dla ktorego chcemy sprawdzic dostepnosc lekarza

4.12.2.2 getImie()

```
string Lekarz::getImie ( )
```

Funkcja zwracajaca imie danego lekarza.

Returns

Imie lekarza

4.12.2.3 getNazwisko()

```
string Lekarz::getNazwisko ( )
```

Funkcja zwracająca nazwisko danego lekarza.

Returns

Nazwisko lekarza

4.12.2.4 getSpecjalizacja()

```
string Lekarz::getSpecjalizacja ( )
```

Funkcja zwracająca specjalizację danego lekarza.

Returns

Specjalizacja lekarza

4.12.2.5 zwrocGodzinyPracyDlaDnia()

```
GodzinyPracy Lekarz::zwrocGodzinyPracyDlaDnia (
    DniTygodnia dzien ) [inline]
```

Funkcja zwracająca godziny pracy lekarza na dany dzień.

Returns

dostępność na konkretny dzień

Parameters

<i>dzien</i>	- dzień, dla którego chcemy sprawdzić godziny pracy
--------------	---

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

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

4.13 MenuLekarza Class Reference

Klasa opisująca całe menu lekarza.

```
#include <MenuLekarza.h>
```

Public Member Functions

- **MenuLekarza** (shared_ptr< [SesjaLekarza](#) > sesjaLekarza)
Konstruktor argumentowy, ustawiamy sesje lekarza.
- [OPCJA_POCZATKOWA_LEKARZA](#) **wyswietlMenuPocztkowe** ()
Funkcja zwracajaca menu pocztkowe dla lekarza.
- void **przegladWizyt** ()
Funkcja wyswietlajaca wszystkie wizyty, ktore sa aktualnie dostepne.
- void **anulujWizyte** ()
Funkcja wyswietlajaca wszystkie wizyty, z opcja ich anulowania.

4.13.1 Detailed Description

Klasa opisujaca cale menu lekarza.

4.13.2 Member Function Documentation

4.13.2.1 wyswietlMenuPocztkowe()

[OPCJA_POCZATKOWA_LEKARZA](#) MenuLekarza::wyswietlMenuPocztkowe ()

Funkcja zwracajaca menu pocztkowe dla lekarza.

Returns

zwraca opcje, ktora wybralismy (przeglad wizyt, anuluj wizyte, wyjdz)

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

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

4.14 MenuLogowania Class Reference

Klasa zajmujaca sie obsluzeniem menu logowania.

```
#include <MenuLogowania.h>
```

Public Member Functions

- string [getLogin](#) ()
Funkcja zwracajaca login.
- string [getHaslo](#) ()
Funkcja zwracajaca haslo.
- [MenuLogowaniaStatusu](#) **wyswietlMenuPocztkowe** ()
Funkcja zwracajaca menu pocztkowe dla uzytkownika.
- void **wyswietlMenuLogowania** ()
Funkcja wyswietlajaca menu logowania dla konkretnego typu uzytkownika.

4.14.1 Detailed Description

Klasa zajmujaca sie obsluzeniem menu logowania.

4.14.2 Member Function Documentation

4.14.2.1 getHaslo()

```
string MenuLogowania::getHaslo ( )
```

Funkcja zwracajaca haslo.

Parameters

<i>haslo</i>	uzytkownika
--------------	-------------

4.14.2.2 getLogin()

```
string MenuLogowania::getLogin ( )
```

Funkcja zwracajaca login.

Parameters

<i>login</i>	uzytkownika
--------------	-------------

4.14.2.3 wyswietlMenuPocatkowe()

```
MenuLogowaniaStatusu MenuLogowania::wyswietlMenuPocatkowe ( )
```

Funkcja zwracajaca menu pocatkowe dla uzytkownika.

Returns

zwraca opcje, ktora uzytkownik wybral (logowanie, wyjscie)

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

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

4.15 MenuWlasicielaZwierzecia Class Reference

Klasa opisujaca cale menu wlascicela zwierzecia.

```
#include <MenuWlasicielaZwierzecia.h>
```

Public Member Functions

- **MenuWlasicielaZwierzecia** (shared_ptr< [SesjaWlasicielaZwierzecia](#) > sesja, shared_ptr< [PlanerWizyt](#) > planer, shared_ptr< [SerwisUslug](#) > serwisUslug, shared_ptr< [SerwisLekarza](#) > serwisLekarza, shared_ptr< [SerwisWizyt](#) > serwisWizyt, shared_ptr< [SerwisZwierzat](#) > serwisZwierzat)
Konstruktor argumentowy, ustawiajacy sesje, planner, serwis uslug, serwis lekarza, serwis zwierzat, serwis wizyt.
- **OPCJA_POCZATKOWE_WLASICIELA** [wyswietlMenuPocztkowe](#) ()
Funkcja wyswietlajaca menu pocztkowe dla wlasciciela zwierzecia.
- **OPCJA_PLANOWANIA_WIZYTY** [wyswietlMenuDoPlanowaniaWizyty](#) ()
Funkcja wyswietla menu do planowania wizyty Wyswietlaja sie wszystkie zwierzeta jakie mozemy wybrac.
- **OPCJA_USLUGI** [wswietlMenuUslug](#) ()
Funkcja wyswietlajaca wszystkie uslugi dla danego zwierzaka.
- **OPCJA_TERMINU_WIZYTY** [wyswietlMenuWyboruTerminu](#) ()
Funkcja wyswietlajaca menu wyboru terminu Najpierw wybieramy jeden miesiac z 3/4 podanych, nastepnie dzien, kolejno odpowiedni blok godzinowy z wybranym lekarzem, ktory nas interesuje.
- void **wyswietlAnulowanieWizyty** ()
Funkcja wyswietlajaca wszystkie wizyty, ktore mozemy anulowac.
- void **wyswietlWizyty** ()
Funkcja wyswietlajaca wizyty dla wlasciceila, dla wszystkich jego zwierzat.
- void **dodajZwierzaka** ()
Funkcja wyswietlajaca menu dodania zwierzaka Wybieramy imie, typ, kolor, umaszczenie, wiek, wage (w gramach), plec.
- void **edycjaZwierzat** ()
Funkcja wyswietlajaca menu edycji zwierzaka (jego imie lub wage)

4.15.1 Detailed Description

Klasa opisujaca cale menu wlascicela zwierzecia.

4.15.2 Member Function Documentation

4.15.2.1 wswietlMenuUslug()

```
OPCJA_USLUGI MenuWlasicielaZwierzecia::wswietlMenuUslug ( )
```

Funkcja wyswietlajaca wszystkie uslugi dla danego zwierzaka.

Returns

zwraca wybrana opcje uslugi

4.15.2.2 wyswietlMenuDoPlanowaniaWizyty()

```
OPCJA_PLANOWANIA_WIZYTY MenuWlasicielaZwierzecia::wyswietlMenuDoPlanowaniaWizyty ( )
```

Funkcja wyswietla menu do planowania wizyty Wyswietlaja sie wszystkie zwierzeta jakie mozemy wybrac.

Returns

zwraca wybrana opcje dla planowania wizyty

4.15.2.3 `wyswietlMenuPoczkowe()`

```
OPCJA_POCZATKOWE_WLASICIELA MenuWlasicielaZwierzecia::wyswietlMenuPoczkowe ( )
```

Funkcja wyświetlająca menu poczkowe dla właściciela zwierzecia.

Returns

zwraca 6 dostępnych opcji menu

4.15.2.4 `wyswietlMenuWyboruTerminu()`

```
OPCJA_TERMINU_WIZYTY MenuWlasicielaZwierzecia::wyswietlMenuWyboruTerminu ( )
```

Funkcja wyświetlająca menu wyboru terminu. Najpierw wybieramy jeden miesiąc z 3/4 podanych, następnie dzień, kolejno odpowiedni blok godzinowy z wybranym lekarzem, który nas interesuje.

Returns

zwraca wybrany termin wizyty przez właściciela zwierzaka

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

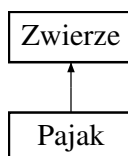
- MenuWlasicielaZwierzecia.h
- MenuWlasicielaZwierzecia.cpp

4.16 Pajak Class Reference

Klasa opisująca pajaka, klasa pochodna. Dziedziczy po klasie [Zwierze](#).

```
#include <Pajak.h>
```

Inheritance diagram for Pajak:



Public Member Functions

- **Pajak** (int idZwierzecia, int wiek, [Plec](#) plec, int liczbaNog, string imie, int idWlasiciela, string umaszczenie, string kolor, int wagaWGramch)
Konstruktor argumentowy, ustawiający pola ID zwierzecia, wiek, plec, liczbe nog, imie, ID właściciela, umaszczenie, kolor, waga w gramach.
- string [dzwiek](#) ()
Metoda, która zwróci typ string, w którym będzie zawarty dzwiek, który wydaje odpowiedni typ zwierzaka. Np. Dla pajaka Skrr skrrr.
- string [getWaga](#) ()
Metoda zwracająca sformatowaną wagę, w zależności od wagi zwierzecia. Np. pajak 200g.

Public Member Functions inherited from [Zwierze](#)

- **Zwierze** (int id, int wiek, [Plec](#) plec, int liczbaNog, string imie, int idWlasciciela, string umaszczenie, string kolor, int [wagaWGramach](#))
Konstruktor argumentowy, ustawiający id, wiek, plec, liczbę nog, imię, id właściciela, umaszczenie, kolor, wagę w gramach.
- int [getId](#) ()
Funkcja zwracająca id zwierzęcia.
- int [getWiek](#) ()
Funkcja zwracająca wiek zwierzęcia.
- [Plec](#) [getPlec](#) ()
Zwraca enum plec, określający czy jest to samiec czy samica.
- int [getLiczbaNog](#) ()
Funkcja zwracająca liczbę nog zwierzęcia.
- string [getImie](#) ()
Funkcja zwracająca imię zwierzęcia.
- int [getIdWlasciciela](#) ()
Funkcja zwracająca id właściciela zwierzęcia.
- string [getUmaszczenie](#) ()
Funkcja zwracająca umaszczenie zwierzęcia.
- string [getKolor](#) ()
Funkcja zwracająca kolor zwierzęcia.
- virtual string [dzwiek](#) ()=0
Funkcja wirtualna, zwracająca dzwiek jaki wydaje konkretne zwierze. Klasy dziedziczące po klasie zwierze muszą zaimplementować metodę [dzwiek\(\)](#). Np. kot ma wydawać dzwiek miau.
- virtual string [getWaga](#) ()=0
Funkcja wirtualna, zwracająca wagę dla konkretnego zwierzęcia. Klasy dziedziczące po klasie zwierze muszą zaimplementować metodę [getWaga\(\)](#). Np. kot ma mieć wyświetlaną wagę w kg, a kanarek w gramach.
- void [setWagaWGramach](#) (int [wagaWGramach](#))
Funkcja ustawiająca wagę, która ma być w gramach.
- void [setImie](#) (string imie)
Funkcja ustawiająca imię.
- string [zamienNaTekst](#) ()
Funkcja zamieniająca obiekt zwierze na tekst, aby można było go zapisać do pliku.

Additional Inherited Members

Protected Attributes inherited from [Zwierze](#)

- int [wagaWGramach](#)
Pole klasy, przechowujące wagę zwierzęcia w gramach.

4.16.1 Detailed Description

Klasa opisująca pajaka, klasa pochodna. Dziedziczy po klasie [Zwierze](#).

4.16.2 Member Function Documentation

4.16.2.1 dzwiek()

```
string Pajak::dzwiek ( ) [virtual]
```

Metoda, która zwróci typ string, w którym będzie zawarty dzwiek, który wydaje odpowiedni typ zwierzaka. Np. Dla pajaka Skrr skrrr.

Implements [Zwierze](#).

4.16.2.2 getWaga()

```
string Pajak::getWaga ( ) [virtual]
```

Metoda zwracająca sformatowaną wagę, w zależności od wagi zwierzęcia. Np. pajak 200g.

Implements [Zwierze](#).

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

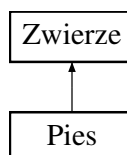
- [Pajak.h](#)
- [Pajak.cpp](#)

4.17 Pies Class Reference

Klasa opisująca psa, klasa pochodna. Dziedziczy po klasie [Zwierze](#).

```
#include <Pies.h>
```

Inheritance diagram for Pies:



Public Member Functions

- **Pies** (int idZwierzecia, int wiek, [Plec](#) plec, int liczbaNog, string imie, int idWlasciciela, string umaszczenie, string kolor, int [wagaWGramach](#))
Konstruktor argumentowy, ustawiający pola ID zwierzęcia, wiek, plec, liczbę nog, imię, ID właściciela, umaszczenie, kolor, wagę w gramach.
- string [dzwiek](#) ()
Metoda, która zwróci typ string, w którym będzie zawarty dzwiek, który wydaje odpowiedni typ zwierzaka. Np. Dla psa Hau hau.
- string [getWaga](#) ()
Metoda zwracająca sformatowaną wagę, w zależności od wagi zwierzęcia. Np. [Pies](#) 6000g.

Public Member Functions inherited from [Zwierze](#)

- **Zwierze** (int id, int wiek, [Plec](#) plec, int liczbaNog, string imie, int idWlasciciela, string umaszczenie, string kolor, int [wagaWGramach](#))
Konstruktor argumentowy, ustawiający id, wiek, plec, liczbę nog, imię, id właściciela, umaszczenie, kolor, wagę w gramach.
- int [getId](#) ()
Funkcja zwracająca id zwierzęcia.
- int [getWiek](#) ()
Funkcja zwracająca wiek zwierzęcia.
- [Plec](#) [getPlec](#) ()
Zwraca enum plec, określający czy jest to samiec czy samica.
- int [getLiczbaNog](#) ()
Funkcja zwracająca liczbę nog zwierzęcia.
- string [getImie](#) ()
Funkcja zwracająca imię zwierzęcia.
- int [getIdWlasciciela](#) ()
Funkcja zwracająca id właściciela zwierzęcia.
- string [getUmaszczenie](#) ()
Funkcja zwracająca umaszczenie zwierzęcia.
- string [getKolor](#) ()
Funkcja zwracająca kolor zwierzęcia.
- virtual string [dzwiek](#) ()=0
Funkcja wirtualna, zwracająca dzwiek jaki wydaje konkretne zwierze. Klasy dziedziczące po klasie zwierze muszą zaimplementować metodę [dzwiek\(\)](#). Np. kot ma wydawać dzwiek miau.
- virtual string [getWaga](#) ()=0
Funkcja wirtualna, zwracająca wagę dla konkretnego zwierzęcia. Klasy dziedziczące po klasie zwierze muszą zaimplementować metodę [getWaga\(\)](#). Np. kot ma mieć wyświetlaną wagę w kg, a kanarek w gramach.
- void [setWagaWGramach](#) (int [wagaWGramach](#))
Funkcja ustawiająca wagę, która ma być w gramach.
- void [setImie](#) (string imie)
Funkcja ustawiająca imię.
- string [zamienNaTekst](#) ()
Funkcja zamieniająca obiekt zwierze na tekst, aby można było go zapisać do pliku.

Additional Inherited Members

Protected Attributes inherited from [Zwierze](#)

- int [wagaWGramach](#)
Pole klasy, przechowujące wagę zwierzęcia w gramach.

4.17.1 Detailed Description

Klasa opisująca psa, klasa pochodna. Dziedziczy po klasie [Zwierze](#).

4.17.2 Member Function Documentation

4.17.2.1 dzwiek()

```
string Pies::dzwiek ( ) [virtual]
```

Metoda, która zwróci typ string, w którym będzie zawarty dzwiek, który wydaje odpowiedni typ zwierzaka. Np. Dla psa Hau hau.

Implements [Zwierze](#).

4.17.2.2 getWaga()

```
string Pies::getWaga ( ) [virtual]
```

Metoda zwracająca sformatowaną wagę, w zależności od wagi zwierzęcia. Np. [Pies](#) 6000g.

Implements [Zwierze](#).

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

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

4.18 PlanerWizyt Class Reference

Klasa odpowiedzialna za planowanie wizyt oraz wszelkie manipulacje data i czasem.

```
#include <PlanerWizyt.h>
```

Public Member Functions

- **PlanerWizyt** (shared_ptr< [SerwisGrafikuPracy](#) > serwisGrafikuPracy, shared_ptr< [SerwisWizyt](#) > serwis↔Wizyt)
Konstruktor argumentowy, ustawiający serwis grafiku pracy i serwis wizyt.
- string [załadujAktualnyDzienTygodnia](#) (time_t currentTime)
Funkcja zwracająca aktualny dzień tygodnia z przekazanego czasu.
- string [załadujMiesiac](#) (time_t currentTime)
Funkcja zwracająca miesiąc z przekazanego czasu.
- string [załadujDzienMiesiaca](#) (time_t currentTime)
Funkcja zwracająca dzień miesiąca z przekazanego czasu.
- string [załadujRok](#) (time_t czas)
Funkcja zwracająca rok z przekazanego czasu.
- pair< [DniTygodnia](#), int > [załadujDostępnaGodzinaNaDzienTygodnia](#) ([DniTygodnia](#) aktualnyDzienTygodnia, [DniTygodnia](#) terminNaNastepnyDzien, int aktualnaGodzina, int pierwszaDostępnaGodzinaWizyty)
Funkcja zwracająca pierwszą dostępną godzinę na aktualny dzień tygodnia i za daną godzinę. Jest to funkcja pomocnicza używana w metodzie [znajdzPierwszyDostępnyTermin\(\)](#)
- pair< [DniTygodnia](#), int > [znajdzPierwszyDostępnyTermin](#) ()
Funkcja zwracająca pierwszy dostępny termin. Używa ona różnych wariacji metod [załadujDostępnaGodzinaNaDzienTygodnia\(\)](#)

