

Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki
W Gliwicach

Programowanie Komputerów

2

Program do rejestracji pacjentów w przychodni

Autor	Agnieszka Góral
Prowadzący	dr inż. Wojciech Łabaj
Rok akademicki	2020/2021
Kierunek	Informatyka
Rodzaj studiów	SSI
Semestr	2
Termin laboratoriów	Poniedziałek 13:45-15.15 Piątek 13.30-15.00
Sekcja	2
Grupa	2

SYSTEM REJESTRACJI W PRZYCHODNI

1. OPIS DZIAŁANIA PROGRAMU (skrótowa analiza programu)

Program ma za zadanie umożliwić pacjentom rejestrację do lekarzy na wybrane godziny, uwzględniając czas pracy konkretnego lekarza. Najpierw pacjent musi się zalogować za pomocą numeru PESEL oraz hasła. Jeśli jeszcze nie ma konta to program poprosi go o podanie danych (imię, nazwisko, data urodzenia, telefon kontaktowy, pesel) i założyć konto.

Z poziomu menu (wyświetlanego zaraz po zalogowaniu) pacjent ma 4 opcje:

1. Zapisanie się na wizytę

-> wybór lekarza (przez wyszukiwanie po specjalizacji)

->możliwość wybrania daty

->możliwość sprawdzenia dnia i już umówionych wizyt oraz wybranie godziny dla siebie

2. Przeglądanie grafików lekarzy

-> wybór lekarza (przez wyszukiwanie po specjalizacji)

3. Sprawdzenie konta pacjenta (dane)

3. Wyświetlenie dat zaplanowanych wizyt i tych, które już się odbyły, możliwość odwołania wizyty

4. Wylogowanie się

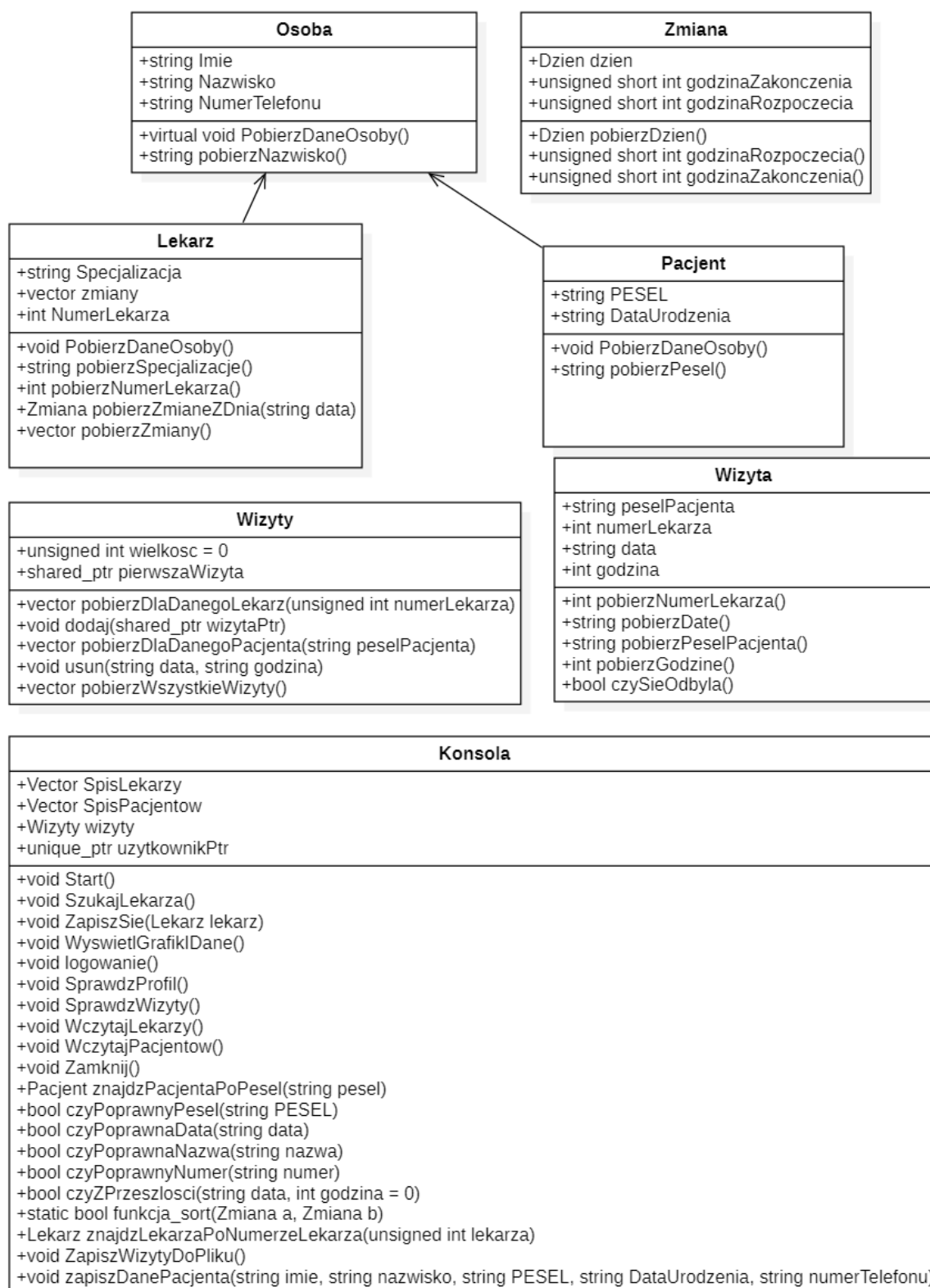
5. Zamknięcie programu

Pacjent może wybrać lekarza, u którego jest zainteresowany wizytą za pomocą wyszukiwarki specjalistów.

Po wybraniu lekarza oraz daty wyświetlany jest jego grafik na zadany dzień oraz dostępne godziny. Pacjent wybiera odpowiednią godzinę oraz zatwierdza wizytę.

Godzina, na którą zapisał się pacjent zostaje oznaczona jako zajęta i inni pacjenci już nie będą mogli się na nią zapisać. Po powrocie do menu, pacjent będzie mógł sprawdzić, czy data zaplanowanej wizyty znajduje się na jego koncie.

2. OPIS KLAS I STRUKTUR DANYCH



OSOBA (class Osoba{};)

Klasa opisująca osobę, podstawowe informacje personalne każdego bywalca przychodni. Zawiera zmienne prywatne: typu string - imie, nazwisko oraz numerTelefonu.

Publiczne metody klasy:

- **virtual void pobierzDaneOsoby()** – metoda zwracająca w stringu „sklejone” dane osoby wraz z opisami.
- **String pobierzNazwisko()** – metoda zwracająca prywatną zmienną klasy - nazwisko

PACJENT (class Pacjent{};)

Pacjent dziedziczy z klasy Osoba zmienne takie jak imie, nazwisko oraz numerTelefonu. Klasa ta posiada zmienne prywatne DataUrodzenia (typ string) oraz pesel (typ string) - obie zmienne przechowują informacje o pacjencie wg nazw zmiennych - datę urodzenia oraz numer PESEL oraz zmienną Wizyty, typu wizyty (przechowuje ona wizyty, na które umówił się pacjent).

Publiczne metody klasy:

- **string PobierzDaneOsoby()** - pozwalająca na pobranie danych z prywatnych zmiennych w klasie (tj. imienia, nazwiska, numeru telefonu, PESELu oraz daty urodzenia pacjenta), zwraca string z tymi danymi
- **string pobierzPesel()** - – metoda zwracająca prywatną zmienną klasy - pesel

LEKARZ (class Lekarz{};)

Klasa ta dziedziczy z klasy OSOBA zmienne takie jak imie, nazwisko oraz numer_telefonu. Dodatkowo klasa ta posiada numerLekarza (typ string, zmienna prywatna klasy) - jest to numer nadany lekarzowi (indywidualny numer licencji dla każdego z lekarzy) oraz pole specjalizacja (typ string, zmienna prywatna klasy) - zmienna przechowuje informacje o specjalizacji danego lekarza.