- `pair< string, time_t > zaladujPierszaDostepnaDate ()`
Funkcja zwracajaca pierwszy dostepna date bazujac na aktualnym czasie.
- `pair< string, time_t > zaladujOstatniaDostepnaDate (time_t poczaktowyCzas, string format)`
Funkcja zwracajaca ostatnia mozliwa date do rezerwacji wizyty.
- `time_t dodajDzien (time_t poczaktowyCzas, int liczbaDni)`
Funkcja dodajaca do zadanego czasu dana liczne dni. Np. time_t czas = 01-01-2000 int liczba_dni = 10 dodaj↔ Dzien(czas, liczba_dni) -> 11-01-2000.
- `time_t dodajGodzine (time_t poczaktowyCzas, int liczbaGodzin)`
Funkcja dodajaca do zadanego czasu dana liczne godzin. Np. time_t czas = 01-01-2000 10:00 int liczba_godzin = 10 dodajGodzine(czas, liczba_godzin) -> 01-01-2000 20:00.
- `time_t ustawCzas (time_t poczaktowyCzas, int godzina, int minuta)`
Funkcja ustawiajaca godzine dla danego czasu Np. time_t poczaktowyCzas = 01-01-2000 10:00 int godzina = 8 int minuta = 30 dodajGodzine(poczaktowyCzas, liczba_godzin) -> 01-01-2000 8:30.
- `time_t dodajGodzinePlusZaokraglenie (time_t poczaktowyCzas, int liczbaGodzin)`
Funkcja zaokraglajaca godzine Np. time_t poczaktowyCzas = 01-01-2000 10:00 int liczbaGodzin = 1 dodajGodzine↔ PlusZaokraglenie(poczaktowyCzas, liczbaGodzin) -> 01-01-2000 11:00.
- `DniTygodnia zamienNaDniTygonida (string dniTygodniaTekst)`
Funkcja zamieniajaca angielski dzien tygodnia na tym enum zawierajacy polskie dnia tygodnia Np. string dniTygodnia↔ Tekst = "Monday" zamienNaDniTygonida(dniTygodniaTekst) -> DniTygodnia::PONIEDZIALEK.
- `DniTygodnia zaladujNastepnyDzienTygodnia (DniTygodnia aktualnyDzienTygodnia)`
Funkcja zwracajaca nastepny dostepny dzien tygodnia Np. zaladujNastepnyDzienTygodnia([DniTygodnia](#)↔ ::PONIEDZIALEK) -> [DniTygodnia](#)::WTOREK zaladujNastepnyDzienTygodnia([DniTygodnia](#)::NIEDZIELA) -> [DniTygodnia](#)::PONIEDZIALEK zaladujNastepnyDzienTygodnia([DniTygodnia](#)::SOBOTA) -> [DniTygodnia](#)::↔ PONIEDZIALEK - w niedziele jest zamkniete.
- `std::pair< DniTygodnia, time_t > zaladujDostepyCzas (time_t aktualnyCzas)`
Funkcja zwracajaca pierwszy dostepny czas - zwraca ona pare, w ktorej jest zawarty dzien tygodnia i czas.
- `void wyswietl (time_t currentTime, string format)`
Funkcja wyswietlajaca dany czas o zadanym formacie.
- `DniTygodnia zamienNaDniTygonida (time_t czas)`
Funkcja zamieniajaca dany czas na tym enum [DniTygodnia](#).
- `time_t getGodzinaZamkneciaWDniu (DniTygodnia ostatniDostepnyDzienTygodnia, time_t czas)`
Funkcja godzine zamknecia w danym dniu.
- `unordered_map< int, time_t > wygenerujMiesiace (time_t czasRozpoczecia, time_t czasZakonczenia)`
Funkcja generujaca wszystkie dostepne miesiace, w ktorych mozna zarezerwowac wizyte.
- `time_t dodajMiesiac (time_t poczaktowyCzas, int liczbaMiesiecy)`
Funkcja dodajaca do zadanego czasu dana liczne miesiecy. Np. time_t czas = 01-01-2000 int liczbaMiesiecy = 3 dodajMiesiac(czas, liczbaMiesiecy) -> 01-04-2000.
- `int getMiesiac (time_t czas)`
Funkcja zwracajaca liczbe okreslajaca indeks miesiaca dla danego czasu Np. time_t czas = 01-04-2000.
- `string zamienNaMiesiac (time_t czas)`
Funkcja zamieniajaca czas na dany miesiac po polsku Np. time_t czas = 01-01-2000 zamienNaMiesiac(czas) -> "Styczen".
- `std::string zamienZAngielskiegoNaPolski (const string &miesiacPoAngielsku)`
Funkcja zamieniajaca miesiac z anigelskiego na miesiac po polsku Np. string miesiacPoAngielsku = "January" zamienZAngielskiegoNaPolski(miesiacPoAngielsku) -> "Styczen".
- `void wypiszKalendarz (int rok, int miesiac, int dzienPocztkowy, int dzienKoncowy)`
Funkcja wyswietlajaca kalenarz dla danego miesiaca.
- `bool czyPierwszDostepnyMiesiac (unordered_map< int, time_t > miesiace, time_t miesiac)`
Funkcja zwracajaca czy zadany miesiac jest pierwszym dostepny miesiacem.
- `bool czyOstatniDostepnyMiesiac (unordered_map< int, time_t > miesiace, time_t miesiac)`
Funkcja zwracajaca czy zadany miesiac jest ostatnim dostepny miesiacem.
- `int ostatniDzienMiesiaca (time_t czasWMillisekundach)`
Funkcja ostatni dzien miesiaca.

- `unordered_map< int, time_t >` `wygenerujDni` (`int rok`, `int miesiac`, `int dzienPoczkowy`, `int dzienKoncowy`)
Funkcja generujaca dni na zadany miesiac uwzgledniajac rok oraz dzien poczkowy i koncowy.
- `bool` `czyJestDniem` (`DniTygodnia` `wybranyDzien`, `time_t` `wybranyCzas`)
Funkcja sprawdzajaca czy podany dzien tygodnia jest dniem tygodnia z czasu.
- `unordered_map< int, unordered_map< int, shared_ptr< Lekarz > > >` `wygenerujBlokiGodzinowe` (`time_t` `wybranyDzien`, `list< shared_ptr< Lekarz > >` `lekarze`)
Funkcja generujaca bloki godzinowe.
- `void` `utworzWizyte` (`Wizyta` `wizyta`)
Funkcja tworzaca wizyte i zapuszczajaca ja do bazy.
- `int` `wygenerujIdWizyty` ()
Funkcja generujaca kolejne Id dla wizyty.

4.18.1 Detailed Description

Klasa odpowiedzialna za planowanie wizyt oraz wszelkie manipulacje data i czasem.

4.18.2 Member Function Documentation

4.18.2.1 czyJestDniem()

```
bool PlanerWizyt::czyJestDniem (
    DniTygodnia wybranyDzien,
    time_t wybranyCzas )
```

Funkcja sprawdzajaca czy podany dzien tygodnia jest dniem tygodnia z czasu.

```
time_t czas = 01-01-2023 // ten czas to niedziela czyJestDniem(DniTygodnia::PONIEDZIALEK, czas) -> false czyJestDniem(DniTygodnia::NIEDZIELA, czas) -> true
```

Parameters

<code>wybranyDzien</code>	- wybrany dzien tygodnia np. <code>DniTygodnia::PONIEDZIALEK</code>
<code>wybranyCzas</code>	- wybrany czas, ktory chcemy sprawdzic czy jest podanym dniem tygodnia (pierwszy paramter)

Returns

`true` jest wybrany dzien tygodnia jest dniem tygodnia z podaego czasu

4.18.2.2 czyOstatniDostepnyMiesiac()

```
bool PlanerWizyt::czyOstatniDostepnyMiesiac (
    unordered_map< int, time_t > miesiace,
    time_t miesiac )
```

Funkcja zwracajaca czy zadany miesiac jest ostatnim dostepny miesiacem.

Parameters

<i>miesiace</i>	- wygenerowana mapa wszystkich dostępnych miesiecy
<i>miesiac</i>	- miesiac który chcemy sprawdzic czy jest ostatnim dostem

Returns

- true jesli zadany miesiac jest ostatnim dostepnym miesiacem, false jesli nim nie jest

4.18.2.3 czyPierwszDostepnyMiesiac()

```
bool PlanerWizyt::czyPierwszDostepnyMiesiac (
    unordered_map< int, time_t > miesiace,
    time_t miesiac )
```

Funkcja zwracajaca czy zadany miesiac jest pierwszym dostepny miesiacem.

Parameters

<i>miesiace</i>	- wygenerowana mapa wszystkich dostępnych miesiecy
<i>miesiac</i>	- miesiac który chcemy sprawdzic czy jest pierwsztn dostem

Returns

- true jesli zadany miesiac jest pierwszym dostepnym miesiacem, false jesli nim nie jest

4.18.2.4 dodajDzien()

```
time_t PlanerWizyt::dodajDzien (
    time_t poczaktowyCzas,
    int liczbaDni )
```

Funkcja dodajaca do zadanego czasu dana liczne dni. Np. time_t czas = 01-01-2000 int liczba_dni = 10 dodaj↔
Dzien(czas, liczba_dni) -> 11-01-2000.

Returns

zmieniony czas o dodanej liczbnie dni

Parameters

<i>poczaktowyCzas</i>	- czas do ktorego chcemy dodac zadana liczbe dni
<i>liczbaDni</i>	- liczba dni, ktora zostanie dodana do istniejacego czasu

4.18.2.5 dodajGodzine()

```
time_t PlanerWizyt::dodajGodzine (
    time_t poczaktowyCzas,
    int liczbaGodzin )
```

Funkcja dodająca do zadanego czasu dana licznę godzin. Np. time_t czas = 01-01-2000 10:00 int liczba_godzin = 10 dodajGodzine(czas, liczba_godzin) -> 01-01-2000 20:00.

Returns

zmieniony czas o dodanej liczbie godzin

Parameters

<i>poczaktowyCzas</i>	- czas do ktorego chcemy dodac zadana liczbę godzin
<i>liczbaGodzin</i>	- liczba godzin, która zostanie dodana do istniejącego czasu

4.18.2.6 dodajGodzinePlusZaokraglenie()

```
time_t PlanerWizyt::dodajGodzinePlusZaokraglenie (
    time_t poczaktowyCzas,
    int liczbaGodzin )
```

Funkcja zaokrąglająca godzinę. Np. time_t poczaktowyCzas = 01-01-2000 10:00 int liczbaGodzin = 1 dodajGodzinePlusZaokraglenie(poczaktowyCzas, liczbaGodzin) -> 01-01-2000 11:00.

Np. time_t poczaktowyCzas = 01-01-2000 10:01 int liczbaGodzin = 1 dodajGodzinePlusZaokraglenie(poczaktowyCzas, liczbaGodzin) -> 01-01-2000 12:00

Returns

zmieniony czas z zaokrągloną godziną

Parameters

<i>poczaktowyCzas</i>	- czas dla ktorego chcemy zaokrąglić godzinę
<i>liczbaGodzin</i>	- liczba godzin dla której chcemy zaokrąglić w górę

4.18.2.7 dodajMiesiac()

```
time_t PlanerWizyt::dodajMiesiac (
    time_t poczaktowyCzas,
    int liczbaMiesiecy )
```

Funkcja dodająca do zadanego czasu daną licznę miesięcy. Np. time_t czas = 01-01-2000 int liczbaMiesiecy = 3 dodajMiesiac(czas, liczbaMiesiecy) -> 01-04-2000.

Returns

zmieniony czas o dodanej liczbie miesiecy

Parameters

<i>poczaktowyCzas</i>	- czas do ktorego chcemy dodac zadana liczbe miesiecy
<i>liczbaMiesiecy</i>	- liczba miesiecy, ktora zostanie dodana do istniejacego czasu

4.18.2.8 getGodzinaZamknieniaWDniu()

```
time_t PlanerWizyt::getGodzinaZamknieniaWDniu (
    DniTygodnia ostatniDostepnyDzienTygodnia,
    time_t czas )
```

Funkcja godzinie zamknienia w danym dniu.

Returns

czas w ktorym jest zamknienie w danym dniu

Parameters

<i>czas</i>	- parametr czasu, dla ktorego sprawdzic jaka jest godzina zamknienia
<i>ostatniDostepnyDzienTygodnia</i>	- parametr dnia tygodnia, dla ktorego chcemy sprawdzic godzinie zamknienia

4.18.2.9 getMiesiac()

```
int PlanerWizyt::getMiesiac (
    time_t czas )
```

Funkcja zwracajaca liczbe okreslajaca indeks miesiaca dla danego czasu Np. time_t czas = 01-04-2000.

getMiesiac(czas) -> 3 - indeskcja jest z przedzialu [0..11]

Returns

indeks miesiaca - przedzial 0-11

Parameters

<i>czas</i>	- czas, z ktorego chcemy wycagnac miesiac
-------------	---

4.18.2.10 ostatniDzienMiesiaca()

```
int PlanerWizyt::ostatniDzienMiesiaca (
    time_t czasWMillisekundach )
```

Funkcja ostatni dzien miesiaca.

Parameters

<i>czasWMillisekundach</i>	- czas dla ktorego chcemy sprawdzic jaki jest ostatni dzien miesiaca np. 31,30,29,28
----------------------------	--

Returns

ostatni dzien miesiaca. Mozliwe liczby to: 31,30,29,28

4.18.2.11 ustawCzas()

```
time_t PlanerWizyt::ustawCzas (
    time_t poczaktowyCzas,
    int godzina,
    int minuta )
```

Funkcja ustawiajaca godzine dla danego czasu Np. time_t poczaktowyCzas = 01-01-2000 10:00 int godzina = 8 int minuta = 30 dodajGodzine(poczaktowyCzas, liczba_godzin) -> 01-01-2000 8:30.

Returns

ustawCzas zwraca zmieniony czas z ustawiona zadana godzina

Parameters

<i>poczaktowyCzas</i>	- czas dla ktorego chcemy ustawic zadana godzine i minute
<i>godzina</i>	- godzina, ktoa chcemy ustawic
<i>minuta</i>	- minuta, ktora chcemy ustawic

4.18.2.12 utworzWizyte()

```
void PlanerWizyt::utworzWizyte (
    Wizyta wizyta )
```

Funkcja tworzaca wizyte i zapusjaca ja do bazy,.

Parameters

<i>wizyta</i>	- obiekt wizyty, ktory chcemy zapisac do bazy
---------------	---

4.18.2.13 wygenerujBlokGodzinowe()

```
unordered_map< int, unordered_map< int, shared_ptr< Lekarz > > > PlanerWizyt::wygeneruj(
    BlokiGodzinowe (
        time_t wybranyDzien,
        list< shared_ptr< Lekarz > > lekarze )
```

Funkcja generująca bloki godzinowe.

Parameters

<i>wybranyDzien</i>	- dzien dla ktorego chcemy wygenerowac dostepne bloki godzinowe
<i>lekarze</i>	- dosepni lekarze w danym dniu

Returns

mapa zawieraca bloki godzinowe bedace kluczem ten mapy. Watoscia tej mapy jest mapa zawierajaca int bedacy id lekarza a wartoscia jest obiekt lekarza

4.18.2.14 wygenerujDni()

```
unordered_map< int, time_t > PlanerWizyt::wygenerujDni (
    int rok,
    int miesiac,
    int dzienPoczątkowy,
    int dzienKoncowy )
```

Funkcja generująca dni na zadany miesiac uwzgledniajac rok oraz dzien początkowy i koncowy.

Parameters

<i>rok</i>	- rok potrzebny to wygenerowania dni na caly miesiac
<i>miesiac</i>	- miesiac dla ktorego chcemy wygenerowac dni
<i>dzienPoczątkowy</i>	- pierwszy dzien miesiaca, od ktorego chcemy zaczac generowac mape z dniami
<i>dzienKoncowy</i>	- ostatni dzien miesiaca, do ktorego chcemy zaczac generowac mape z dniami

Returns

mapa z wygenerowanymi dniami miesiaca, kluczem tej mapy jest dzien miesiaca np. 1 jest pierwszym dniem, a wartoscia jest czas dla danego dnia

4.18.2.15 wygenerujIdWizyty()

```
int PlanerWizyt::wygenerujIdWizyty ( )
```

Funkcja generująca kolejne Id dla wizyty.

Returns

id kolejnej wizyty, ktora mozemy zapisac w bazie

4.18.2.16 wygenerujMiesiace()

```
unordered_map< int, time_t > PlanerWizyt::wygenerujMiesiace (
    time_t czasRozpoczecia,
    time_t czasZakonczenia )
```

Funkcja generująca wszystkie dostępne miesiące, w których można zarezerwować wizytę.

Returns

mapa, która zawiera miesiąc i czas - oznaczają one dostępne miesiące do rezerwacji wizyty

Parameters

<i>czasRozpoczecia</i>	- parametr czasu, od którego będziemy generować pierwszy dostępny miesiąc
<i>czasZakonczenia</i>	- parametr czasu, do którego będziemy generować ostatni dostępny miesiąc

4.18.2.17 wypiszKalendarz()

```
void PlanerWizyt::wypiszKalendarz (
    int rok,
    int miesiac,
    int dzienPoczątkowy,
    int dzienKoncowy )
```

Funkcja wyświetlająca kalendarz dla danego miesiąca.

Parameters

<i>rok</i>	- rok, który ma być uwzględniony przy wyświetlaniu kalendarza
<i>miesiac</i>	- miesiąc dla którego rysujemy kalendarz
<i>dzienPoczątkowy</i>	- dzień początkowy - dni przed tym dniem w ramach tego samego miesiąca nie będą wyświetlane
<i>dzienKoncowy</i>	- dzień końcowy - dni po tym dniu w ramach tego samego miesiąca nie będą wyświetlane

4.18.2.18 wyswietl()

```
void PlanerWizyt::wyswietl (
    time_t currentTime,
    string format )
```

Funkcja wyświetlająca dany czas o zadanym formacie.

Parameters

<i>currentTime</i>	- czas, który chcemy odpowiednio sformatować
<i>format</i>	- zadany format czasu

4.18.2.19 `zaladujAktualnyDzienTygodnia()`

```
std::string PlanerWizyt::zaladujAktualnyDzienTygodnia (
    time_t currentTime )
```

Funkcja zwracająca aktualny dzień tygodnia z przekazanego czasu.

Returns

Aktualny dzień tygodnia (Poniedziałek - Niedziela) po angielsku

Parameters

<i>currentTime</i>	- zmienna opisująca czas, dla którego ma być wyciągnięty dzień tygodnia
--------------------	---

4.18.2.20 `zaladujDostepnaGodzineNaDzienTygodnia()`

```
std::pair< DniTygodnia, int > PlanerWizyt::zaladujDostepnaGodzineNaDzienTygodnia (
    DniTygodnia aktualnyDzienTygodnia,
    DniTygodnia terminNaNastepnyDzien,
    int aktualnaGodzina,
    int pierwszaDostepnaGodzinaWizyty )
```

Funkcja zwracająca pierwszą dostępną godzinę na aktualny dzień tygodnia i za daną godzinę. Jest to funkcja pomocnicza używana w metodzie [znajdzPierwszyDostepnyTermin\(\)](#)

Returns

Zwraca parę, której pierwszym elementem jest dzień tygodnia, a drugim godzina - reprezentują one pierwszy możliwy termin

Parameters

<i>aktualnyDzienTygodnia</i>	- parametr opisujący aktualny dzień tygodnia, dla którego chcemy sprawdzić dostępność
<i>terminNaNastepnyDzien</i>	- parametr opisujący następny dzień tygodnia z dostępną wizytą po aktualnym dniu tygodnia
<i>aktualnaGodzina</i>	- parametr opisujący aktualnie rozważaną godzinę wizyty
<i>pierwszaDostepnaGodzinaWizyty</i>	- parametr opisujący pierwszą dostępną godzinę

4.18.2.21 `zaladujDostepyCzas()`

```
std::pair< DniTygodnia, time_t > PlanerWizyt::zaladujDostepyCzas (
    time_t aktualnyCzas )
```

Funkcja zwracająca pierwszy dostępny czas - zwraca ona parę, w której jest zawarty dzień tygodnia i czas.

Returns

zwraca pierwszy dostępny czas dla wizyty - typem zwracanym jest para, gdzie pierwszym elementem jest pierwszy dostępny dzień tygodnia oraz czas

Parameters

<i>aktualnyCzas</i>	- zadany czas dla ktorego chcemy znalezc pierwszy dostepny dzien tygodnia z czasem (typ <code>time_t</code>)
---------------------	---

4.18.2.22 `zaladujDzienMiesiaca()`

```
std::string PlanerWizyt::zaladujDzienMiesiaca (
    time_t currentTime )
```

Funkcja zwracajaca dzien miesiaca z przekazanego czasu.

Returns

Dzien miesiaca (1 - 31)/(1-30)/(1-28/29)

Parameters

<i>currentTime</i>	- zmienna opisujaca czas, dla ktorego ma byc wyciagniety dzien miesiaca
--------------------	---

4.18.2.23 `zaladujMiesiac()`

```
std::string PlanerWizyt::zaladujMiesiac (
    time_t currentTime )
```

Funkcja zwracajaca miesiac z przekazanego czasu.