Publiczne metody klasy:

- **string pobierzDaneOsoby()** - pozwalająca na pobranie danych z prywatnych zmiennych w klasie (tj. imienia, nazwiska, numeru telefonu, numeru lekarza, specjalizacji), zwraca string z tymi danymi
- **string pobierzSpecjalizacje()** – metoda odpowiedzialna za zwracanie stringa, który przechowuje specjalizację lekarza.

- **Zmiana pobierzZmianeZDnia (std::string data)** - metoda zwraca zmianę lekarza z danego dnia (zmiana tj godziny pracy w danym dniu)
- **int pobierzNumerLekarza()** – metoda zwracająca zmienną prywatną klasy - numerLekarza
- **vector<Zmiana> pobierzZmiany()** - metoda odpowiedzialna za zwracanie zmian (godzin pracy z dniami) danego lekarza

WIZYTA (class Wizyta{};)

Klasa ta ma obrazować pojedynczą wizytę pacjenta u lekarza. Prywatne zmienne klasy: peselPacjenta (typ string), numerLekarza (typ int), data (typ string), godzina (typ int).

Publiczne metody klasy:

- **string pobierzDate()** - metoda zwracająca zmienną prywatną klasy – data
- **int pobierzGodzine()** – metoda zwracająca zmienną prywatną klasy – godzina
- **string pobierzPeselPacjenta()** - metoda zwracająca zmienną prywatną klasy – peselPacjenta
- **int pobierzNumerLekarza()** - metoda zwracająca zmienną prywatną klasy – numerLekarza
- **bool czySieOdbyla()** – metoda sprawdzająca, czy dana wizyta już się odbyła

KONSOLA (class Konsola{};)

Klasa ta jest odpowiedzialna za komunikację z użytkownikiem oraz wyświetlanie danych w konsoli i za wczytanie danych.

W konsoli znajdują się publicznie zmienne takie jak: lekarze (vector <Lekarz>), spisPacjentow(vector <Pacjent>) oraz unique_ptr<Pacjent> uzytkownikPtr – inteligentny wskaźnik na klasę Pacjent. Zmienne spisPacjentow i lekarze to wektory, do których są przekazywane dane pobrane z plików wejściowych. Zmienne te mają symbolizować listę pacjentów oraz lekarzy.

Publiczne metody klasy:

- ***void Start()*** - metoda odpowiedzialna za wyświetlanie ekranu startowego w konsoli oraz udostępnienie użytkownikowi poszczególnych opcji. W pierwszej kolejności metoda wywołuje metodę `logowanie()`, następnie, jeśli użytkownik się zalogował wyświetla menu i opcje do wyboru. Wybór opcji został zaimplementowany za pomocą `switch'a`.

Prywatne metody klasy:

- ***void SzukajLekarza()*** - pozwala na znalezienie lekarza po specjalizacji oraz umówienie wizyty do wybranego lekarza. Metoda posiada zmienne lokalną `specjalizacja`, typu `string`, zmienną typu `bool` `czyIstniejeDanaSpecjalizacja`, z domyślną wartością `false` oraz zmienną typu `int` o domyślnej wartości `0` `numerLekarza`.

Najpierw wywołana metoda prosi o podanie specjalizacji lekarza, jakiego poszukujemy i umieszcza ją w zmiennej `specjalizacja`. Następnie za pomocą pętli `for` sprawdza, czy lekarz o podanej specjalizacji istnieje w wektorze `lekarze`.

Jeśli istnieje to zmienia zmienną `czyIstniejeDanaSpecjalizacja` na `true` i wyświetla wszystkich lekarzy posiadających daną specjalizację i ich dane. Następnie wyświetlone zostaje menu wyboru – pacjent może wybrać czy chce się zapisać do lekarza (poprzez podanie jego numeru) lub, czy chce wrócić go głównego menu (podając `0`). Odpowiedź użytkownika jest przechowywana w zmiennej `numerLekarza`. Jeśli numer lekarza jest poprawny to wywołana zostaje metoda `zapiszSie()`, jeśli numer został podany błędnie pojawi się komunikat o niepoprawnie wprowadzonych danych.

Jeśli dana specjalizacja nie istnieje (jeśli zmienna `czyIstniejeDanaSpecjalizacja` ma wartość `false`) to wyświetla komunikat informujący o tym, że w tej przychodni nie przyjmuje lekarz o zadanej specjalizacji i powraca do menu wyboru.

- ***void ZapiszSie()*** - metoda umożliwiająca zapisanie się do konkretnego lekarza.. W pierwszej kolejności użytkownik proszony jest o podanie daty wizyty jaka go interesuje. Jego odpowiedź jest przechowywana w zmiennej `data`. Następnie szuka tylko grafiku na dany dzień dla danego lekarza. Później wyświetla grafik na dany dzień i informuje o dostępności godzin. Pacjent może podać godzinę jaka go interesuje. Jeśli godzina jest poprawna to utworzony zostaje nowy obiekt klasy `wizyta` i zostaje ona przypiana do konta pacjenta oraz grafiku lekarza.
- ***bool czyPoprawnaNazwa(std::string nazwa)***- metoda odpowiedzialna za sprawdzanie czy w podanym stringu na pewno znajdują się same litery.

- **bool czyPoprawnaNazwa(std::string nazwa)** - metoda odpowiedzialna za sprawdzanie czy w podanym stringu na pewno znajdują się same litery.
- **bool czyPoprawnaData(std::string data)** - metoda odpowiedzialna za sprawdzanie czy data została podana w poprawny sposób.
- **bool czyPoprawnyNumer(std::string numer)** - metoda odpowiedzialna za sprawdzanie czy w podanym stringu (numerze telefonu) na pewno znajdują się same cyfry oraz czy ciąg składa się z 9 cyfr.
- **bool czyPoprawnyPesel(std::string PESEL)** - metoda odpowiedzialna za sprawdzanie czy podany sting (Pesel) został wprowadzony poprawnie tj, czy spełnia założenia sumy kontrolnej (ustalonej ogólnie) oraz czy ma 11 cyfr.
- **void wczytajPacjentow()** - metoda odpowiedzialna za wczytywanie danych pacjentów z pliku.
- **void wczytajLekarzy()** - metoda odpowiedzialna za wczytywanie danych lekarzy z pliku.
- **void ZapiszWizytyDoPliku()** - metoda odpowiedzialna za wczytywanie wizyt z pliku tekstowego.
- **void SzukajLekarza()** - metoda odpowiedzialna za wyszukiwanie lekarza.
- **bool czyZPrzeszlosci(std::string data, int godzina)** - metoda odpowiedzialna za sprawdzanie czy data wraz z godziną nie jest z przeszłości (tj. czy na pewno wyprzedza czas pobrany z komputera).
- **Lekarz znajdzLekarzaPoNumerzeLekarza(unsigned int numerLekarza)** - metoda odpowiedzialna za wyszukiwanie lekarza po jego numerze.
- **void Logowanie()** - metoda odpowiedzialna za logowanie do systemu oraz zakładanie konta użytkownikom. Posiada zmienną lokalną typu bool NiePoprawnyPesel, o domyślnej wartości true. Najpierw prosi użytkownika o podanie numeru pesel, w celu sprawdzenia, czy istnieje już w bazie, czy należy utworzyć nowe konto. W pierwszej kolejności sprawdza poprawność wprowadzonego numeru (jeśli jest błędny to wyświetla komunikat i prosi o ponowne wprowadzenie) , następnie (jeśli numer jest poprawny) za pomocą pętli „for” sprawdza czy podany pesel znajduje się w wektorze pacjentów spisPacjentow. Jeśli tak to loguje użytkownika, jeśli jest to osoba niezarejestrowana to przechodzi to sekcji rejestracji i prosi o podanie danych osobowych (tworzy przy tym nowy obiekt klasy Pacjent) oraz przypisuje do zmiennej uzytkownikPtr wskaźnik na danego (zalogowanego) użytkownika.