Returns

Miesiac (Styczen - Grudzien) po angelsku

Parameters

<i>currentTime</i>	- zmienna opisujaca czas, dla ktorego ma byc wyciagniety miesiac
--------------------	--

4.18.2.24 `zaladujNastepnyDzienTygodnia()`

```
DniTygodnia PlanerWizyt::zaladujNastepnyDzienTygodnia (
    DniTygodnia aktualnyDzienTygodnia )
```

Funkcja zwracajaca nastepny dostepny dzien tygodnia Np. `zaladujNastepnyDzienTygodnia(DniTygodnia::PONIEDZIALEK)` -> `DniTygodnia::WTOREK` `zaladujNastepnyDzienTygodnia(DniTygodnia::NIEDZIELA)` -> `DniTygodnia::PONIEDZIALEK` `zaladujNastepnyDzienTygodnia(DniTygodnia::SOBOTA)` -> `DniTygodnia::PONIEDZIALEK` - w niedziele jest zamkniete.

Returns

nastepny dostepny dzien tygodnia, na ktory mozna zarezerwować wizyte

Parameters

<i>aktualnyDzienTygodnia</i>	- zadany dzien tygodnia, dla ktorego szukamy nastepnego dnia ktory bedzie dostepny
------------------------------	--

4.18.2.25 `zaladujOstatniaDostepnaDate()`

```
std::pair< string, time_t > PlanerWizyt::zaladujOstatniaDostepnaDate (
    time_t poczaktowyCzas,
    string format )
```

Funkcja zwracajaca ostatnia mozliwa date do rezerwacji wizyty.

Returns

Zwraca pare, ktorej pierwszym elementem jest sformatowana data, a drugim czas (typ `time_t`) - reprezentuja one ostatni mozliwy termin

Parameters

<i>poczaktowyCzas</i>	- parametr typu <code>time_t</code> , w ktorym jest zapisany czas poczatkowy
<i>format</i>	- format wyjsciowy ostatniej dostepnej daty

4.18.2.26 `zaladujPierszaDostepnaDate()`

```
pair< string, time_t > PlanerWizyt::zaladujPierszaDostepnaDate ( )
```

Funkcja zwracajaca pierwszy dostepna date bazujac na aktualnym czasie.

Returns

Zwraca pare, ktorej pierwszym elementem jest dzien tygodnia, a drugim godzina - reprezentuja one pierwszy mozliwy termin

4.18.2.27 `zaladujRok()`

```
std::string PlanerWizyt::zaladujRok (
    time_t czas )
```

Funkcja zwracajaca rok z przekazanego czasu.

Returns

Rok

Parameters

<i>czas</i>	- zmienna opisujaca czas, dla ktorego ma byc wyciagniety rok
-------------	--

4.18.2.28 zamienNaDniTygonida() [1/2]

```
DniTygodnia PlanerWizyt::zamienNaDniTygonida (
    string dniTygodniaTekst )
```

Funkcja zamienia angielski dzień tygodnia na tym enum zawierający polskie dni tygodnia. Np. string dniTygodniaTekst = "Monday" zamienNaDniTygonida(dniTygodniaTekst) -> DniTygodnia::PONIEDZIALEK.

Returns

dzień tygodnia, będący polskim odpowiednikiem

Parameters

<i>dniTygodniaTekst</i>	- dzień tygodnia po angielski
-------------------------	-------------------------------

4.18.2.29 zamienNaDniTygonida() [2/2]

```
DniTygodnia PlanerWizyt::zamienNaDniTygonida (
    time_t czas )
```

Funkcja zamienia dany czas na tym enum DniTygodnia.

Returns

dzień tygodnia odpowiadający zadanemu czasowi

Parameters

<i>czas</i>	- parametr czasu, dla którego sprawdzić jakim jest dniem tygodnia
-------------	---

4.18.2.30 zamienNaMiesiac()

```
string PlanerWizyt::zamienNaMiesiac (
    time_t czas )
```

Funkcja zamienia czas na dany miesiąc po polsku. Np. time_t czas = 01-01-2000 zamienNaMiesiac(czas) -> "Styczen".

Returns

miesiąc po polsku

Parameters

<i>czas</i>	- czas, dla którego chcemy zwrócić miesiąc po polsku
-------------	--

4.18.2.31 zamienZAngielskiegoNaPolski()

```
std::string PlanerWizyt::zamienZAngielskiegoNaPolski (
    const string & miesiacPoAngielsku )
```

Funkcja zamienia miesiąc z angielskiego na miesiąc po polsku. Np. `string miesiacPoAngielsku = "January"`
`zamienZAngielskiegoNaPolski(miesiacPoAngielsku) -> "Styczen"`.

Returns

miesiąc po polsku

Parameters

<i>miesiacPoAngielsku</i>	- miesiąc w języku angielskim
---------------------------	-------------------------------

4.18.2.32 znajdzPierwszyDostepnyTermin()

```
std::pair< DniTygodnia, int > PlanerWizyt::znajdzPierwszyDostepnyTermin ( )
```

Funkcja zwracająca pierwszy dostępny termin. Używa ona różnych wariacji metod [załadujDostepnaGodzineNaDzienTygodnia\(\)](#)

Returns

Zwraca parę, której pierwszym elementem jest dzień tygodnia, a drugim godzina - reprezentują one pierwszy możliwy termin

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

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

4.19 SerwisGrafikuPracy Class Reference

Klasa odpowiedzialna za serwis grafiku pracy.

```
#include <SerwisGrafikuPracy.h>
```

Public Member Functions

- **SerwisGrafikuPracy** (shared_ptr< [CzytaczGrafikuPracy](#) > czytaczGrafikuPracy)
Konstruktor argumentowy, ustawiający czytacz grafiku pracy.
- unordered_map< [DniTygodnia](#), [GodzinyPracy](#) > [załadujGrafik](#) (shared_ptr< [Lekarz](#) > lekarz)
Funkcja zwracająca grafik pracy dla podanego w argumencie lekarza.

4.19.1 Detailed Description

Klasa odpowiedzialna za serwis grafiku pracy.

4.19.2 Member Function Documentation

4.19.2.1 załadujGrafik()

```
unordered_map< DniTygodnia, GodzinyPracy > SerwisGrafikuPracy::załadujGrafik (
    shared_ptr< Lekarz > lekarz )
```

Funkcja zwracająca grafik pracy dla podanego w argumencie lekarza.

Parameters

<i>lekarz</i>	- lekarz, dla ktorego chcemy znalezc grafik pracy
---------------	---

Returns

zwraca mape zawierająca dni tygodnia wraz z godzinami dostenosci lekarza

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

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

4.20 SerwisLekarza Class Reference

Klasa odpowiedzialna za serwis lekarza.

```
#include <SerwisLekarza.h>
```

Public Member Functions

- **SerwisLekarza** (shared_ptr< [CzytaczLekarza](#) > czytaczLekarza, shared_ptr< [SerwisLogowaniaUzytkownika](#) > czytaczUzytkownikow)
Konstruktor argumentowy, ustawiający czytacza lekarza, czytacza uzytkownikow.
- shared_ptr< [Lekarz](#) > [załadujLekarza](#) (shared_ptr< [Uzytkownik](#) > uzytkownik)
Funkcja zwracająca lekarza, na podstawie przekazanego uzytkownika. Szukamy lekarza po id uzytkownika.
- list< shared_ptr< [Lekarz](#) > > [załadujWszystkichLekarzy](#) ()
Funkcja zwracająca wszystkich lekarzy.
- shared_ptr< [Lekarz](#) > [znajdzLekarzaPold](#) (int idLekarza, list< shared_ptr< [Lekarz](#) > > lekarze)
Funkcja szukająca lekarza po przekazanym id w podanej liscie lekarzy.

4.20.1 Detailed Description

Klasa odpowiedzialna za serwis lekarza.

4.20.2 Member Function Documentation

4.20.2.1 załadujLekarza()

```
shared_ptr< Lekarz > SerwisLekarza::załadujLekarza (
    shared_ptr< Uzytkownik > uzytkownik )
```

Funkcja zwracająca lekarza, na podstawie przekazanego użytkownika. Szukamy lekarza po id użytkownika.

Parameters

<i>uzytkownik</i>	- wskaznik na obiekt uzytkownik
-------------------	---------------------------------

Returns

zwraca wskaznik na lekarza, o ile zostal znaleziony

4.20.2.2 załadujWszystkichLekarzy()

```
list< shared_ptr< Lekarz > > SerwisLekarza::załadujWszystkichLekarzy ( )
```

Funkcja zwracająca wszystkich lekarzy.

Returns

zwracający typ jest lista wskazników na lekarzy, jeżeli go nie znajdzie to zwraca NULL

4.20.2.3 znajdzLekarzaPoId()

```
shared_ptr< Lekarz > SerwisLekarza::znajdzLekarzaPoId (
    int idLekarza,
    list< shared_ptr< Lekarz > > lekarze )
```

Funkcja szukająca lekarza po przekazanym id w podanej liście lekarzy.

Parameters

<i>idLekarza</i>	- id szukanego lekarza
<i>lekarze</i>	- lista lekarzy, wśród których chcemy znaleźć o podanym id

Returns

zwraca szukanego lekarza, jeżeli go nie znajdzie to zwraca NULL

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

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

4.21 SerwisLogowaniaUzytkownika Class Reference

Klasa odpowiedzialna za serwis logowania uzytkownikow.

```
#include <SerwisLogowaniaUzytkownika.h>
```

Public Member Functions

- **SerwisLogowaniaUzytkownika** (shared_ptr< [CzytaczUzytkownikow](#) > czytaczUzytkownikow)
Konstruktor argumentowy, ustawiajacy czytacza uzytkownikow.
- shared_ptr< [Uzytkownik](#) > [zaloguj](#) (string login, string haslo)
Funkcja zwracajaca uzytkownikow, na podstawie przekazanego loginu i hasla.
- shared_ptr< [Uzytkownik](#) > [znajdz](#) (string id)
Funkcja zwracajaca uzytkownika.

4.21.1 Detailed Description

Klasa odpowiedzialna za serwis logowania uzytkownikow.

4.21.2 Member Function Documentation

4.21.2.1 zaloguj()

```
shared_ptr< Uzytkownik > SerwisLogowaniaUzytkownika::zaloguj (  
    string login,  
    string haslo )
```

Funkcja zwracajaca uzytkownikow, na podstawie przekazanego loginu i hasla.

Parameters

<i>login</i>	- login uzytkownika
<i>haslo</i>	- haslo uzytkownika

Returns

zwraca uzytkownika

4.21.2.2 znajdz()

```
shared_ptr< Uzytkownik > SerwisLogowaniaUzytkownika::znajdz (  
    string id )
```

Funkcja zwracajaca uzytkownika.

Parameters

<i>id</i>	- id uzytkownika, ktorego szukamy
-----------	-----------------------------------

Returns

zwraca id

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

- SerwisLogowaniaUzytkownika.h
- SerwisLogowaniaUzytkownika.cpp

4.22 SerwisUslug Class Reference

Klasa odpowiedzialna za serwis uslug.

```
#include <SerwisUslug.h>
```

Public Member Functions

- **SerwisUslug ()**
Konstruktor bezargumentowy.
- void **zainicjalizujUslugi ()**
Funkcja zwracajaca zainicjalizowane uslugi dla zwierzat.
- shared_ptr< [Usługa](#) > **znajdzPoTypie** (string typ)
Funkcja szukajaca uslugi po przekazanym typie w podanej liscie uslug.
- vector< shared_ptr< [Usługa](#) > > **stworzUslugeDlaZwierzaka** (shared_ptr< [Zwierze](#) > zwierze)
Funkcja tworzaca usluge dla konkretnego juz zwierzaka.

4.22.1 Detailed Description

Klasa odpowiedzialna za serwis uslug.

4.22.2 Member Function Documentation

4.22.2.1 stworzUslugeDlaZwierzaka()

```
vector< shared_ptr< Usługa > > SerwisUslug::stworzUslugeDlaZwierzaka (  
    shared_ptr< Zwierze > zwierze )
```

Funkcja tworzaca usluge dla konkretnego juz zwierzaka.

Parameters

<code>zwierze</code>	- dla niego szukamy odpowiednich usług
----------------------	--

Returns

zwraca wektor usług dla danego zwierza

4.22.2.2 `znajdzPoTypie()`

```
shared_ptr< Usługa > SerwisUslug::znajdzPoTypie (
    string typ )
```

Funkcja szukająca usługi po przekazanym typie w podanej liście usług.

Parameters

<code>typ</code>	- typ zwierza
------------------	---------------

Returns

zwraca szukana usługę

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

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

4.23 SerwisWizyt Class Reference

Klasa odpowiedzialna za serwis wizyt.

```
#include <SerwisWizyt.h>
```

Public Member Functions

- **SerwisWizyt** (`shared_ptr< CzytaczWizyt > czytaczWizyt`)
Konstruktor argumentowy, ustawiający czytacz wizyt.
- StatusWizyty **zarezerwujWizyte** (`Wizyta wizyta`)
Funkcja rezerwująca wizyte.
- StatusWizyty **anulujWizyte** (`int idWizyty`)
Funkcja anulująca wizyte.
- StatusWizyty **anulujWizyteDlaZwierzaka** (`int idZwierzaka`)
Funkcja anulująca wizyte dla zwierza.
- `list< shared_ptr< Wizyta > >` **przeglądWizytPoldWlasiciela** (`int idWlasicielZwierzecia`)
Funkcja zwracająca wizyty dla danego właściciela po id.
- `list< shared_ptr< Wizyta > >` **załadujWizytyDlaDaty** (`int rok, int miesiac, int dzien`)
Funkcja zwracająca wizyty na podstawie przekazanego roku, miesiąca, dnia.
- `list< shared_ptr< Wizyta > >` **przeglądWizytPoldLekarza** (`int idLekarza`)
Funkcja zwracająca wizyty dla danego lekarza po id.
- `int` **getNextId** ()
Funkcja generująca następnego id, które można użyć do zapisu na kolejną wizytę.

4.23.1 Detailed Description

Klasa odpowiedzialna za serwis wizyt.

4.23.2 Member Function Documentation

4.23.2.1 anulujWizyte()

```
StatusWizyty SerwisWizyt::anulujWizyte (
    int idWizyty )
```

Funkcja anulujacy wizyte.

Parameters

<i>idwizyta</i>	- wizyta, ktora chcemy zarezerwowac, szukamy jej po id
-----------------	--

Returns

zwraca status czy operacaj jest powiodla

4.23.2.2 anulujWizyteDlaZwierzaka()

```
StatusWizyty SerwisWizyt::anulujWizyteDlaZwierzaka (
    int idZwierzaka )
```

Funkcja anulujacy wizyte dle zwierzaka.

Parameters

<i>idZwierzaka</i>	- wizyta, ktora chcemy zarezerwowac, szukamy jej po id zwierzaka
--------------------	--

Returns

zwraca status czy operacaj jest powiodla

4.23.2.3 getNextId()

```
int SerwisWizyt::getNextId ( )
```

Funkcja generujaca nastepnego id, ktore mozna uzyc do zapisu na kolejna wizyte.

Returns

zwraca nastepne id do zapisu wizyty

4.23.2.4 przeglądWizytPoldLekarza()

```
list< shared_ptr< Wizyta > > SerwisWizyt::przeglądWizytPoIdLekarza (
    int idLekarza )
```

Funkcja zwracająca wizyty dla danego lekarza po id.

Parameters

<i>idLekarza</i>	- id lekarza, dla którego chcemy znaleźć wizyty
------------------	---

Returns

zwraca liste wizyt dla konkretnego lekarza

4.23.2.5 przeglądWizytPoldWlasciciela()

```
list< shared_ptr< Wizyta > > SerwisWizyt::przeglądWizytPoIdWlasciciela (
    int idWlascicielZwierzecia )
```

Funkcja zwracająca wizyty dla danego właściciela po id.

Parameters

<i>idWlascicielZwierzecia</i>	- id właściciela, dla którego chcemy znaleźć wizyty
-------------------------------	---

Returns

zwraca liste wizyt dla danego właściciela

4.23.2.6 załadujWizytyDlaDaty()

```
list< shared_ptr< Wizyta > > SerwisWizyt::załadujWizytyDlaDaty (
    int rok,
    int miesiac,
    int dzien )
```

Funkcja zwracająca wizyty na podstawie przekazanego roku, miesiąca, dnia.

Parameters

<i>rok</i>	- rok dla przeglądających wizyt
<i>miesiac</i>	- miesiac dla przeglądających wizyt
<i>dzien</i>	- dzien dla przeglądających wizyt

Returns

zwraca liste wizyt dla konkretnej daty

4.23.2.7 zarezerujWizyte()

```
StatusWizyty SerwisWizyt::zarezerujWizyte (
    Wizyta wizyta )
```

Funkcja rezerwujaca wizyte.

Parameters

wizyta	- wizyta, ktorya chcemy zarezerwowac
--------	--------------------------------------

Returns

zwraca status czy operacja jest powiodla

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

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

4.24 SerwisWlasicielaZwierzecia Class Reference

Klasa odpowiedzialna za serwis wlasciciela zwierzecia.

```
#include <SerwisWlasicielaZwierzecia.h>
```

Public Member Functions

- **SerwisWlasicielaZwierzecia** (std::shared_ptr< [CzytaczWlasicielaZwierzecia](#) > czytaczWlasicielaZwierzecia, std::shared_ptr< [SerwisLogowaniaUzytkownika](#) > serwisLogowaniaUzytkownika)
Konstruktor argumentowy, ustawiajacy czytacz wlasciciela zwierzecia, serwis logowania uzytkownika.
- std::shared_ptr< [WlascicielZwierzecia](#) > [zaladujWlasicielaZwierzecia](#) (std::shared_ptr< [Uzytkownik](#) > uzytkownik)
Funkcja zwracajaca wlasciciela zwierzecia, na podstawie przekazanego uzytkownika. Szukamy wlasciciela zwierzecia po id uzytkownika.
- std::shared_ptr< [WlascicielZwierzecia](#) > [zaladujWlasicielaZwierzeciaPold](#) (int id)
Funkcja szukajaca wlasciciela zwierzecia po przekazanym id w podanej liscie wlascieli.

4.24.1 Detailed Description

Klasa odpowiedzialna za serwis wlasciciela zwierzecia.

4.24.2 Member Function Documentation

4.24.2.1 załadujWlascicielaZwierzecia()

```
std::shared_ptr< WlascicielZwierzecia > SerwisWlascicielaZwierzecia::załadujWlascicielaZwierzecia  
(  
    std::shared_ptr< Uzytkownik > uzytkownik )
```

Funkcja zwracająca właściciela zwierzecia, na podstawie przekazanego użytkownika. Szukamy właściciela zwierzecia po id użytkownika.

Parameters

<i>uzytkownik</i>	- wskaźnik na obiekt użytkownik
-------------------	---------------------------------

Returns

zwraca wskaźnik na właściciela zwierzecia, o ile został znaleziony

4.24.2.2 załadujWlascicielaZwierzeciaPold()

```
std::shared_ptr< WlascicielZwierzecia > SerwisWlascicielaZwierzecia::załadujWlascicielaZwierzecia↵  
Pold (   
    int id )
```

Funkcja szukająca właściciela zwierzecia po przekazanym id w podanej liście właścicieli.

Parameters

<i>id</i>	- id szukanego właściciela
-----------	----------------------------

Returns

zwraca szukanego właściciela, jeżeli go nie znajdzie to zwraca NULL

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

- SerwisWlascicielaZwierzecia.h
- SerwisWlascicielaZwierzecia.cpp

4.25 SerwisZwierzat Class Reference

Klasa odpowiedzialna za serwis zwierząt.

```
#include <SerwisZwierzat.h>
```

Public Member Functions

- **SerwisZwierzat** (shared_ptr< [CzytaczZwierzat](#) > czytaczZwierzat)
Konstruktor argumentowy, ustawiający czytacz zwierząt.
- vector< shared_ptr< [Zwierze](#) > > [załadujZwierzetaDlaWlasciciela](#) (int idWlasciciela)
Funkcja zwracająca wszystkich zwierząt dla danego właściciela.
- shared_ptr< [Zwierze](#) > [znajdzPold](#) (int idZwierzecia, vector< shared_ptr< [Zwierze](#) > > zwierzeta)
Funkcja szukająca zwierzęcia po przekazanym id i zwierzętach właściciela.
- shared_ptr< [Zwierze](#) > [załadujZwierzePold](#) (int idZwierzecia)
Funkcja zwracająca zwierze po jego id.
- int [getNastepneId](#) ()
Funkcja generująca następnego id.
- void [dodaj](#) (shared_ptr< [Zwierze](#) > zwierze)
Funkcja dodająca zwierze.
- void [usunZwierzaka](#) (int idZwierzaka)
Funkcja usuwająca zwierze.
- void [zaktualizuj](#) (shared_ptr< [Zwierze](#) > zwierze)
Funkcja aktualizująca zwierze.

4.25.1 Detailed Description

Klasa odpowiedzialna za serwis zwierząt.

4.25.2 Member Function Documentation

4.25.2.1 dodaj()

```
void SerwisZwierzat::dodaj (
    shared_ptr< Zwierze > zwierze )
```

Funkcja dodająca zwierze.

Parameters

zwierze	- zwierze, które chcemy dodać
-------------------------	-------------------------------

4.25.2.2 getNextId()

```
int SerwisZwierzat::getNextId ( )
```

Funkcja generująca następnego id.

Returns

zwraca następne id

4.25.2.3 usunZwierzaka()

```
void SerwisZwierzat::usunZwierzaka (
    int idZwierzaka )
```

Funkcja usuwająca zwierze.

Parameters

<i>idZwierzaka</i>	- id usuwanego zwierzaka
--------------------	--------------------------

4.25.2.4 zaktualizuj()

```
void SerwisZwierzat::zaktualizuj (
    shared_ptr< Zwierze > zwierze )
```

Funkcja aktualizująca zwierze.

Parameters

<i>zwierze</i>	- zwierze, które chcemy zaktualizować (dodane bądź usunięte)
----------------	--

4.25.2.5 załadujZwierzePoId()

```
std::shared_ptr< Zwierze > SerwisZwierzat::załadujZwierzePoId (
    int idZwierzecia )
```

Funkcja zwracająca zwierze po jego id.

Parameters

<i>idZwierzecia</i>	- id zwierzecia szukanego
---------------------	---------------------------

Returns

zwraca zwierze, które szukaliśmy

4.25.2.6 załadujZwierzetaDlaWlasciciela()

```
vector< shared_ptr< Zwierze > > SerwisZwierzat::załadujZwierzetaDlaWlasciciela (
    int idWlasciciela )
```

Funkcja zwracająca wszystkich zwierzeta dla danego właściciela.

Parameters

<i>idWlasciciela</i>	- id właściciela, dla którego chcemy załadować zwierzeta
----------------------	--

Returns

zwracający typ jest wektorem, zwraca liste zwierząt dla właściciela o odpowiednim id

4.25.2.7 `znajdzPold()`

```
std::shared_ptr< Zwierze > SerwisZwierzat::znajdzPoId (
    int idZwierzecia,
    vector< shared_ptr< Zwierze > > zwierzeta )
```

Funkcja szukająca zwierzecia po przekazanym id i zwierzetach właściciela.

Parameters

<i>idZwierzecia</i>	- id zwierzecia
<i>zwierzeta</i>	- wektor wszystkich zwierząt danego właściciela

Returns

zwraca szukane zwierze

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

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

4.26 SesjaLekarza Class Reference

Klasa odpowiedzialna za sesje lekarza (za jego wizyty)

```
#include <SesjaLekarza.h>
```

Public Member Functions

- **SesjaLekarza** (shared_ptr< [Lekarz](#) > lekarz, shared_ptr< [SerwisWizyt](#) > serwisWizyt, shared_ptr< [SerwisZwierzat](#) > serwisZwierzat, shared_ptr< [SerwisUslug](#) > serwisUslug, shared_ptr< [SerwisWlasicielaZwierzecia](#) > serwisWlasicielaZwierzecia)
Konstruktor argumentowy, ustawiający lekarza, serwis wizyt, serwis zwierząt, serwis usług oraz serwis właściciela zwierzecia.
- shared_ptr< [Lekarz](#) > [getLekarz](#) ()
Funkcja zwracająca lekarza Zwraca jego imię, nazwisko, specjalizację, dostępność godzinową.
- shared_ptr< [SerwisWizyt](#) > [getSerwisWizyt](#) ()
Funkcja zwracająca serwis wizyt.
- shared_ptr< [SerwisZwierzat](#) > [getSerwisZwierzat](#) ()
Funkcja zwracająca serwis zwierząt.
- shared_ptr< [SerwisUslug](#) > [getFabrykaUslug](#) ()
Funkcja zwracająca fabrykę usług.
- shared_ptr< [SerwisWlasicielaZwierzecia](#) > [getSerwisWlasicielaZwierzecia](#) ()
Funkcja zwracająca serwis właściciela zwierzecia.

4.26.1 Detailed Description

Klasa odpowiedzialna za sesje lekarza (za jego wizyty)

4.26.2 Member Function Documentation

4.26.2.1 getFabrykaUslug()

```
shared_ptr< SerwisUslug > SesjaLekarza::getFabrykaUslug ( )
```

Funkcja zwracajaca fabryke uslug.

Returns

fabryka usug

4.26.2.2 getLekarz()

```
shared_ptr< Lekarz > SesjaLekarza::getLekarz ( )
```

Funkcja zwracajaca lekarza Zwraca jego imie, nazwisko, specjalizacje, dostepnosc godzinowa.

Returns

lekarz

4.26.2.3 getSerwisWizyt()

```
shared_ptr< SerwisWizyt > SesjaLekarza::getSerwisWizyt ( )
```

Funkcja zwracajaca serwis wizyt.

Returns

serwis wizyt

4.26.2.4 getSerwisWlasicielaZwierzecia()

```
shared_ptr< SerwisWlasicielaZwierzecia > SesjaLekarza::getSerwisWlasicielaZwierzecia ( )
```

Funkcja zwracajaca serwis wlasciciela zwierzecia.

Returns

serwis wlasciciela zwierzecia

4.26.2.5 getSerwisZwierzat()

```
shared_ptr< SerwisZwierzat > SesjaLekarza::getSerwisZwierzat ( )
```

Funkcja zwracająca serwis zwierzat.

Returns

serwis zwierzat

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

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

4.27 SesjaWlascicielaZwierzecia Class Reference

Klasa odpowiedzialna za sesje wlasciciela zwierzecia.

```
#include <SesjaWlascicielaZwierzecia.h>
```

Public Member Functions

- **SesjaWlascicielaZwierzecia** (shared_ptr< [WlascicielZwierzecia](#) > wlascicielZwierzecia, vector< shared_ptr< [Zwierzeta](#) > > zwierzeta)
Konstruktor argumentowy, ustawiajacy wlasciciela zwierzecia, zwierzeta.
- vector< shared_ptr< [Zwierzeta](#) > > [getZwierzeta](#) ()
Funkcja zwracajaca wszystkie zwierzeta aktualnie zalogowanego wlasciciela.
- shared_ptr< [Zwierzeta](#) > [getWybraneZwierzeta](#) ()
Funkcja zwracajaca wszystkie wybrane zwierzeta aktualnie zalogowanego wlasciciela.
- shared_ptr< [WlascicielZwierzecia](#) > [getWlascicielZwierzecia](#) ()
Funkcja zwracajaca zalogowanego wlasciciela.
- void [setWybraneZwierzeta](#) (shared_ptr< [Zwierzeta](#) > zwierzeta)
Funkcja ustawiajaca wybrane zwierzeta.
- void [setWybranaUslug](#) (shared_ptr< [Uslug](#) > wybranaUslug)
Funkcja ustawiajaca wybrana usluge.
- shared_ptr< [Uslug](#) > [getWybranaUslug](#) ()
Funkcja zwracajaca wybrana usluge.
- void [setWybranyMiesiac](#) (shared_ptr< pair< int, time_t > > wybranyMiesiac)
Funkcja ustawiajaca wybrany miesiac.
- shared_ptr< pair< int, time_t > > [getWybranyMiesiac](#) ()
Funkcja zwracajaca wybrany miesiac.
- void [setWybranyDzien](#) (shared_ptr< pair< int, time_t > > wybranyDzien)
Funkcja ustawiajaca wybrany dzien.
- shared_ptr< pair< int, time_t > > [getWybranyDzien](#) ()
Funkcja zwracajaca wybrany dzien.
- void [setWybranyRok](#) (shared_ptr< pair< int, time_t > > wybranyRok)
Funkcja ustawiajaca wybrany rok.
- shared_ptr< pair< int, time_t > > [getWybranyRok](#) ()

- Funkcja zwracająca wybrany rok.*
- void `setWybranaGodzina` (int wybranaGodzina)
- Funkcja ustawiająca wybrana godzinę.*
- int `getWybranaGodzina` ()
- Funkcja zwracająca wybrana godzinę.*
- void `setIdWybranegoLekarza` (int idLekarza)
- Funkcja ustawiająca id wybranego lekarza.*
- int `getIdWybranegoLekarza` ()
- Funkcja zwracająca id wybranego lekarza.*
- bool `czyMoznaUtworzycWizyte` ()
- Funkcja sprawdzająca czy wszystkie potrzebne dane są ustawione do rezerwacji wizyty.*
- void `odswiezZwierzeta` (vector< shared_ptr< `Zwierze` > > zwierzeta)
- Funkcja odswiezająca wszystkie zwierzeta. Jej wywołanie jest wymagane po dodaniu nowego zwierzaka, aby mieć odświeżoną listę (ma zawierać nowo dodanego zwierzaka)*

4.27.1 Detailed Description

Klasa odpowiedzialna za sesję właściciela zwierzęcia.

4.27.2 Member Function Documentation

4.27.2.1 `czyMoznaUtworzycWizyte()`

```
bool SesjaWlascicielaZwierzecia::czyMoznaUtworzycWizyte ( )
```

Funkcja sprawdzająca czy wszystkie potrzebne dane są ustawione do rezerwacji wizyty.

Returns

zwraca true jeżeli można utworzyć wizytę, false jeżeli nie można

4.27.2.2 `getIdWybranegoLekarza()`

```
int SesjaWlascicielaZwierzecia::getIdWybranegoLekarza ( )
```

Funkcja zwracająca id wybranego lekarza.

Returns

zwraca id wybranego lekarza przez właściciela zwierzęcia

4.27.2.3 `getWlascicielZwierzecia()`

```
shared_ptr< WlascicielZwierzecia > SesjaWlascicielaZwierzecia::getWlascicielZwierzecia ( )
```

Funkcja zwracająca zalogowanego właściciela.

Returns

zwraca właściciela zwierzęcia

4.27.2.4 getWybarneZwierze()

```
shared_ptr< Zwierze > SesjaWlascicielaZwierzecia::getWybarneZwierze ( )
```

Funkcja zwracająca wszystkie wybrane zwierzęta aktualnie zalogowanego właściciela.

Returns

zwraca wybrane zwierzęta

4.27.2.5 getWybranaGodzina()

```
int SesjaWlascicielaZwierzecia::getWybranaGodzina ( )
```

Funkcja zwracająca wybraną godzinę.

Returns

zwraca wybraną godzinę przez właściciela zwierzęcia

4.27.2.6 getWybranaUsługa()

```
shared_ptr< Usługa > SesjaWlascicielaZwierzecia::getWybranaUsługa ( )
```

Funkcja zwracająca wybraną usługę.

Returns

zwraca wybraną usługę przez właściciela zwierzęcia

4.27.2.7 getWybranyDzien()

```
shared_ptr< pair< int, time_t > > SesjaWlascicielaZwierzecia::getWybranyDzien ( )
```

Funkcja zwracająca wybrany dzień.

Returns

zwraca wybrany dzień przez właściciela zwierzęcia

4.27.2.8 getWybranyMiesiac()

```
shared_ptr< pair< int, time_t > > SesjaWlascicielaZwierzecia::getWybranyMiesiac ( )
```

Funkcja zwracająca wybrany miesiąc.

Returns

zwraca wybrany miesiąc przez właściciela zwierzęcia

4.27.2.9 getWybranyRok()

```
shared_ptr< pair< int, time_t > > SesjaWlascicielaZwierzecia::getWybranyRok ( )
```

Funkcja zwracająca wybrany rok.

Returns

zwraca wybrany rok przez właściciela zwierzecia

4.27.2.10 getZwierzeta()

```
vector< shared_ptr< Zwierze > > SesjaWlascicielaZwierzecia::getZwierzeta ( )
```

Funkcja zwracająca wszystkie zwierzęta aktualnie zalogowanego właściciela.

Returns

zwraca wektor zwierząt

4.27.2.11 setIdWybranegoLekarza()

```
void SesjaWlascicielaZwierzecia::setIdWybranegoLekarza (
    int idLekarza )
```

Funkcja ustawiająca id wybranego lekarza.

Parameters

<i>idLekarza</i>	- id wybranego lekarza
------------------	------------------------

4.27.2.12 setWybranaGodzina()

```
void SesjaWlascicielaZwierzecia::setWybranaGodzina (
    int wybranaGodzina )
```

Funkcja ustawiająca wybrana godzinę.

Parameters

<i>wybranaGodzina</i>	- wybrana godzina
-----------------------	-------------------

4.27.2.13 setWybranaUsługa()

```
void SesjaWlascicielaZwierzecia::setWybranaUsługa (
    shared_ptr< Usługa > wybranaUsługa )
```

Funkcja ustawiajaca wybrana usluge.

Parameters

<i>wybranaUsługa</i>	- wybrane usluga
----------------------	------------------

4.27.2.14 setWybraneZwierze()

```
void SesjaWlascicielaZwierzecia::setWybraneZwierze (
    shared_ptr< Zwierze > zwierze )
```

Funkcja ustawiajaca wybrane zwierze.

Parameters

<i>zwierze</i>	- wybrane zwierze
----------------	-------------------

4.27.2.15 setWybranyDzien()

```
void SesjaWlascicielaZwierzecia::setWybranyDzien (
    shared_ptr< pair< int, time_t > > wybranyDzien )
```

Funkcja ustawiajaca wybrany dzien.

Parameters

<i>wybranyDzien</i>	- wybrany dzien
---------------------	-----------------

4.27.2.16 setWybranyMiesiac()

```
void SesjaWlascicielaZwierzecia::setWybranyMiesiac (
    shared_ptr< pair< int, time_t > > wybranyMiesiac )
```

Funkcja ustawiajaca wybrany miesiac.

Parameters

<i>wybranyMiesiac</i>	- wybrany miesiac
-----------------------	-------------------

4.27.2.17 setWybranyRok()

```
void SesjaWlascicielaZwierzecia::setWybranyRok (
    shared_ptr< pair< int, time_t > > wybranyRok )
```

Funkcja ustawiajaca wybrany rok.

Parameters

<i>wybranyRok</i>	- wybrany rok
-------------------	---------------

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

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

4.28 SprawdzaczLiczb Class Reference

Klasa odpowiedzialna za sprawdzenie czy to co jest wprowadzane jest liczba.

```
#include <SprawdzaczLiczb.h>
```

Public Member Functions

- bool [jestLiczba](#) (const string &tekst)
Funkcja sprawdzająca czy podana fraza jest liczba.

4.28.1 Detailed Description

Klasa odpowiedzialna za sprawdzenie czy to co jest wprowadzane jest liczba.

4.28.2 Member Function Documentation

4.28.2.1 [jestLiczba\(\)](#)

```
bool SprawdzaczLiczb::jestLiczba (  
    const string & tekst )
```

Funkcja sprawdzająca czy podana fraza jest liczba.

Parameters

<i>tekst</i>	- wprowadzony tekst do sprawdzenia
--------------	------------------------------------

Returns

true jest liczba

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

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

4.29 Usługa Class Reference

Klasa odpowiedzialna za usługi.

```
#include <Usługa.h>
```

Public Member Functions

- **Usługa** (int cena, string nazwaUslugi, string typUslugi)
Konstruktor argumentowy, ustawiający cenę, nazwę usługi i typ usługi.
- int [getCena](#) ()
Funkcja zwracająca cenę.
- string [getTypUslugi](#) ()
Funkcja zwracająca typ usługi.
- string [getNazwaUslugi](#) ()
Funkcja zwracająca nazwę usługi.

4.29.1 Detailed Description

Klasa odpowiedzialna za usługi.

4.29.2 Member Function Documentation

4.29.2.1 [getCena\(\)](#)

```
int Usługa::getCena ( )
```

Funkcja zwracająca cenę.

Returns

zwraca cenę usługi

4.29.2.2 [getNazwaUslugi\(\)](#)

```
std::string Usługa::getNazwaUslugi ( )
```

Funkcja zwracająca nazwę usługi.

Returns

zwraca nazwę usługi

4.29.2.3 getTypUslugi()

```
string Usługa::getTypUslugi ( )
```

Funkcja zwracająca typ usługi.

Returns

zwraca typ usługi

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

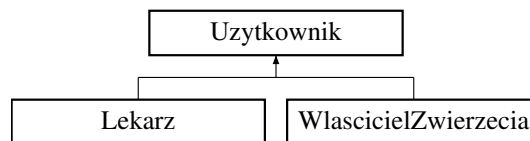
- [Usługa.h](#)
- [Usługa.cpp](#)

4.30 Uzytkownik Class Reference

Klasa odpowiedzialna za uzytkownikow.

```
#include <Uzytkownik.h>
```

Inheritance diagram for Uzytkownik:



Public Member Functions

- **Uzytkownik** (string login, string haslo, int id, [TypUzytkownika](#) typUzytkownika)
Konstruktor argumentowy, ustawiajacy login, haslo, typ uzytkownika.
- string [zamienNaTekst](#) ()
Funkcja zamieniajaca obiekt uzytkownika na tekst, aby mozna bylo go zapisac do pliku.
- int [getId](#) ()
Funkcja zwracajaca id uzytkownika.
- [TypUzytkownika](#) [getTypUzytkownika](#) ()
Funkcja zwracajaca typ uzytkownika (lekarz, wlasciciela zwierzecia)

Static Public Attributes

- static const int **ID_DLA_NIEZNANEGO_UZYTKOWNIKA** = -1
Stala okreslajaca id nieznanego uzytkownika.

4.30.1 Detailed Description

Klasa odpowiedzialna za uzytkownikow.

4.30.2 Member Function Documentation

4.30.2.1 getId()

```
int Uzytkownik::getId ( )
```

Funkcja zwracajaca id uzytkownika.

Returns

zwraca id uzytkownika

4.30.2.2 getTypUzytkownika()

```
TypUzytkownika Uzytkownik::getTypUzytkownika ( )
```

Funkcja zwracajaca typ uzytkownika (lekarz, wlasciciela zwierzecia)

Returns

zwraca typ uzytkownika

4.30.2.3 zamienNaTekst()

```
string Uzytkownik::zamienNaTekst ( )
```

Funkcja zamieniajaca obiekt uzytkownika na tekst, aby mozna bylo go zapisac do pliku.