- **bool funkcja_sort(Zmiana a, Zmiana b)** - metoda odpowiedzialna za sortowanie godzin pracy lekarzy.
- **void SprawdzProfil()** - metoda umożliwiająca sprawdzenie swojego profilu przez pacjenta.
- **Pacjent znajdzPacjentaPoPesel(string pesel)** – metoda, która przyjmuje zmienną typu string (przekazujemy do niej pesel osoby szukanej), następnie przeszukuje wektor – spisPacjentow i zwraca instancje o podanym numerze pesel.
- **void SprawdzWizyty()** - metoda służąca do wyświetlania wizyt zalogowanego użytkownika.
- **void WyswietlGrafikIDane()**- metoda odpowiedzialna za wyświetlanie grafiku oraz danych wybranego lekarza.
- **void WyszukiwanieLekrza()** - Metoda odpowiedzialna za wyszukiwanie lekarzy (specjalizacji).
- **void wczytajPacjentow()** -wczytuje pacjentow z pliku tekstowego (pacjenci.txt), a następnie umieszcza je w wektorze o nazwie spisPacjentow. Po zakończeniu zaczytywania danych zamyka plik.
- **void zapiszDanePacjenta(std::string imie, std::string nazwisko, std::string PESEL, std::string DataUrodzenia, std::string numerTelefonu)** - metoda odpowiedzialna za zapisywanie danych pacjenta do pliku. Po zakończeniu zapisywania zamyka plik.

WIZYTY (class Wizyty{;})

Zbiór pojedynczych wizyt dla pacjenta i dla lekarza. Zawiera prywatne zmienne wielkosc (unsigned int) oraz shared_ptr<Wizyta> pierwszaWizyta – inteligentny wskaźnik na pierwszą wizytę.

Publiczne metody klasy:

- **void Dodaj (std::shared_ptr<Wizyta> wizytaPtr)** - pozwalająca na dodanie wizyty do grafiku
- **void Usun (std::string data, std::string godzina)** - pozwalająca na usunięcie wizyty z grafiku

- **vector<Wizyta> pobierzDlaDanegoLekarza(unsigned int numerLekarza)** – metoda odpowiedzialna za zwrócenie wizyt danego lekarza.
- **vector<Wizyta> pobierzDlaDanegoPacjenta(std::string peselPacjenta)** - metoda odpowiedzialna za zwrócenie wizyt danego pacjenta.
- **vector<Wizyta> pobierzWszystkieWizyty()** - metoda która zwraca wszystkie wizyty jakie istnieją

ZMIANA (class Zmiana{};)

Obrazuje pojedynczy dzień pracy lekarza tj dzień tygodnia, wraz z godzinami pracy. Klasa zawiera zmienne prywatne dzień (typ Dzień – zdefiniowany w pliku nagłówkowym Dzień.h) oraz godzinaRozpoczenia i godzinaZakonczenia (typu unsigned short int).

Publiczne metody klasy:

- **Dzień pobierzDzień()** - metoda odpowiedzialna za zwracanie zmiennej typu Dzień, zwraca wartość ze zmiennej prywatnej klasy o nazwie dzień.
- **int pobierzGodzineRozpoczenia()**- metoda odpowiedzialna za zwracanie zmiennej typu int, zwraca wartość ze zmiennej prywatnej klasy o nazwie godzinaRozpoczenia.
- **int pobierzGodzineZakonczenia()**- metoda odpowiedzialna za zwracanie zmiennej typu int, zwraca wartość ze zmiennej prywatnej klasy o nazwie godzinaZakonczenia.

3. SPECYFIKACJA ZEWNĘTRZNA

Program prowadzi użytkownika przez poszczególne opcje, użytkownik dokonuje wyborów za pomocą wpisywania poleceń w konsolę. Głównym punktem nawigacyjnym jest menu główne.