Returns

tekst, ktory zosatnie zapisany do pliku

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

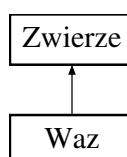
- [Uzytkownik.h](#)
- [Uzytkownik.cpp](#)

4.31 Waz Class Reference

Klasa opisujaca weza, klasa pochodna. Dziedziczy po klasie [Zwierze](#).

```
#include <Waz.h>
```

Inheritance diagram for Waz:



Public Member Functions

- **Waz** (int idZwierzecia, int wiek, [Plec](#) plec, int liczbaNog, string imie, int idWlasciciela, string umaszczenie, string kolor, int [wagaWGramach](#))
Konstruktor argumentowy, ustawiajacy pola ID zwierzecia, wiek, plec, liczbe nog, imie, ID wlasciciela, umaszczenie, kolor, wage w gramach.
- string [dzwiek](#) ()
Metoda, ktora zwroci typ string, w ktorym bedzie zawarty dzwiek, ktory wydaje odpowiedniego typu zwierzaka. Np. Dla weza Ssss ssss.
- string [getWaga](#) ()
Metoda zwracajaca sformatowana wage, w zaleznosci od wagi zwierzecia. Np. [Waz](#) 700g.

Public Member Functions inherited from [Zwierze](#)

- **Zwierze** (int id, int wiek, [Plec](#) plec, int liczbaNog, string imie, int idWlasciciela, string umaszczenie, string kolor, int [wagaWGramach](#))
Konstruktor argumentowy, ustawiajacy id, wiek, plec, liczbe nog, imie, id wlasciciela, umaszczenie, kolor, wage w gramach.
- int [getId](#) ()
Funkcja zwracajaca id zwierzecia.
- int [getWiek](#) ()
Funkcja zwracajaca wiek zwierzecia.
- [Plec](#) [getPlec](#) ()
Zwraca enum plec, okreslajacy czy jest to samiec czy samica.
- int [getLiczbaNog](#) ()
Funkcja zwracajaca liczbe nog zwierzecia.
- string [getImie](#) ()
Funkcja zwracajaca imie zwierzecia.
- int [getIdWlasciciela](#) ()
Funkcja zwracajaca id wlasciciela zwierzaka.
- string [getUmaszczenie](#) ()
Funkcja zwracajaca umaszczenie zwierzaka.
- string [getKolor](#) ()
Funkcja zwracajaca kolor zwierzaka.
- virtual string [dzwiek](#) ()=0
Funkcja wirtualna, zwracajaca dzwiek jaki wydaje konkretne zwierzecie. Klasy dziedziczace po klasie zwierzecie musza zaimplementowac metode [dzwiek\(\)](#). Np. kot ma wydawac dzwiek miau.
- virtual string [getWaga](#) ()=0
Funkcja wirtualna, zwracajaca wage dla konkretnego zwierzecia. Klasy dziedziczace po klasie zwierzecie musza zaimplementowac metode [getWaga\(\)](#). Np. kot ma miec wyswietlana wage w kg, a kanarek w gramach.
- void [setWagaWGramach](#) (int [wagaWGramach](#))
Funkcja ustawiajaca wage, ktora ma byc w gramach.
- void [setImie](#) (string imie)
Funkcja ustawiajaca imie.
- string [zamienNaTekst](#) ()
Funkcja zamieniajaca obiekt zwierzecie na tekst, aby mozna bylo go zapisac do pliku.

Additional Inherited Members

Protected Attributes inherited from [Zwierze](#)

- int [wagaWGramach](#)
Pole klasy, przechowujace wage zwierzecia w gramach.

4.31.1 Detailed Description

Klasa opisująca weza, klasa pochodna. Dziedziczy po klasie [Zwierze](#).

4.31.2 Member Function Documentation

4.31.2.1 dzwiek()

```
string Waz::dzwiek ( ) [virtual]
```

Metoda, która zwróci typ string, w którym będzie zawarty dzwiek, który wydaje odpowiedni typ zwierzaka. Np. Dla weza Ssss ssss.

Implements [Zwierze](#).

4.31.2.2 getWaga()

```
string Waz::getWaga ( ) [virtual]
```

Metoda zwracająca sformatowaną wagę, w zależności od wagi zwierzęcia. Np. [Waz](#) 700g.

Implements [Zwierze](#).

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

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

4.32 Wizyta Class Reference

Klasa odpowiedzialna wizyte.

```
#include <Wizyta.h>
```

Public Member Functions

- **Wizyta** ([DataGodzinaWizyty](#) dataWizty, int idWizyty, int idWlasciciela, int idZwierzecia, int idLekarza, string typUslugi)
Konstruktor argumentowy, ustawiający date wizyty, id wizyty, id właściciela, id zwierzęcia, id lekarza, typ usługi.
- string [zamieNaTekst](#) ()
Funkcja zamieniająca obiekt wizyty na tekst, aby można było go zapisać do pliku.
- [DataGodzinaWizyty](#) [getDataGodzina](#) ()
Funkcja zwracająca date i godzinę wizyty.
- int [getIdWizyty](#) ()
Funkcja generująca id wizyty.
- int [getIdWlasciciela](#) ()
Funkcja generująca id właściciela zwierzęcia.
- int [getIdLekarza](#) ()
Funkcja generująca id lekarza.
- int [getIdZwierzecia](#) ()
Funkcja generująca id zwierzęcia.
- string [getTypUslugi](#) ()
Funkcja zwracająca typ usługi.
- string [getData](#) ()
Funkcja zwracająca date.
- string [getGodzina](#) ()
Funkcja zwracająca godzinę.

4.32.1 Detailed Description

Klasa odpowiedzialna wizyte.

4.32.2 Member Function Documentation

4.32.2.1 `getData()`

```
string Wizyta::getData ( )
```

Funkcja zwracajaca date.

Returns

zwraca date

4.32.2.2 `getDataGodzina()`

```
DataGodzinaWizyty Wizyta::getDataGodzina ( )
```

Funkcja zwracajaca date i godzine wizyty.

Returns

zwraca date i godzine

4.32.2.3 `getGodzina()`

```
string Wizyta::getGodzina ( )
```

Funkcja zwracajaca godzine.

Returns

zwraca godzine

4.32.2.4 `getIdLekarza()`

```
int Wizyta::getIdLekarza ( )
```

Funkcja generujaca id lekarza.

Returns

zwraca id lekarza

4.32.2.5 getIdWizyty()

```
int Wizyta::getIdWizyty ( )
```

Funkcja generujaca id wizyty.

Returns

zwraca id wizyty

4.32.2.6 getIdWlasciciela()

```
int Wizyta::getIdWlasciciela ( )
```

Funkcja generujaca id wlasciciela zwierzecia.

Returns

zwraca id wlasciciela zwierzecia

4.32.2.7 getIdZwierzecia()

```
int Wizyta::getIdZwierzecia ( )
```

Funkcja generujaca id zwierzecia.

Returns

zwraca id zwierzecia

4.32.2.8 getTypUslugi()

```
string Wizyta::getTypUslugi ( )
```

Funkcja zwracajaca typ uslugi.

Returns

zwraca typ uslugi

4.32.2.9 zamieNaTekst()

```
string Wizyta::zamieNaTekst ( )
```

Funkcja zamieniajaca obiekt wizyty na tekst, aby mozna bylo go zapisac do pliku.

Returns

tekst, ktory zosatnie zapisany do pliku

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

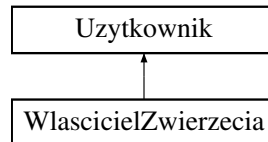
- [Wizyta.h](#)
- [Wizyta.cpp](#)

4.33 WlascicielZwierzecia Class Reference

Klasa odpowiedzialna za wlasciciela zwierzecia.

```
#include <WlascicielZwierzecia.h>
```

Inheritance diagram for WlascicielZwierzecia:



Public Member Functions

- **WlascicielZwierzecia** (string imie, int id, string miasto, string kodPocztowy, string ulicaINumerDomu, string numerTelefonu, [Uzytkownik](#) &uzytkownik)
Konstruktor argumentowy, ustawiajacy imie, id, miasto, kod pocztowy, ulice i numer domu, numer telefonu, uzytkownika.
- string [getImie](#) ()
Funkcja zwracajaca imie.
- int [getId](#) ()
Funkcja zwracajaca id wlasciceila zwierzecia.
- string [getMiasto](#) ()
Funkcja zwracajaca miasto.
- string [getKodPoczowy](#) ()
Funkcja zwracajaca kod pocztowy.
- string [getUlicaINumerDomu](#) ()
Funkcja zwracajaca ulice i numer domu.
- string [getNumerTelefonu](#) ()
Funkcja zwracajaca numer telefonu.
- void [setMiasto](#) (string miasto)
Funkcja ustawiajaca miasto.
- void [setKodPocztowy](#) (string kodPocztowy)
Funkcja ustawiajaca kod pocztowy.
- void [setUlicaINumerDomu](#) (string ulicaINumerDomu)
Funkcja ustawiajaca ulice i numer domu.
- void [setNumerTelefonu](#) (string numerTelefonu)
Funkcja ustawiajaca numer telefonu.

Public Member Functions inherited from [Uzytkownik](#)

- **Uzytkownik** (string login, string haslo, int id, [TypUzytkownika](#) typUzytkownika)
Konstruktor argumentowy, ustawiajacy login, haslo, typ uzytkownika.
- string [zamienNaTekst](#) ()
Funkcja zamieniajaca obiekt uzytkownika na tekst, aby mozna bylo go zapisac do pliku.
- int [getId](#) ()
Funkcja zwracajaca id uzytkownika.
- [TypUzytkownika](#) [getTypUzytkownika](#) ()
Funkcja zwracajaca typ uzytkownika (lekarz, wlasciciela zwierzecia)

Additional Inherited Members

Static Public Attributes inherited from [Uzytkownik](#)

- static const int **ID_DLA_NIEZNANEGO_UZYTEKOWNIKA** = -1
Stala okreslajaca id nieznanego uzytkownika.

4.33.1 Detailed Description

Klasa odpowiedzialna za wlasciciela zwierzecia.

4.33.2 Member Function Documentation

4.33.2.1 getId()

```
int WlascicielZwierzecia::getId ( )
```

Funkcja zwracajaca id wlasciciela zwierzecia.

Returns

id wlasciciela zwierzecia

4.33.2.2 getImie()

```
string WlascicielZwierzecia::getImie ( )
```

Funkcja zwracajaca imie.

Returns

imie wlasciciela zwierzecia

4.33.2.3 getKodPoczowy()

```
string WlascicielZwierzecia::getKodPoczowy ( )
```

Funkcja zwracajaca kod pocztowy.

Returns

kod pocztowy wlasciciela zwierzecia

4.33.2.4 getMiasto()

```
string WlascicielZwierzecia::getMiasto ( )
```

Funkcja zwracająca miasto.

Returns

miasto wlasciciela zwierzecia

4.33.2.5 getNumerTelefonu()

```
string WlascicielZwierzecia::getNumerTelefonu ( )
```

Funkcja zwracająca numer telefonu.

Returns

numer telefonu wlasciciela zwierzecia

4.33.2.6 getUlicaINumerDomu()

```
string WlascicielZwierzecia::getUlicaINumerDomu ( )
```

Funkcja zwracająca ulice i numer domu.

Returns

ulice i numer domu wlasciciela zwierzecia

4.33.2.7 setKodPocztowy()

```
void WlascicielZwierzecia::setKodPocztowy (
    string kodPocztowy )
```

Funkcja ustawiająca kod pocztowy.

Parameters

<i>kodPocztowy</i>	- kodPocztowy
--------------------	---------------

4.33.2.8 setMiasto()

```
void WlascicielZwierzecia::setMiasto (
    string miasto )
```

Funkcja ustawiajaca miasto.

Parameters

<i>miasto</i>	- miasto
---------------	----------

4.33.2.9 setNumerTelefonu()

```
void WlascicielZwierzecia::setNumerTelefonu (  
    string numerTelefonu )
```

Funkcja ustawiajaca numer telefonu.

Parameters

<i>numerTelefonu</i>	- numer telefonu
----------------------	------------------

4.33.2.10 setUlicaINumerDomu()

```
void WlascicielZwierzecia::setUlicaINumerDomu (  
    string ulicaINumerDomu )
```

Funkcja ustawiajaca ulice i numer domu.

Parameters

<i>ulicaINumerDomu</i>	- ulica i numer domu
------------------------	----------------------

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

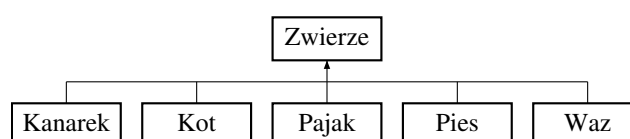
- [WlascicielZwierzecia.h](#)
- WlascicielZwierzecia.cpp

4.34 Zwierze Class Reference

Klasa odpowiedzialna za zwierzeta.

```
#include <Zwierze.h>
```

Inheritance diagram for Zwierze:



Public Member Functions

- **Zwierze** (int id, int wiek, [Plec](#) plec, int liczbaNog, string imie, int idWlasciciela, string umaszczenie, string kolor, int [wagaWGramach](#))
Konstruktor argumentowy, ustawiający id, wiek, plec, liczbę nog, imię, id właściciela, umaszczenie, kolor, wagę w gramach.
- int [getId](#) ()
Funkcja zwracająca id zwierzęcia.
- int [getWiek](#) ()
Funkcja zwracająca wiek zwierzęcia.
- [Plec](#) [getPlec](#) ()
Zwraca enum plec, określający czy jest to samiec czy samica.
- int [getLiczbaNog](#) ()
Funkcja zwracająca liczbę nog zwierzęcia.
- string [getImie](#) ()
Funkcja zwracająca imię zwierzęcia.
- int [getIdWlasciciela](#) ()
Funkcja zwracająca id właściciela zwierzęcia.
- string [getUmaszczenie](#) ()
Funkcja zwracająca umaszczenie zwierzęcia.
- string [getKolor](#) ()
Funkcja zwracająca kolor zwierzęcia.
- virtual string [dzwiek](#) ()=0
Funkcja wirtualna, zwracająca dźwięk jaki wydaje konkretne zwierze. Klasy dziedziczące po klasie zwierze muszą zaimplementować metodę [dzwiek\(\)](#). Np. kot ma wydawać dźwięk miau.
- virtual string [getWaga](#) ()=0
Funkcja wirtualna, zwracająca wagę dla konkretnego zwierzęcia. Klasy dziedziczące po klasie zwierze muszą zaimplementować metodę [getWaga\(\)](#). Np. kot ma mieć wyświetlaną wagę w kg, a kanarek w gramach.
- void [setWagaWGramach](#) (int [wagaWGramach](#))
Funkcja ustawiająca wagę, która ma być w gramach.
- void [setImie](#) (string imie)
Funkcja ustawiająca imię.
- string [zamienNaTekst](#) ()
Funkcja zamieniająca obiekt zwierze na tekst, aby można było go zapisać do pliku.

Protected Attributes

- int [wagaWGramach](#)
Pole klasy, przechowujące wagę zwierzęcia w gramach.

4.34.1 Detailed Description

Klasa odpowiedzialna za zwierzęta.

4.34.2 Member Function Documentation

4.34.2.1 dzwiek()

```
virtual string Zwierze::dzwiek ( ) [pure virtual]
```

Funkcja wirtualna, zwracajaca dzwiek jaki wydaje konkretne zwierze. Klasy dziedziczace po klasie zwierze musza zaimplementowac metode [dzwiek\(\)](#). Np. kot ma wydawac dzwiek miau.

Returns

dzwiek jaki wydaje dane zwierze

Implemented in [Kanarek](#), [Kot](#), [Pajak](#), [Pies](#), and [Waz](#).

4.34.2.2 getId()

```
int Zwierze::getId ( )
```

Funkcja zwracajaca id zwierzecia.

Returns

id zwierzecia

4.34.2.3 getIdWlasciciela()

```
int Zwierze::getIdWlasciciela ( )
```

Funkcja zwracajaca id wlasciciela zwierzaka.

Returns

id wlasciciela zwierzaka

4.34.2.4 getImie()

```
std::string Zwierze::getImie ( )
```

Funkcja zwracajaca imie zwierzecia.

Returns

imie zwierzecia

4.34.2.5 getKolor()

```
std::string Zwierze::getKolor ( )
```

Funkcja zwracająca kolor zwierzaka.

Returns

kolor

4.34.2.6 getliczaNog()

```
int Zwierze::getliczaNog ( )
```

Funkcja zwracająca liczbę nog zwierzecia.

Returns

liczba nog zwierzecia

4.34.2.7 getPlec()

```
Plec Zwierze::getPlec ( )
```

Zwraca enum plec, określający czy jest to samiec czy samica.

Returns

plec

4.34.2.8 getUmaszczenie()

```
std::string Zwierze::getUmaszczenie ( )
```

Funkcja zwracająca umaszczenie zwierzaka.

Returns

umaszczenie

4.34.2.9 getWaga()

```
virtual string Zwierze::getWaga ( ) [pure virtual]
```

Funkcja wirtualna, zwracająca wagę dla konkretnego zwierzecia. Klasy dziedziczące po klasie zwierze muszą zaimplementować metodę [getWaga\(\)](#). Np. kot ma mieć wyświetlaną wagę w kg, a kanarek w gramch.

Returns

dźwięk jaki wydaje dane zwierze

Implemented in [Kanarek](#), [Kot](#), [Pajak](#), [Pies](#), and [Waz](#).

4.34.2.10 `getWiek()`

```
int Zwierze::getWiek ( )
```

Funkcja zwracająca wiek zwierzęcia.

Returns

wiek zwierzęcia

4.34.2.11 `setImie()`

```
void Zwierze::setImie (
    string imie )
```

Funkcja ustawiająca imię.

Parameters

<i>imie</i>	- imię zwierzęcia
-------------	-------------------

4.34.2.12 `setWagaWGramach()`

```
void Zwierze::setWagaWGramach (
    int wagaWGramach )
```

Funkcja ustawiająca wagę, która ma być w gramach.

Parameters

<i>wagaWGramach</i>	- waga w gramach
---------------------	------------------

4.34.2.13 `zamienNaTekst()`

```
string Zwierze::zamienNaTekst ( )
```

Funkcja zamieniająca obiekt zwierzę na tekst, aby można było go zapisać do pliku.

Returns

tekst, który zostanie zapisany do pliku

4.34.3 Member Data Documentation

4.34.3.1 `wagaWGramach`

```
int Zwierze::wagaWGramach [protected]
```

Pole klasy, przechowujące wagę zwierzęcia w gramach.

Returns

waga w gramach

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

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

Chapter 5

File Documentation

5.1 CzytaczGrafikuPracy.h File Reference

```
#include "CzytaczPlikow.h"
#include <string>
```

Classes

- class [CzytaczGrafikuPracy](#)

Klasa do czytania pliku, w którym jest zapisany grafik pracy kliniki.

5.2 CzytaczGrafikuPracy.h

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

```
00001
00004 #pragma once
00005 #include "CzytaczPlikow.h"
00006 #include <string>
00007
00008 using namespace std;
00009
00014 class CzytaczGrafikuPracy : public CzytaczPlikow
00015 {
00021     string podajNazwePliku();
00022 };
00023
00024
```

5.3 CzytaczLekarza.h File Reference

```
#include "CzytaczPlikow.h"
#include <string>
```

Classes

- class [CzytaczLekarza](#)

Klasa do czytania pliku, w którym są zapisani lekarze.

5.4 CzytaczLekarza.h

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

```
00001
00004 #pragma once
00005 #include "CzytaczPlikow.h"
00006 #include <string>
00007 using namespace std;
00008
00009
00014 class CzytaczLekarza : public CzytaczPlikow
00015 {
00021     string podajNazwePliku();
00022 };
```

5.5 CzytaczPlikow.h File Reference

```
#include <vector>
#include <string>
```

Classes

- class [CzytaczPlikow](#)

Klasa bazowa do czytania pliku, po której dziedziczą inne klasy, które muszą zaimplementować funkcję podajNazwePliku.