4. SPECYFIKACJA WEWNĘTRZNA

Program został zrealizowany zgodnie z paradygmatem strukturalnym. W programie rozdzielono interfejs (komunikacji z użytkownikiem) od logiki aplikacji.

4.1 OGÓLNA STRUKTURA PROGRAMU

Program tworzy obiekt klasy Konsola(), a następnie wywołuje metodę Start(). Pierwsze pojawia się logowanie za pomocą numeru pesel. Użytkownik może się zalogować lub utworzyć konto, jeśli nie znajduje się w bazie pacjentów następnie za pomocą menu opartego na instrukcji switch może wybrać opcje, która go interesuje. Może np. zapisać się na wizytę lub ją odwołać. Program na chwilę po rozpoczęciu pracy pobiera dane z plików

tekstowych i zapisuje w odpowiednich kontenerach na dane (pobiera listę pacjentów, lekarzy oraz spis wizyt do lekarzy), po zakończeniu sczytywania plików zamyka je. Po zakończeniu pracy wypisuje wizyty do pliku (jeśli jakieś przybyły od ostatniego uruchomienia).

4.2 SZCZEGÓŁOWY OPIS TYPÓW I FUNKCJI

Szczegółowy opis typów i funkcji zawarty jest w załączniku.

5. TESTOWANIE

Program został przetestowany na różnych plikach wejściowych. Włącznie z plikami, w których nie wszystkie dane są prawidłowe (część linijek jest pusta, w pliku znajdują się losowe znaki lub brakuje jakiejś danej).

Program wczytując dane od użytkownika sprawdza ich poprawność, w razie podania błędnych danych wyświetla komunikat.

Program został przetestowany pod kątem wycieków pamięci. Pomimo dużych rozmiarów plików na jakich program był testowany, nie pojawił się komunikat o naruszeniu ochrony pamięci.

6. WNIOSKI

Program podczas tworzenia sprawił mi kilka trudności. Przede wszystkim dużym utrudnieniem okazało się pracowanie z biblioteką związaną z czasem – potrzeba była ona do kontrolowania aktualności daty i oceniania, czy podana data aby na pewno jest z przyszłości. Rzutowanie dat na typ int oraz wyliczanie konkretnych wartości dla tm. Kolejnym dość sporym problemem okazał się dostęp do zmiennych z klas. Jednak finalnie większość problemów została rozwiązana i zdobyłam nową wiedzę.

Literatura:

- „C++ Przewodnik Dla Początkujących” Alex Allain
- „Podstawy Programowania w Języku C++” Józef Zieliński

Rejestracja W Przychodni

Wygenerowano przez Doxygen 1.9.1

1 Indeks hierarchiczny	1
1.1 Hierarchia klas	1
2 Indeks klas	3
2.1 Lista klas	3
3 Indeks plików	5
3.1 Lista plików	5
4 Dokumentacja klas	7
4.1 Dokumentacja klasy Konsola	7
4.1.1 Opis szczegółowy	7
4.1.2 Dokumentacja funkcji składowych	7
4.1.2.1 start()	7
4.2 Dokumentacja klasy Lekarz	8
4.2.1 Opis szczegółowy	8
4.2.2 Dokumentacja konstruktora i destruktora	8
4.2.2.1 Lekarz()	8
4.2.3 Dokumentacja funkcji składowych	8
4.2.3.1 pobierzDaneOsoby()	9
4.2.3.2 pobierzNumerLekarza()	9
4.2.3.3 pobierzSpecjalizacje()	9
4.2.3.4 pobierzZmianeZDnia()	9
4.2.3.5 pobierzZmiany()	10
4.3 Dokumentacja klasy Osoba	10
4.3.1 Opis szczegółowy	10
4.3.2 Dokumentacja konstruktora i destruktora	10
4.3.2.1 Osoba()	11
4.3.3 Dokumentacja funkcji składowych	11
4.3.3.1 pobierzDaneOsoby()	11
4.3.3.2 pobierzNazwisko()	11
4.4 Dokumentacja klasy Pacjent	11
4.4.1 Opis szczegółowy	12
4.4.2 Dokumentacja konstruktora i destruktora	12
4.4.2.1 Pacjent()	12
4.4.3 Dokumentacja funkcji składowych	12
4.4.3.1 pobierzDaneOsoby()	12
4.4.3.2 pobierzPesel()	13
4.5 Dokumentacja klasy Wizyta	13
4.5.1 Opis szczegółowy	13
4.5.2 Dokumentacja konstruktora i destruktora	13
4.5.2.1 Wizyta()	14
4.5.3 Dokumentacja funkcji składowych	14