5.6 CzytaczPlikow.h

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

```
00001
00004 #pragma once
00005 #include <vector>
00006 #include <string>
00007
00008 using namespace std;
00013 class CzytaczPlikow {
00014
00015 private:
00021     string zapiszZawartoscPlikuDoZmiennej(ifstream& plik);
00028     int znajdzIndeksOstatniejKolumny(string& calyTekst);
00036     int znajdzWystapienieSrodkowejKolumny(string& calyTekst, int n);
00037
00038 public:
00043     string wczytajPlik();
00048     void dopiszDoPliku(string& nowyWiersz);
00054     virtual string podajNazwePliku() = 0;
00060     vector<string> zmaiennaWiersze(string& trescPliku);
00071     string getPozycjaTextu(string calyTekst, int pozycja);
00078     void usunWierszPoId(int indeksId, int id, bool czyJestUnikalny);
00085     int getPozycjaWiersza(vector<string> wiersze, string wiersz);
00091     void zapiszDoPliku(const vector<string>& wektor, const string& nazwaPliku);
00104     int getNextId(int indeksId);
00105 };
00106
```

5.7 CzytaczUzytkownikow.h File Reference

```
#include "CzytaczPlikow.h"
```

Classes

- class [CzytaczUzytkownikow](#)
Klasa bazowa do czytania uzytkownikow (lekarz, wlasciciel zwierzecia)

5.8 CzytaczUzytkownikow.h

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

```
00001
00004 #pragma once
00005 #include "CzytaczPlikow.h"
00006
00007 using namespace std;
00008
00013 class CzytaczUzytkownikow : public CzytaczPlikow
00014 {
00015 public:
00021     string podajNazwePliku();
00022 };
00023
```

5.9 CzytaczWizyt.h File Reference

```
#include "CzytaczPlikow.h"
#include "Wizyta.h"
```

Classes

- class [CzytaczWizyt](#)
Klasa bazowa do czytania wizyt.

5.10 CzytaczWizyt.h

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

```
00001
00004 #pragma once
00005 #include "CzytaczPlikow.h"
00006 #include "Wizyta.h"
00007
00008 using namespace std;
00009
00014 class CzytaczWizyt : public CzytaczPlikow
00015 {
00016 public:
00022     string podajNazwePliku();
00023 };
00024
```

5.11 CzytaczWlasicielaZwierzecia.h

```
00001
00004 #pragma once
00005 #include "CzytaczPlikow.h"
00006 #include <string>
00007 using namespace std;
00008
00013 class CzytaczWlasicielaZwierzecia : public CzytaczPlikow
00014 {
00020     string podajNazwePliku();
00021 };
00022
```

5.12 CzytaczZwierzat.h File Reference

```
#include "CzytaczPlikow.h"
```

Classes

- class [CzytaczZwierzat](#)
Klasa bazowa do czytania zwierzat.

5.13 CzytaczZwierzat.h

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

```
00001
00004 #pragma once
00005 #include "CzytaczPlikow.h"
00006
00007 using namespace std;
00008
00013 class CzytaczZwierzat: public CzytaczPlikow {
00014 public:
00020     string podajNazwePliku();
00021 };
00022
```

5.14 DniTygodnia.h File Reference

Enumerations

- enum [DniTygodnia](#) {
 Poniedzialek , **Wtorek** , **Sroda** , **Czwartek** ,
 Piatek , **Sobota** , **Niedziela** }
Typ enum reprezentujący wszystkie dni tygodnia.

5.15 DniTygodnia.h

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

```
00001
00004 #pragma once
00005
00010 enum DniTygodnia
00011 {
00012     Poniedzialek,
00013     Wtorek,
00014     Sroda,
00015     Czwartek,
00016     Piatek,
00017     Sobota,
00018     Niedziela
00019 };
```

5.16 GodzinyPracy.h File Reference

Classes

- class [GodzinyPracy](#)

Klasa opisująca strukturę godzin pracy, gdzie mamy godzinę rozpoczęcia i zakończenia wraz z minutami.

5.17 GodzinyPracy.h

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

```
00001
00004 #pragma once
00009 class GodzinyPracy {
00010
00011 private:
00012     int godzinaRozpoczecia;
00013     int minutaRozpoczecia;
00014     int godzinaZakonczenia;
00015     int minutaZakonczenia;
00016
00017 public:
00021     GodzinyPracy();
00025     GodzinyPracy(int godzinaRozpoczecia, int minutaRozpoczecia, int godzinaZakonczenia, int
        minutaZakonczenia);
00030     int getGodzinaRozpoczecia();
00035     int getGodzinaZakonczenia();
00040     int getMinutaRozpoczecia();
00045     int getMinutaZakonczenia();
00046 };
```

5.18 Kanarek.h File Reference

```
#include "Zwierze.h"
```

Classes

- class [Kanarek](#)

Klasa opisująca kanarkę, klasa pochodna. Dziedziczy po klasie [Zwierze](#).

5.19 Kanarek.h

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

```
00001
00004 #pragma once
00005 #include "Zwierze.h"
00006
00007 using namespace std;
00013 class Kanarek : public Zwierze {
00014
00015 public:
00016
00020     Kanarek(int idZwierzecia, int wiek, Plec plec, int liczbaNog, string imie, int idWlasciciela,
        string umaszczenie, string kolor, int wagaWGramach);
00025     string dzwiek();
00030     string getWaga();
00031
00032 };
00033
```

5.20 Kot.h File Reference

```
#include "Zwierze.h"
#include <string>
```

Classes

- class [Kot](#)

Klasa opisująca kota, klasa pochodna. Dziedziczy po klasie [Zwierze](#).

5.21 Kot.h

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

```
00001
00004 #pragma once
00005 #include "Zwierze.h"
00006 #include<string>
00007
00008 using namespace std;
00009
00014 class Kot :public Zwierze {
00015
00016 public:
00020     Kot(int id, int wiek, Plec plec, int liczbaNog, string imie, int idWlasciciela, string
        umaszczenie, string kolor, int wagaWGramach);
00025     string dzwiek();
00030     string getWaga();
00031 };
00032
```

5.22 Lekarz.h File Reference

```
#include <unordered_map>
#include <string>
#include "Uzytkownik.h"
#include "GodzinyPracy.h"
#include "DniTygodnia.h"
```

Classes

- class [Lekarz](#)

Klasa opisująca lekarza, klasa pochodna. Dziedziczy po klasie [Uzytkownik](#).

5.23 Lekarz.h

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

```

00001
00004 #pragma once
00005 #include <unordered_map>
00006 #include <string>
00007 #include "Uzytkownik.h"
00008 #include "GodzinyPracy.h"
00009 #include "DniTygodnia.h"
00010
00011 using namespace std;
00012
00017 class Lekarz : public Uzytkownik{
00018
00019 private :
00020     string imie;
00021     string nazwisko;
00022     string specjalizacja;
00023     unordered_map<DniTygodnia, GodzinyPracy> dostepnosc;
00024
00025 public :
00029     Lekarz(string imie, string nazwisko, string specjalizacja, Uzytkownik& uzytkownik);
00034     string getImie();
00039     string getNazwisko();
00044     string getSpecjalizacja();
00050     GodzinyPracy zwrocGodzinyPracyDlaDnia(DniTygodnia dzien) {
00051         return dostepnosc[dzien];
00052     }
00058     bool czyDostepnyWGodzinach(int godzina, DniTygodnia dzienTygodnia);
00059 };

```

5.24 MenuLekarza.h File Reference

```

#include "SesjaLekarza.h"
#include <memory>

```

Classes

- class [MenuLekarza](#)
Klasa opisujaca cale menu lekarza.

Enumerations

- enum class [OPCJA_POCZATKOWA_LEKARZA](#) { PRZEGLAD_WIZYT , ANULUJ_WIZYTE , WYJDZ , NIE_WYBRANO_ZADNEJ_OPCJI }
Typ enum reprezentujacy wszystkie dostepne opcje menu dla lekarza.

5.25 MenuLekarza.h

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

```

00001
00004 #pragma once
00005 #include "SesjaLekarza.h"
00006 #include <memory>
00007
00008 using namespace std;
00009
00014 enum class OPCJA_POCZATKOWA_LEKARZA
00015 {
00016     PRZEGLAD_WIZYT, ANULUJ_WIZYTE, WYJDZ, NIE_WYBRANO_ZADNEJ_OPCJI

```

```

00017 };
00018
00023 class MenuLekarza
00024 {
00025 private:
00026
00027     shared_ptr<SesjaLekarza> sesjaLekarza;
00028 public:
00032     MenuLekarza(shared_ptr<SesjaLekarza> sesjaLekarza);
00037     OPCJA_POCZATKOWA_LEKARZA wyswietlMenuPocztkowe();
00041     void przegladWizyt();
00045     void anulujWizyte();
00046 };
00047

```

5.26 MenuLogowania.h File Reference

```

#include <string>
#include "MenuLogowaniaStatusu.h"

```

Classes

- class [MenuLogowania](#)
Klasa zajmująca się obsługą menu logowania.

5.27 MenuLogowania.h

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

```

00001
00004 #pragma once
00005 #include <string>
00006 #include "MenuLogowaniaStatusu.h"
00007
00008 using namespace std;
00009
00014 class MenuLogowania
00015 {
00016 private:
00017
00018     string login = "";
00019     string haslo = "";
00020
00021 public:
00026     string getLogin();
00031     string getHaslo();
00036     MenuLogowaniaStatusu wyswietlMenuPocztkowe();
00040     void wyswietMenuLogowania();
00041 };
00042

```

5.28 MenuLogowaniaStatusu.h File Reference

Enumerations

- enum [MenuLogowaniaStatusu](#) { WYBRANO_LOGOWANIE , WYBRANO_WYSJSCIE , BLAD_↵
WYBRANEJ_OPCJI }
Dostępne opcje logowania na początku (logowanie lub wyjście)

5.29 MenuLogowaniaStatusu.h

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

```
00001
00004 #pragma once
00005
00010 enum MenuLogowaniaStatusu
00011 {
00012     WYBRANO_LOGOWANIE, WYBRANO_WYSJSCIE, BLAD_WYBRANEJ_OPCJI
00013 };
00014
00015
```

5.30 MenuWlasicielaZwierzecia.h

```
00001
00004 #pragma once
00005 #include "SesjaWlascicielaZwierzecia.h"
00006 #include "PlanerWizyt.h"
00007 #include "SerwisLekarza.h"
00008 #include "SerwisUslug.h"
00009 #include "SerwisZwierzat.h"
00010 #include "SerwisWizyt.h"
00011
00016 enum OPCJA_POCZATKOWE_WLASICIELA
00017 {
00018     ZAPLANUJ_WIZYTE, PRZEGLAD_WIZYT, ANULUJ_WIZYTE, WYJDZ, DODAJ_ZWIERZAKA, NIE_WYBRANO_ZADNEJ_OPCJI,
00019     EDYCJA_ZWIERZAKA
00020 };
00025 enum OPCJA_PLANOWANIA_WIZYTY {
00026     WYBRANO_ZWIERZAKA, BLEDNA_OPCJA_WIZYTY
00027 };
00028
00033 enum OPCJA_USLUGI {
00034     WYBRANO_USLUGE, BLEDNA_OPCJA, USLUGA_POZA_ZAKRESEM
00035 };
00036
00041 enum class OPCJA_TERMINU_WIZYTY {
00042     WYBRANO_TERMIN, BLEDNA_OPCJA_TERMINU, TERMIN_POZA_ZAKRESEM
00043 };
00044
00049 class MenuWlasicielaZwierzecia
00050 {
00051 private:
00052     shared_ptr<SesjaWlascicielaZwierzecia> sesja;
00053     shared_ptr<PlanerWizyt> planer;
00054     shared_ptr<SerwisUslug> serwisUslug;
00055     shared_ptr<SerwisLekarza> serwisLekarza;
00056     shared_ptr<SerwisWizyt> serwisWizyt;
00057     shared_ptr<SerwisZwierzat> serwisZwierzat;
00058 public:
00062     MenuWlasicielaZwierzecia(shared_ptr<SesjaWlascicielaZwierzecia> sesja, shared_ptr<PlanerWizyt>
        planer, shared_ptr<SerwisUslug> serwisUslug, shared_ptr<SerwisLekarza> serwisLekarza,
        shared_ptr<SerwisWizyt> serwisWizyt, shared_ptr<SerwisZwierzat> serwisZwierzat);
00067     OPCJA_POCZATKOWE_WLASICIELA wyswietlMenuPoczątkowe();
00073     OPCJA_PLANOWANIA_WIZYTY wyswietlMenuDoPlanowaniaWizyty();
00078     OPCJA_USLUGI wyswietlMenuUslug();
00084     OPCJA_TERMINU_WIZYTY wyswietlMenuWyboruTerminu();
00088     void wyswietlAnulowanieWizyty();
00092     void wyswietlWizyty();
00097     void dodajZwierzaka();
00101     void edycjaZwierzat();
00102};
```

5.31 Pajak.h File Reference

```
#include "Zwierze.h"
```

Classes

- class [Pajak](#)

Klasa opisująca pajaka, klasa pochodna. Dziedziczy po klasie [Zwierze](#).

5.32 Pajak.h

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

```
00001
00004 #pragma once
00005 #include "Zwierze.h"
00006
00007 using namespace std;
00008
00013 class Pajak : public Zwierze {
00014
00015 public:
00019     Pajak(int idZwierzecia, int wiek, Plec plec, int liczbaNog, string imie, int idWlasciciela, string
        umaszczzenie, string kolor, int wagaWGramch);
00024     string dzwiek();
00029     string getWaga();
00030 };
00031
```

5.33 Pies.h File Reference

```
#include "Zwierze.h"
```

Classes

- class [Pies](#)

Klasa opisująca psa, klasa pochodna. Dziedziczy po klasie [Zwierze](#).

5.34 Pies.h

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

```
00001
00004 #pragma once
00005 #include "Zwierze.h"
00006
00007 using namespace std;
00008
00013 class Pies : public Zwierze {
00014
00015 public:
00019     Pies(int idZwierzecia, int wiek, Plec plec, int liczbaNog, string imie, int idWlasciciela, string
        umaszczzenie, string kolor, int wagaWGramach);
00024     string dzwiek();
00029     string getWaga();
00030 };
```

5.35 PlanerWizyt.h File Reference

```
#include <iostream>
#include <list>
#include <iomanip>
#include <ctime>
#include <chrono>
#include <sstream>
#include "GodzinyPracy.h"
#include "DniTygodnia.h"
#include "SerwisGrafikuPracy.h"
#include "SerwisWizyt.h"
#include <unordered_map>
```

Classes

- class [PlanerWizyt](#)

Klasa odpowiedzialna za planowanie wizyt oraz wszelkie manipulacje data i czasem.

5.36 PlanerWizyt.h

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

```
00001
00004 #pragma once
00005 #include <iostream>
00006 #include <list>
00007 #include <iomanip>
00008 #include <ctime>
00009 #include <chrono>
00010 #include <sstream>
00011 #include "GodzinyPracy.h"
00012 #include "DniTygodnia.h"
00013 #include "SerwisGrafikuPracy.h"
00014 #include "SerwisWizyt.h"
00015 #include <unordered_map>
00016
00017 using namespace std;
00018
00023 class PlanerWizyt
00024 {
00025 private:
00026     unordered_map<DniTygodnia, GodzinyPracy> dniOtwarcia;
00027     shared_ptr<SerwisGrafikuPracy> serwisGrafikuPracy;
00028     shared_ptr<SerwisWizyt> serwisWizyt;
00029     string zaladuj(time_t aktualnyCzas, std::string format);
00030 public:
00034     PlanerWizyt(shared_ptr<SerwisGrafikuPracy> serwisGrafikuPracy, shared_ptr<SerwisWizyt>
00035     serwisWizyt);
00040     string zaladujAktualnyDzienTygodnia(time_t currentTime);
00046     string zaladujMiesiac(time_t currentTime);
00052     string zaladujDzienMiesiaca(time_t currentTime);
00058     string zaladujRok(time_t czas);
00068     pair<DniTygodnia, int> zaladujDostepnaGodzineNaDzienTygodnia(DniTygodnia aktualnyDzienTygodnia,
00069     DniTygodnia terminNaNastepnyDzien, int aktualnaGodzina, int pierwszaDostepnaGodzinaWizty);
00073     pair<DniTygodnia, int> znajdzPierwszyDostepnyTermin();
00078     pair<string, time_t> zaladujPierwszaDostepnaDate();
00079
00086     pair<string, time_t> zaladujOstatniaDostepnaDate(time_t poczynatyCzas, string format);
00087
00098     time_t dodajDzien(time_t poczynatyCzas, int liczbaDni);
00099
00110     time_t dodajGodzine(time_t poczynatyCzas, int liczbaGodzin);
00123     time_t ustawCzas(time_t poczynatyCzas, int godzina, int minuta);
00140     time_t dodajGodzinePlusZaokraglenie(time_t poczynatyCzas, int liczbaGodzin);
00150     DniTygodnia zamienNaDniTygonida(string dniTygodniaTekst);
00160     DniTygodnia zaladujNastepnyDzienTygodnia(DniTygodnia aktualnyDzienTygodnia);
00166     std::pair<DniTygodnia, time_t> zaladujDostepyCzas(time_t aktualnyCzas);
00172     void wyswietl(time_t currentTime, string format);
00178     DniTygodnia zamienNaDniTygonida(time_t czas);
00185     time_t getGodzinaZamknieciaWDniu(DniTygodnia ostatniDostepnyDzienTygodnia, time_t czas);
00186
00193     unordered_map<int, time_t> wygenerujMiesiace(time_t czasRozpoczecia, time_t czasZakonczenia);
00194
00205     time_t dodajMiesiac(time_t poczynatyCzas, int liczbaMiesiecy);
00215     int getMiesiac(time_t czas);
00216
00225     string zamienNaMiesiac(time_t czas);
00226
00235     std::string zamienZAngielskiegoNaPolski(const string& miesiacPoAngielsku);
00236
00244     void wypiszKalendarz(int rok, int miesiac, int dzienPoczynaty, int dzienKoncowy);
00251     bool czyPierwszDostepnyMiesiac(unordered_map<int, time_t> miesiace, time_t miesiac);
00258     bool czyOstatniDostepnyMiesiac(unordered_map<int, time_t> miesiace, time_t miesiac);
00259
00265     int ostatniDzienMiesiaca(time_t czasWMillisekundach);
00266
00275     unordered_map<int, time_t> wygenerujDni(int rok, int miesiac, int dzienPoczynaty, int
00276     dzienKoncowy);
00276
00288     bool czyJestDniem(DniTygodnia wybranyDzien, time_t wybranyCzas);
00295     unordered_map<int, unordered_map<int, shared_ptr<Lekarz>> wygenerujBlokGodzinowe(time_t
00296     wybranyDzien, list<shared_ptr<Lekarz> lekarze);
00300     void utworzWizyte(Wizyta wizyta);
```

```
00301
00306     int wygenerujIdWizyty();
00307 };
00308
```

5.37 Plec.h File Reference

Enumerations

- enum [Plec](#) { **SAMIEC** , **SAMICA** }
Typ enum reprezentujący płci zwierząt (samiec, samica)

5.38 Plec.h

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

```
00001
00004 #pragma once
00005
00010 enum Plec { SAMIEC, SAMICA };
```

5.39 SerwisGrafikuPracy.h File Reference

```
#include "CzytaczGrafikuPracy.h"
#include "Lekarz.h"
#include <unordered_map>
#include <memory>
```

Classes

- class [SerwisGrafikuPracy](#)
Klasa odpowiedzialna za serwis grafiku pracy.

5.40 SerwisGrafikuPracy.h

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

```
00001
00004 #pragma once
00005 #include "CzytaczGrafikuPracy.h"
00006 #include "Lekarz.h"
00007 #include <unordered_map>
00008 #include <memory>
00009
00014 class SerwisGrafikuPracy {
00015 private:
00016     shared_ptr<CzytaczGrafikuPracy> czytaczGrafikuPracy;
00017 public:
00021     SerwisGrafikuPracy(shared_ptr<CzytaczGrafikuPracy> czytaczGrafikuPracy);
00027     unordered_map<DniTygodnia, GodzinyPracy> zaladujGrafik(shared_ptr<Lekarz> lekarz);
00028 };
00029
```

5.41 SerwisLekarza.h File Reference

```
#include "CzytaczLekarza.h"
#include "Lekarz.h"
#include <vector>
#include "SerwisLogowaniaUzytkownika.h"
#include <memory>
```

Classes

- class [SerwisLekarza](#)

Klasa odpowiedzialna za serwis lekarza.

5.42 SerwisLekarza.h

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

```
00001
00004 #pragma once
00005 #include "CzytaczLekarza.h"
00006 #include "Lekarz.h"
00007 #include <vector>
00008 #include "SerwisLogowaniaUzytkownika.h"
00009 #include <memory>
00010
00011 using namespace std;
00012
00017 class SerwisLekarza
00018 {
00019 private:
00020     shared_ptr<CzytaczLekarza> czytaczLekarza;
00021     shared_ptr<SerwisLogowaniaUzytkownika> serwisLogowaniaUzytkownika;
00022     const int indeksId = 0;
00023     const int indeksImienia = 1;
00024     const int indeksNazwiska = 2;
00025     const int indeksSpecjalizacji = 3;
00026 public:
00030     SerwisLekarza(shared_ptr<CzytaczLekarza> czytaczLekarza, shared_ptr<SerwisLogowaniaUzytkownika>
czytaczUzytkownikow);
00037     shared_ptr<Lekarz> zaladujLekarza(shared_ptr<Uzytkownik> uzytkownik);
00042     list<shared_ptr<Lekarz>> zaladujWszystkichLekarzy();
00049     shared_ptr<Lekarz> znajdzLekarzaPoId(int idLekarza, list<shared_ptr<Lekarz>> lekarze);
00050 };
00051
```

5.43 SerwisLogowaniaUzytkownika.h

```
00001
00004 #pragma once
00005 #include "Uzytkownik.h"
00006 #include "CzytaczUzytkownikow.h"
00007 #include <string>
00008 #include <memory>
00009
00010 using namespace std;
00011
00016 class SerwisLogowaniaUzytkownika {
00017 private:
00018     shared_ptr<CzytaczUzytkownikow> czytaczUzytkownikow;
00019
00020 public:
00024     SerwisLogowaniaUzytkownika(shared_ptr<CzytaczUzytkownikow> czytaczUzytkownikow);
00031     shared_ptr<Uzytkownik> zaloguj(string login, string haslo);
00037     shared_ptr<Uzytkownik> znajdz(string id);
00038 };
```

5.44 SerwisUslug.h File Reference

```
#include <vector>
#include "Usluga.h"
#include "Zwierze.h"
#include "Pies.h"
#include "Kot.h"
#include "Kanarek.h"
#include "Waz.h"
#include "Pajak.h"
#include <memory>
#include <list>
```

Classes

- class [SerwisUslug](#)

Klasa odpowiedzialna za serwis uslug.

5.45 SerwisUslug.h

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

```
00001
00004 #pragma once
00005
00006 #include <vector>
00007 #include "Usluga.h"
00008 #include "Zwierze.h"
00009 #include "Pies.h"
00010 #include "Kot.h"
00011 #include "Kanarek.h"
00012 #include "Waz.h"
00013 #include "Pajak.h"
00014 #include <memory>
00015 #include <list>
00016
00017 using namespace std;
00018
00023 class SerwisUslug
00024 {
00025 private:
00026     list<shared_ptr<Usluga>> dostepneUslugi;
00027 public:
00031     SerwisUslug();
00035     void zainicjalizujUslugi();
00041     shared_ptr<Usluga> znajdzPoTypie(string typ);
00047     vector<shared_ptr<Usluga>> stworzUsluguDlaZwierzaka(shared_ptr<Zwierze> zwierze);
00048 };
00049
```

5.46 SerwisWizyt.h File Reference

```
#include "StatusWizyty.h"
#include "Wizyta.h"
#include "CzytaczWizyt.h"
#include <list>
#include <memory>
```


Classes

- class [SerwisWizyt](#)

Klasa odpowiedzialna za serwis wizyt.

5.47 SerwisWizyt.h

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

```
00001
00004 #pragma once
00005
00006 #include "StatusWizyty.h"
00007 #include "Wizyta.h"
00008 #include "CzytaczWizyt.h"
00009 #include<list>
00010 #include <list>
00011 #include <memory>
00012
00013 using namespace std;
00014
00019 class SerwisWizyt {
00020
00021 private:
00022     shared_ptr<CzytaczWizyt> czytaczWizyt;
00023
00024     const int idWizytyIndeks = 0;
00025     const int idLekarzaIndeks = 1;
00026     const int idWlasicielaZwierzeciaIndeks = 2;
00027     const int idZwierzeciaIndeks = 3;
00028     const int godzinaWizytyIndeks = 4;
00029     const int dzienWizytyIndeks = 5;
00030     const int miesiacWizytyIndeks = 6;
00031     const int rokWizytyIndeks = 7;
00032     const int typUslugiIndeks = 8;
00033 public:
00037     SerwisWizyt(shared_ptr<CzytaczWizyt> czytaczWizyt);
00043     StatusWizyty zarezerwujWizyte(Wizyta wizyta);
00049     StatusWizyty anulujWizyte(int idWizyty);
00055     StatusWizyty anulujWizyteDlaZwierzaka(int idZwierzaka);
00061     list<shared_ptr<Wizyta>> przegladWizytPoIdWlasiciela(int idWlasicielZwierzecia);
00069     list<shared_ptr<Wizyta>> zaladujWizytyDlaDaty(int rok, int miesiac, int dzien);
00075     list<shared_ptr<Wizyta>> przegladWizytPoIdLekarza(int idLekarza);
00080     int getNextId();
00081 };
```

5.48 SerwisWlasicielaZwierzecia.h

```
00001
00004 #pragma once
00005 #include "CzytaczWlasicielaZwierzecia.h"
00006 #include "SerwisLogowaniaUzytkownika.h"
00007 #include "WlasicielZwierzecia.h"
00008
00013 class SerwisWlasicielaZwierzecia
00014 {
00015 private:
00016     std::shared_ptr<CzytaczWlasicielaZwierzecia> czytaczWlasicielaZwierzecia;
00017     std::shared_ptr<SerwisLogowaniaUzytkownika> serwisLogowaniaUzytkownika;
00018     const int indeksImienia = 0;
00019     const int indeksId = 1;
00020     const int indeksMiasta = 2;
00021     const int indeksKoduPocztowego = 3;
00022     const int indeksUlicyINumeruDomu = 4;
00023     const int indeksNumeruTelefonu = 5;
00024 public:
00028     SerwisWlasicielaZwierzecia(std::shared_ptr<CzytaczWlasicielaZwierzecia>
czytaczWlasicielaZwierzecia, std::shared_ptr<SerwisLogowaniaUzytkownika> serwisLogowaniaUzytkownika);
00035     std::shared_ptr<WlasicielZwierzecia> zaladujWlasicielaZwierzecia(std::shared_ptr<Uzytkownik>
uzytkownik);
00041     std::shared_ptr<WlasicielZwierzecia> zaladujWlasicielaZwierzeciaPoId(int id);
00042
00043 };
00044
```

5.49 SerwisZwierzat.h File Reference

```
#include "CzytaczZwierzat.h"
#include "Zwierze.h"
#include <vector>
#include <memory>
```

Classes

- class [SerwisZwierzat](#)
Klasa odpowiedzialna za serwis zwierzat.

5.50 SerwisZwierzat.h

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

```
00001
00004 #pragma once
00005 #include "CzytaczZwierzat.h"
00006 #include "Zwierze.h"
00007 #include <vector>
00008 #include <memory>
00009
00010 using namespace std;
00011
00016 class SerwisZwierzat
00017 {
00018 private:
00019     std::shared_ptr<CzytaczZwierzat> czytaczZwierzat;
00020     const int indeksId = 0;
00021     const int indeksImienia = 1;
00022     const int indeksPlci = 2;
00023     const int indeksWieku = 3;
00024     const int indeksIlosciNog = 4;
00025     const int indeksIdWlasciciela = 5;
00026     const int indeksUmaszczenia = 6;
00027     const int indeksKoloru = 7;
00028     const int indeksTypu = 8;
00029     const int indeksWagi = 9;
00030 public:
00034     SerwisZwierzat(shared_ptr<CzytaczZwierzat> czytaczZwierzat);
00040     vector<shared_ptr<Zwierze>> zaladujZwierzetaDlaWlasciciela(int idWlasciciela);
00047     shared_ptr<Zwierze> znajdzPoId(int idZwierzecia, vector<shared_ptr<Zwierze>> zwierzeta);
00053     shared_ptr<Zwierze> zaladujZwierzetaPoId(int idZwierzecia);
00058     int getNextPoId();
00063     void dodaj(shared_ptr<Zwierze> zwierzeta);
00068     void usunZwierzaka(int idZwierzaka);
00073     void zaktualizuj(shared_ptr<Zwierze> zwierzeta);
00074 };
00075
```

5.51 SesjaLekarza.h File Reference

```
#include "SerwisWizyt.h"
#include "Lekarz.h"
#include "SerwisZwierzat.h"
#include "SerwisUslug.h"
#include "SerwisWlascicielaZwierzecia.h"
#include <memory>
```

Classes

- class [SesjaLekarza](#)
Klasa odpowiedzialna za sesje lekarza (za jego wizyty)

5.52 SesjaLekarza.h

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

```

00001
00004 #pragma once
00005 #include "SerwisWizyt.h"
00006 #include "Lekarz.h"
00007 #include "SerwisZwierzat.h"
00008 #include "SerwisUslug.h"
00009 #include "SerwisWlasicielaZwierzecia.h"
00010 #include <memory>
00011
00012 using namespace std;
00013
00018 class SesjaLekarza
00019 {
00020 private:
00021     shared_ptr<Lekarz> lekarz;
00022     shared_ptr<SerwisWizyt> serwisWizyt;
00023     shared_ptr<SerwisZwierzat> serwisZwierzat;
00024     shared_ptr<SerwisUslug> serwisUslug;
00025     shared_ptr<SerwisWlasicielaZwierzecia> serwisWlasicielaZwierzecia;
00026 public:
00030     SesjaLekarza(shared_ptr<Lekarz> lekarz, shared_ptr<SerwisWizyt> serwisWizyt,
shared_ptr<SerwisZwierzat> serwisZwierzat, shared_ptr<SerwisUslug> serwisUslug,
shared_ptr<SerwisWlasicielaZwierzecia> serwisWlasicielaZwierzecia);
00036     shared_ptr<Lekarz> getLekarz();
00041     shared_ptr<SerwisWizyt> getSerwisWizyt();
00046     shared_ptr<SerwisZwierzat> getSerwisZwierzat();
00051     shared_ptr<SerwisUslug> getFabrykaUslug();
00056     shared_ptr<SerwisWlasicielaZwierzecia> getSerwisWlasicielaZwierzecia();
00057 };
00058

```

5.53 SesjaWlasicielaZwierzecia.h File Reference

```

#include "Zwierze.h"
#include "Usluga.h"
#include <vector>
#include "WlasicielZwierzecia.h"
#include <memory>

```

Classes

- class [SesjaWlasicielaZwierzecia](#)
Klasa odpowiedzialna za sesje wlasciciela zwierzecia.

5.54 SesjaWlascicielaZwierzecia.h

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

```
00001
00004 #pragma once
00005 #include "Zwierze.h"
00006 #include "Usługa.h"
00007 #include <vector>
00008 #include "WlascicielZwierzecia.h"
00009 #include <memory>
00010
00011 using namespace std;
00012
00017 class SesjaWlascicielaZwierzecia
00018 {
00019 private:
00020     shared_ptr<WlascicielZwierzecia> wlascicielZwierzecia;
00021     shared_ptr<Zwierze> wybraneZwierze;
00022     vector<shared_ptr<Zwierze>> zwierzetaUztkownika;
00023     shared_ptr<Usługa> wybranaUsługa;
00024     shared_ptr<pair<int, time_t>> wybranyDzien;
00025     shared_ptr<pair<int, time_t>> wybranyMiesiac;
00026     shared_ptr<pair<int, time_t>> wybranyRok;
00027     int wybranaGodzina;
00028     int idWybranegoLekarza;
00029
00030 public:
00031
00035     SesjaWlascicielaZwierzecia(shared_ptr<WlascicielZwierzecia> wlascicielZwierzecia,
vector<shared_ptr<Zwierze>> zwierzeta);
00040     vector<shared_ptr<Zwierze>> getZwierzeta();
00045     shared_ptr<Zwierze> getWybarneZwierze();
00050     shared_ptr<WlascicielZwierzecia> getWlascicielZwierzecia();
00055     void setWybraneZwierze(shared_ptr<Zwierze> zwierzeta);
00060     void setWybranaUsługa(shared_ptr<Usługa> wybranaUsługa);
00065     shared_ptr<Usługa> getWybranaUsługa();
00070     void setWybranyMiesiac(shared_ptr<pair<int, time_t>> wybranyMiesiac);
00075     shared_ptr<pair<int, time_t>> getWybranyMiesiac();
00080     void setWybranyDzien(shared_ptr<pair<int, time_t>> wybranyDzien);
00085     shared_ptr<pair<int, time_t>> getWybranyDzien();
00090     void setWybranyRok(shared_ptr<pair<int, time_t>> wybranyRok);
00095     shared_ptr<pair<int, time_t>> getWybranyRok();
00100     void setWybranaGodzina(int wybranaGodzina);
00105     int getWybranaGodzina();
00110     void setIdWybranegoLekarza(int idLekarza);
00115     int getIdWybranegoLekarza();
00120     bool czyMoznaUtworzycWizyte();
00125     void odswiezZwierzeta(vector<shared_ptr<Zwierze>> zwierzeta);
00126 };
```

5.55 SprawdzaczLiczb.h File Reference

```
#include <string>
```

Classes

- class [SprawdzaczLiczb](#)

Klasa odpowiedzialna za sprawdzenie czy to co jest wprowadzane jest liczba.

5.56 SprawdzaczLiczb.h

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

```
00001
00004 #pragma once
00005 #include <string>
00006 using namespace std;
00007
```

```
00012 class SprawdzaczLiczba
00013 {
00014
00015 public:
00021     bool jestLiczba(const string& tekst);
00022 };
00023
```

5.57 StatusWizyty.h

```
00001
00004 #pragma once
00005
00010 enum StatusWizyty { POTWIERDZONA, NIEPOTWIERDZONA };
```

5.58 TypUzytkownika.h File Reference

Enumerations

- enum [TypUzytkownika](#) { LEKARZ , WLASCICIEL_ZWIERZECIA , NIEZNANY_TYP }
Typ enum reprezentujący 2 typy użytkownika (lekarz, właściciel zwierzęcia)

5.59 TypUzytkownika.h

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

```
00001
00004 #pragma once
00005
00010 enum TypUzytkownika
00011 {
00012     LEKARZ, WLASCICIEL_ZWIERZECIA, NIEZNANY_TYP
00013 };
```

5.60 Usługa.h File Reference

```
#include <string>
#include <vector>
```

Classes

- class [Usługa](#)
Klasa odpowiedzialna za usługi.

5.61 Usluga.h

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

```
00001
00004 #pragma once
00005 #include <string>
00006 #include <vector>
00007
00008 using namespace std;
00009
00014 class Usluga {
00015 private:
00016     int cena;
00017     string nazwaUslugi;
00018     string typUslugi;
00019
00020 public:
00024     Usluga(int cena, string nazwaUslugi, string typUslugi);
00029     int getCena();
00034     string getTypUslugi();
00039     string getNazwaUslugi();
00040 };
```

5.62 Uzytkownik.h File Reference

```
#include <string>
#include "TypUzytkownika.h"
```

Classes

- class [Uzytkownik](#)

Klasa odpowiedzialna za uzytkownikow.

5.63 Uzytkownik.h

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

```
00001
00004 #pragma once
00005 #include <string>
00006 #include "TypUzytkownika.h"
00007
00008 using namespace std;
00009
00014 class Uzytkownik {
00015 private:
00016     string login;
00017     string haslo;
00018     int id;
00019     TypUzytkownika typUzytkownika;
00020
00021 public:
00025     Uzytkownik(string login, string haslo, int id, TypUzytkownika typUzytkownika);
00030     string zamienNaTekst();
00035     int getId();
00040     TypUzytkownika getTypUzytkownika();
00044     static const int ID_DLA_NIEZNANEGO_UZYTKOWNIKA = -1;
00045 };
```

5.64 Waz.h File Reference

```
#include "Zwierze.h"
```

Classes

- class [Waz](#)

Klasa opisująca weza, klasa pochodna. Dziedziczy po klasie [Zwierze](#).

5.65 Waz.h

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

```
00001
00004 #pragma once
00005 #include "Zwierze.h"
00006
00011 class Waz : public Zwierze {
00012
00013 public:
00017     Waz(int idZwierzecia, int wiek, Plec plec, int liczbaNog, string imie, int idWlasciciela, string
        umaszczzenie, string kolor, int wagaWGramach);
00022     string dzwiek();
00027     string getWaga();
00028 };
```

5.66 Wizyta.h File Reference

```
#include <string>
#include "GodzinyPracy.h"
```

Classes

- class [DataGodzinaWizyty](#)

Klasa odpowiedzialna za date i godzinę wizyty.

- class [Wizyta](#)

Klasa odpowiedzialna wizyte.

5.67 Wizyta.h

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

```
00001
00004 #pragma once
00005 #include <string>
00006 #include "GodzinyPracy.h"
00007 #include <string>
00008
00009 using namespace std;
00010
00015 class DataGodzinaWizyty {
00016 private:
00017     int dzien;
00018     int miesiac;
00019     int rok;
00020     int godzina;
00021
00022 public:
00023
00027     DataGodzinaWizyty(int godzina, int dzien, int miesiac, int rok);
00031     DataGodzinaWizyty();
00036     int getDzien();
00041     int getMiesiac();
00046     int getRok();
00051     int getGodzina();
```

```

00052 };
00053
00058 class Wizyta {
00059 private:
00060     int idWizyty;
00061     DataGodzinaWizyty dataWizyty;
00062     int idWlasciciela;
00063     int idLekarza;
00064     int idZwierzecia;
00065     string typUslugi;
00066
00067 public:
00071     Wizyta(DataGodzinaWizyty dataWizty, int idWizyty, int idWlasciciela, int idZwierzecia, int
idLekarza, string typUslugi);
00076     string zamieNaTekst();
00081     DataGodzinaWizyty getDataGodzina();
00086     int getIdWizyty();
00091     int getIdWlasciciela();
00096     int getIdLekarza();
00101     int getIdZwierzecia();
00106     string getTypUslugi();
00111     string getData();
00116     string getGodzina();
00117 };

```

5.68 WlascicielZwierzecia.h File Reference

```

#include "Uzytkownik.h"
#include <string>

```

Classes

- class [WlascicielZwierzecia](#)
Klasa odpowiedzialna za wlasciciela zwierzecia.