4.5.3.1 czySieOdbyla()	14
4.5.3.2 pobierzDate()	14
4.5.3.3 pobierzGodzine()	14
4.5.3.4 pobierzNumerLekarza()	15
4.5.3.5 pobierzPeselPacjenta()	15
4.5.4 Dokumentacja atrybutów składowych	15
4.5.4.1 kolejnaWizyta	15
4.6 Dokumentacja klasy Wizyty	15
4.6.1 Opis szczegółowy	16
4.6.2 Dokumentacja konstruktora i destruktora	16
4.6.2.1 Wizyty()	16
4.6.3 Dokumentacja funkcji składowych	16
4.6.3.1 dodaj()	16
4.6.3.2 pobierzDlaDanegoLekarz()	16
4.6.3.3 pobierzDlaDanegoPacjenta()	17
4.6.3.4 pobierzWszystkieWizyty()	17
4.6.3.5 usun()	17
4.7 Dokumentacja klasy Zmiana	18
4.7.1 Opis szczegółowy	18
4.7.2 Dokumentacja konstruktora i destruktora	18
4.7.2.1 Zmiana()	18
4.7.3 Dokumentacja funkcji składowych	18
4.7.3.1 pobierzDzien()	18
4.7.3.2 pobierzGodzineRozpoczecia()	19
4.7.3.3 pobierzGodzineZakonczenia()	19
5 Dokumentacja plików	21
5.1 Dokumentacja pliku Dzień.h	21
5.1.1 Dokumentacja typów wyliczanych	21
5.1.1.1 Dzień	21
5.2 Dokumentacja pliku Konsola.cpp	22
5.3 Dokumentacja pliku Konsola.h	22
5.4 Dokumentacja pliku Lekarz.cpp	22
5.5 Dokumentacja pliku Lekarz.h	22
5.6 Dokumentacja pliku Osoba.cpp	23
5.7 Dokumentacja pliku Osoba.h	23
5.8 Dokumentacja pliku Pacjent.cpp	23
5.9 Dokumentacja pliku Pacjent.h	23
5.10 Dokumentacja pliku Projektsem2Przychodnia.cpp	23
5.10.1 Dokumentacja funkcji	24
5.10.1.1 main()	24
5.11 Dokumentacja pliku Wizyta.cpp	24

5.12 Dokumentacja pliku Wizyta.h	24
5.13 Dokumentacja pliku Wizyty.cpp	24
5.14 Dokumentacja pliku Wizyty.h	24
5.15 Dokumentacja pliku Zmiana.cpp	25
5.16 Dokumentacja pliku Zmiana.h	25
Indeks	27

Rozdział 1

Indeks hierarchiczny

1.1 Hierarchia klas

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

Konsola	7
Osoba	10
Lekarz	8
Pacjent	11
Wizyta	13
Wizyty	15
Zmiana	18

Rozdział 2

Indeks klas

2.1 Lista klas

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

Konsola	Klasa ta jest odpowiedzialna za komunikację z użytkownikiem oraz wyświetlanie danych w konsoli i za wczytanie danych	7
Lekarz	Klasa, która dziedziczy z klasy Osoba , mająca za zadanie obrazować Lekarza w przychodni . .	8
Osoba	Klasa opisująca osobę, podstawowe informacje personalne każdego bywalca przychodni . . .	10
Pacjent	Klasa, która dziedziczy z klasy Osoba , mająca za zadanie obrazować Pacjenta w przychodni .	11
Wizyta	Klasa mająca za zadanie obrazowanie pojedynczej wizyty	13
Wizyty	Klasa mająca symbolizować spis Wizyt np dla danego lekarza	15
Zmiana	Klasa obrazująca poszczególne zmiany lekarza	18