5.69 WlascicielZwierzecia.h

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

```

00001
00004 #pragma once
00005 #include "Uzytkownik.h"
00006 #include <string>
00007
00008 using namespace std;
00009
00014 class WlascicielZwierzecia : public Uzytkownik {
00015
00016 private:
00017     string imie;
00018     int id;
00019     string miasto;
00020     string kodPocztowy;
00021     string ulicaINumerDomu;
00022     string numerTelefonu;
00023
00024
00025 public:
00029     WlascicielZwierzecia(string imie, int id, string miasto, string kodPocztowy, string
ulicaINumerDomu, string numerTelefonu, Uzytkownik& uzytkownik);
00034     string getImie();
00039     int getId();
00044     string getMiasto();
00049     string getKodPocztowy();
00054     string getUlicaINumerDomu();
00059     string getNumerTelefonu();
00064     void setMiasto(string miasto);
00069     void setKodPocztowy(string kodPocztowy);
00074     void setUlicaINumerDomu(string ulicaINumerDomu);
00079     void setNumerTelefonu(string numerTelefonu);
00080 };

```


5.70 Zwierze.h File Reference

```
#include <string>
#include "Plec.h"
```

Classes

- class [Zwierze](#)

Klasa odpowiedzialna za zwierzeta.

5.71 Zwierze.h

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

```
00001
00004 #pragma once
00005 #include <string>
00006 #include "Plec.h"
00007
00008 using namespace std;
00009
00014 class Zwierze {
00015
00016 private:
00017     int id;
00018     int wiek;
00019     Plec plec;
00020     int liczbaNog;
00021     string imie;
00022     int idWlasciciela;
00023     string umaszczenie;
00024     string kolor;
00025
00026 protected:
00031     int wagaWGramach;
00032
00033 public:
00034
00038     Zwierze(int id, int wiek, Plec plec, int liczbaNog, string imie, int idWlasciciela, string
umaszczenie, string kolor, int wagaWGramach);
00043     int getId();
00048     int getWiek();
00053     Plec getPlec();
00058     int getLiczbaNog();
00063     string getImie();
00068     int getIdWlasciciela();
00073     string getUmaszczenie();
00078     string getKolor();
00084     virtual string dzwiek() = 0;
00090     virtual string getWaga() = 0;
00095     void setWagaWGramach(int wagaWGramach);
00100     void setImie(string imie);
00105     string zamienNaTekst();
00106 };
00107
```


Index

anulujWizyte
 SerwisWizyt, [55](#)
anulujWizyteDlaZwierzaka
 SerwisWizyt, [55](#)

czyDostepnyWGodzinach
 Lekarz, [26](#)
czyJestDniem
 PlanerWizyt, [37](#)
czyMoznaUtworzycWizyte
 SesjaWlascicielaZwierzecia, [64](#)
czyOstatniDostepnyMiesiac
 PlanerWizyt, [37](#)
czyPierwszDostepnyMiesiac
 PlanerWizyt, [38](#)
CzytaczGrafikuPracy, [7](#)
CzytaczGrafikuPracy.h, [87](#)
CzytaczLekarza, [8](#)
CzytaczLekarza.h, [87](#)
CzytaczPlikow, [9](#)
 dopiszDoPliku, [10](#)
 getNastepneld, [10](#)
 getPozycjaTextu, [10](#)
 getPozycjaWiersza, [11](#)
 podajNazwePliku, [11](#)
 usunWieszPold, [11](#)
 wczytajPlik, [11](#)
 zapiszDoPliku, [12](#)
 zmaiienNaWiersze, [12](#)
CzytaczPlikow.h, [88](#)
CzytaczUzytkownikow, [12](#)
 podajNazwePliku, [13](#)
CzytaczUzytkownikow.h, [89](#)
CzytaczWizyt, [14](#)
 podajNazwePliku, [15](#)
CzytaczWizyt.h, [89](#)
CzytaczWlascicielaZwierzecia, [15](#)
CzytaczZwierzat, [16](#)
 podajNazwePliku, [17](#)
CzytaczZwierzat.h, [90](#)

DataGodzinaWizyty, [18](#)
 getDzien, [18](#)
 getGodzina, [18](#)
 getMiesiac, [19](#)
 getRok, [19](#)
DniTygodnia.h, [90](#)
dodaj
 SerwisZwierzat, [59](#)
dodajDzien
 PlanerWizyt, [38](#)
dodajGodzine
 PlanerWizyt, [38](#)
dodajGodzinePlusZaokraglenie
 PlanerWizyt, [39](#)
dodajMiesiac
 PlanerWizyt, [39](#)
dopiszDoPliku
 CzytaczPlikow, [10](#)
dzwiek
 Kanarek, [23](#)
 Kot, [25](#)
 Pajak, [33](#)
 Pies, [35](#)
 Waz, [73](#)
 Zwierz, [82](#)

getCena
 Usługa, [69](#)
getData
 Wizyta, [74](#)
getDataGodzina
 Wizyta, [74](#)
getDzien
 DataGodzinaWizyty, [18](#)
getFabrykaUsług
 SesjaLekarza, [62](#)
getGodzina
 DataGodzinaWizyty, [18](#)
 Wizyta, [74](#)
getGodzinaRozpoczenia
 GodzinyPracy, [20](#)
getGodzinaZakonczenia
 GodzinyPracy, [20](#)
getGodzinaZamknieniaWDniu
 PlanerWizyt, [40](#)
getHaslo
 MenuLogowania, [29](#)
getId
 Uzytkownik, [71](#)
 WlascicielZwierzecia, [77](#)
 Zwierz, [82](#)
getIdLekarza
 Wizyta, [74](#)
getIdWizyty
 Wizyta, [74](#)
getIdWlasciciela
 Wizyta, [75](#)
 Zwierz, [82](#)
getIdWybranegoLekarza

SesjaWlascicielaZwierzecia, 64
getIdZwierzecia
Wizyta, 75
getImie
Lekarz, 26
WlascicielZwierzecia, 77
Zwierze, 82
getKodPoczowy
WlascicielZwierzecia, 77
getKolor
Zwierze, 82
getLekarz
SesjaLekarza, 62
getliczaNog
Zwierze, 83
getLogin
MenuLogowania, 29
getMiasto
WlascicielZwierzecia, 77
getMiesiac
DataGodzinaWizyty, 19
PlanerWizyt, 40
getMinutaRozpoczecia
GodzinyPracy, 20
getMinutaZakonczenia
GodzinyPracy, 20
getNastepnelid
CzytaczPlikow, 10
SerwisWizyt, 55
SerwisZwierzat, 59
getNazwaUslugi
Usługa, 69
getNazwisko
Lekarz, 26
getNumerTelefonu
WlascicielZwierzecia, 78
getPlec
Zwierze, 83
getPozycjaTextu
CzytaczPlikow, 10
getPozycjaWiersza
CzytaczPlikow, 11
getRok
DataGodzinaWizyty, 19
getSerwisWizyt
SesjaLekarza, 62
getSerwisWlascicielaZwierzecia
SesjaLekarza, 62
getSerwisZwierzat
SesjaLekarza, 62
getSpecjalizacja
Lekarz, 27
getTypUslugi
Usługa, 69
Wizyta, 75
getTypUzytkownika
Uzytkownik, 71
getUlicaINumerDomu
WlascicielZwierzecia, 78
getUmaszczenie
Zwierze, 83
getWaga
Kanarek, 23
Kot, 25
Pajak, 33
Pies, 35
Waz, 73
Zwierze, 83
getWiek
Zwierze, 83
getWlascicielZwierzecia
SesjaWlascicielaZwierzecia, 64
getWybarneZwierze
SesjaWlascicielaZwierzecia, 64
getWybranaGodzina
SesjaWlascicielaZwierzecia, 65
getWybranaUsługa
SesjaWlascicielaZwierzecia, 65
getWybranyDzien
SesjaWlascicielaZwierzecia, 65
getWybranyMiesiac
SesjaWlascicielaZwierzecia, 65
getWybranyRok
SesjaWlascicielaZwierzecia, 65
getZwierzeta
SesjaWlascicielaZwierzecia, 66
GodzinyPracy, 19
getGodzinaRozpoczecia, 20
getGodzinaZakonczenia, 20
getMinutaRozpoczecia, 20
getMinutaZakonczenia, 20
GodzinyPracy.h, 91
jestLiczba
SprawdzaczLiczby, 68
Kanarek, 21
dzwiek, 23
getWaga, 23
Kanarek.h, 91
Kot, 23
dzwiek, 25
getWaga, 25
Kot.h, 92
Lekarz, 25
czyDostepnyWGodzinach, 26
getImie, 26
getNazwisko, 26
getSpecjalizacja, 27
zwrocGodzinyPracyDlaDnia, 27
Lekarz.h, 92
MenuLekarza, 27
wyswietlMenuPocatkowe, 28
MenuLekarza.h, 93
MenuLogowania, 28

- getHaslo, 29
- getLogin, 29
- wyswietlMenuPocztkowe, 29
- MenuLogowania.h, 94
- MenuLogowaniaStatusu.h, 94
- MenuWlasicielaZwierzecia, 29
 - wswietlMenuUslug, 30
 - wyswietlMenuDoPlanowaniaWizyty, 30
 - wyswietlMenuPocztkowe, 30
 - wyswietlMenuWyboruTerminu, 31
- ostatniDzienMiesiaca
 - PlanerWizyt, 40
- Pajak, 31
 - dzwiek, 33
 - getWaga, 33
- Pajak.h, 95
- Pies, 33
 - dzwiek, 35
 - getWaga, 35
- Pies.h, 96
- PlanerWizyt, 35
 - czyJestDniem, 37
 - czyOstatniDostepnyMiesiac, 37
 - czyPierwszDostepnyMiesiac, 38
 - dodajDzien, 38
 - dodajGodzine, 38
 - dodajGodzinePlusZaokraglenie, 39
 - dodajMiesiac, 39
 - getGodzinaZamknieniaWDniu, 40
 - getMiesiac, 40
 - ostatniDzienMiesiaca, 40
 - ustawCzas, 41
 - utworzWizyte, 41
 - wygenerujBlokiGodzinowe, 41
 - wygenerujDni, 42
 - wygenerujIdWizyty, 42
 - wygenerujMiesiace, 42
 - wypiszKalendarz, 43
 - wyswietl, 43
 - zaladujAktualnyDzienTygodnia, 43
 - zaladujDostepnaGodzineNaDzienTygodnia, 44
 - zaladujDostepyCzas, 44
 - zaladujDzienMiesiaca, 46
 - zaladujMiesiac, 46
 - zaladujNastepnyDzienTygodnia, 46
 - zaladujOstatniaDostepnaDate, 47
 - zaladujPierwszaDostepnaDate, 47
 - zaladujRok, 47
 - zamienNaDniTygonida, 48
 - zamienNaMiesiac, 48
 - zamienZAngielskiegoNaPolski, 48
 - znajdzPierwszyDostepnyTermin, 49
- PlanerWizyt.h, 96
- Plec.h, 98
- podajNazwePliku
 - CzytaczPlikow, 11
 - CzytaczUzytkownikow, 13
 - CzytaczWizyt, 15
 - CzytaczZwierzat, 17
- przegladWizytPoldLekarza
 - SerwisWizyt, 55
- przegladWizytPoldWlasiciela
 - SerwisWizyt, 56
- SerwisGrafikuPracy, 49
 - zaladujGrafik, 50
- SerwisGrafikuPracy.h, 98
- SerwisLekarza, 50
 - zaladujLekarza, 51
 - zaladujWszystkichLekarzy, 51
 - znajdzLekarzaPold, 51
- SerwisLekarza.h, 99
- SerwisLogowaniaUzytkownika, 52
 - zaloguj, 52
 - znajdz, 52
- SerwisUslug, 53
 - stworzUslugeDlaZwierzaka, 53
 - znajdzPoTypie, 54
- SerwisUslug.h, 100
- SerwisWizyt, 54
 - anulujWizyte, 55
 - anulujWizyteDlaZwierzaka, 55
 - getNastepneld, 55
 - przegladWizytPoldLekarza, 55
 - przegladWizytPoldWlasiciela, 56
 - zaladujWizytyDlaDaty, 56
 - zarezerujWizyte, 57
- SerwisWizyt.h, 100
- SerwisWlasicielaZwierzecia, 57
 - zaladujWlasicielaZwierzecia, 58
 - zaladujWlasicielaZwierzeciaPold, 58
- SerwisZwierzat, 58
 - dodaj, 59
 - getNastepneld, 59
 - usunZwierzaka, 59
 - zaktualizuj, 60
 - zaladujZwierzePold, 60
 - zaladujZwierzetaDlaWlasiciela, 60
 - znajdzPold, 61
- SerwisZwierzat.h, 102
- SesjaLekarza, 61
 - getFabrykaUslug, 62
 - getLekarz, 62
 - getSerwisWizyt, 62
 - getSerwisWlasicielaZwierzecia, 62
 - getSerwisZwierzat, 62
- SesjaLekarza.h, 102
- SesjaWlasicielaZwierzecia, 63
 - czyMoznaUtworzycWizyte, 64
 - getIdWybranegoLekarza, 64
 - getWlasicielZwierzecia, 64
 - getWybarneZwierze, 64
 - getWybranaGodzina, 65
 - getWybranaUsluga, 65
 - getWybranyDzien, 65
 - getWybranyMiesiac, 65

- getWybranyRok, 65
- getZwierzeta, 66
- setIdWybranegoLekarza, 66
- setWybranaGodzina, 66
- setWybranaUsluga, 66
- setWybraneZwierze, 67
- setWybranyDzien, 67
- setWybranyMiesiac, 67
- setWybranyRok, 67
- SesjaWlascicielaZwierzecia.h, 103
- setIdWybranegoLekarza
 - SesjaWlascicielaZwierzecia, 66
- setImie
 - Zwierze, 84
- setKodPocztowy
 - WlascicielZwierzecia, 78
- setMiasto
 - WlascicielZwierzecia, 78
- setNumerTelefonu
 - WlascicielZwierzecia, 80
- setUlicaINumerDomu
 - WlascicielZwierzecia, 80
- setWagaWGramach
 - Zwierze, 84
- setWybranaGodzina
 - SesjaWlascicielaZwierzecia, 66
- setWybranaUsluga
 - SesjaWlascicielaZwierzecia, 66
- setWybraneZwierze
 - SesjaWlascicielaZwierzecia, 67
- setWybranyDzien
 - SesjaWlascicielaZwierzecia, 67
- setWybranyMiesiac
 - SesjaWlascicielaZwierzecia, 67
- setWybranyRok
 - SesjaWlascicielaZwierzecia, 67
- SprawdzaczLiczba, 68
 - jestLiczba, 68
- SprawdzaczLiczba.h, 104
- stworzUslugeDlaZwierzaka
 - SerwisUslug, 53
- TypUzytkownika.h, 105
- Usluga, 69
 - getCena, 69
 - getNazwaUslugi, 69
 - getTypUslugi, 69
- Usluga.h, 105
- ustawCzas
 - PlanerWizyt, 41
- usunWieszPold
 - CzytaczPlikow, 11
- usunZwierzaka
 - SerwisZwierzat, 59
- utworzWizyte
 - PlanerWizyt, 41
- Uzytkownik, 70
 - getId, 71
 - getTypUzytkownika, 71
 - zamienNaTekst, 71
- Uzytkownik.h, 106
- wagaWGramach
 - Zwierze, 84
- Waz, 71
 - dzwiek, 73
 - getWaga, 73
- Waz.h, 106
- wczytajPlik
 - CzytaczPlikow, 11
- Wizyta, 73
 - getData, 74
 - getDataGodzina, 74
 - getGodzina, 74
 - getIdLekarza, 74
 - getIdWizyty, 74
 - getIdWlasciciela, 75
 - getIdZwierzecia, 75
 - getTypUslugi, 75
 - zamieNaTekst, 75
- Wizyta.h, 107
- WlascicielZwierzecia, 76
 - getId, 77
 - getImie, 77
 - getKodPoczowy, 77
 - getMiasto, 77
 - getNumerTelefonu, 78
 - getUlicaINumerDomu, 78
 - setKodPocztowy, 78
 - setMiasto, 78
 - setNumerTelefonu, 80
 - setUlicaINumerDomu, 80
- WlascicielZwierzecia.h, 108
- wswietlMenuUslug
 - MenuWlascicielaZwierzecia, 30
- wygenerujBlokGodzinowe
 - PlanerWizyt, 41
- wygenerujDni
 - PlanerWizyt, 42
- wygenerujIdWizyty
 - PlanerWizyt, 42
- wygenerujMiesiace
 - PlanerWizyt, 42
- wypiszKalendarz
 - PlanerWizyt, 43
- wyswietl
 - PlanerWizyt, 43
- wyswietlMenuDoPlanowaniaWizyty
 - MenuWlascicielaZwierzecia, 30
- wyswietlMenuPoczkowe
 - MenuLekarza, 28
 - MenuLogowania, 29
 - MenuWlascicielaZwierzecia, 30
- wyswietlMenuWyboruTerminu
 - MenuWlascicielaZwierzecia, 31
- zaktualizuj

SerwisZwierzat, 60
zaladujAktualnyDzienTygodnia
 PlanerWizyt, 43
zaladujDostepnaGodzineNaDzienTygodnia
 PlanerWizyt, 44
zaladujDostepyCzas
 PlanerWizyt, 44
zaladujDzienMiesiaca
 PlanerWizyt, 46
zaladujGrafik
 SerwisGrafikuPracy, 50
zaladujLekarza
 SerwisLekarza, 51
zaladujMiesiac
 PlanerWizyt, 46
zaladujNastepnyDzienTygodnia
 PlanerWizyt, 46
zaladujOstatniaDostepnaDate
 PlanerWizyt, 47
zaladujPierszaDostepnaDate
 PlanerWizyt, 47
zaladujRok
 PlanerWizyt, 47
zaladujWizytyDlaDaty
 SerwisWizyt, 56
zaladujWlascicielaZwierzecia
 SerwisWlascicielaZwierzecia, 58
zaladujWlascicielaZwierzeciaPold
 SerwisWlascicielaZwierzecia, 58
zaladujWszystkichLekarzy
 SerwisLekarza, 51
zaladujZwierzePold
 SerwisZwierzat, 60
zaladujZwierzetaDlaWlasciciela
 SerwisZwierzat, 60
zaloguj
 SerwisLogowaniaUzytkownika, 52
zamienNaTekst
 Wizyta, 75
zamienNaDniTygonida
 PlanerWizyt, 48
zamienNaMiesiac
 PlanerWizyt, 48
zamienNaTekst
 Uzytkownik, 71
 Zwierze, 84
zamienZAngielskiegoNaPolski
 PlanerWizyt, 48
zapiszDoPliku
 CzytaczPlikow, 12
zarezerujWizyte
 SerwisWizyt, 57
zmaiienNaWiersze
 CzytaczPlikow, 12
znajdz
 SerwisLogowaniaUzytkownika, 52
znajdzLekarzaPold
 SerwisLekarza, 51
znajdzPierwszyDostepnyTermin
 PlanerWizyt, 49
znajdzPold
 SerwisZwierzat, 61
znajdzPoTypie
 SerwisUslug, 54
Zwierze, 80
 dzwiek, 82
 getId, 82
 getIdWlasciciela, 82
 getImie, 82
 getKolor, 82
 getIcizaNog, 83
 getPlec, 83
 getUmaszczenie, 83
 getWaga, 83
 getWiek, 83
 setImie, 84
 setWagaWGramach, 84
 wagaWGramach, 84
 zamienNaTekst, 84
Zwierze.h, 109
zwrocGodzinyPracyDlaDnia
 Lekarz, 27