Rozdział 3

Indeks plików

3.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

Dzien.h	21
Konsola.cpp	22
Konsola.h	22
Lekarz.cpp	22
Lekarz.h	22
Osoba.cpp	23
Osoba.h	23
Pacjent.cpp	23
Pacjent.h	23
Projektsem2Przychodnia.cpp	23
Wizyta.cpp	24
Wizyta.h	24
Wizyty.cpp	24
Wizyty.h	24
Zmiana.cpp	25
Zmiana.h	25

Rozdział 4

Dokumentacja klas

4.1 Dokumentacja klasy Konsola

Klasa ta jest odpowiedzialna za komunikację z użytkownikiem oraz wyświetlanie danych w konsoli i za wczytanie danych.

```
#include <Konsola.h>
```

Metody publiczne

- void [start](#) ()

4.1.1 Opis szczegółowy

Klasa ta jest odpowiedzialna za komunikację z użytkownikiem oraz wyświetlanie danych w konsoli i za wczytanie danych.

4.1.2 Dokumentacja funkcji składowych

4.1.2.1 start()

```
void Konsola::start ( )
```

Metoda, która wyświetla główne menu programu i przekierowuje użytkownika do konkretnych funkcji programu, poza tym metoda ta wywołuje funkcje odpowiedzialne za wczytywanie danych z pliku (pacjentów, lekarzy oraz wizyt).

Dokumentacja dla tej klasy została wygenerowana z plików:

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

4.2 Dokumentacja klasy Lekarz

Klasa, która dziedziczy z klasy [Osoba](#), mająca za zadanie obrazować Lekarza w przychodni.

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

Diagram dziedziczenia dla Lekarz

Diagram współpracy dla Lekarz:

Metody publiczne

- [Lekarz](#) (const std::string &imie, const std::string &nazwisko, const std::string &numerTelefonu, int numerLekarza, const std::string &specjalizacja, const std::vector< [Zmiana](#) > &zmiiany)
- std::string [pobierzSpecjalizacje](#) ()
- std::string [pobierzDaneOsoby](#) ()
- int [pobierzNumerLekarza](#) ()
- [Zmiana](#) [pobierzZmianeZDnia](#) (std::string data)
- std::vector< [Zmiana](#) > [pobierzZmiany](#) ()

4.2.1 Opis szczegółowy

Klasa, która dziedziczy z klasy [Osoba](#), mająca za zadanie obrazować Lekarza w przychodni.

4.2.2 Dokumentacja konstruktora i destruktor

4.2.2.1 Lekarz()

```
Lekarz::Lekarz (
    const std::string & imie,
    const std::string & nazwisko,
    const std::string & numerTelefonu,
    int numerLekarza,
    const std::string & specjalizacja,
    const std::vector< Zmiana > & zmiiany )
```

4.2.3 Dokumentacja funkcji składowych

4.2.3.1 pobierzDaneOsoby()

```
std::string Lekarz::pobierzDaneOsoby ( ) [virtual]
```

Metoda odpowiedzialna za zwracanie stringa złożonego z wyrazów

Zwraca

retVal - zwraca string w tym przypadku zmienną o nazwie retVal, która przechowuje dane lekarza

Reimplementowana z [Osoba](#).

4.2.3.2 pobierzNumerLekarza()

```
int Lekarz::pobierzNumerLekarza ( )
```

Metoda odpowiedzialna za zwracanie zmiennej typu int, który przechowuje numer lekarza

Zwraca

numerLekarza - int, w którym przechowany jest numer lekarza

4.2.3.3 pobierzSpecjalizacje()

```
std::string Lekarz::pobierzSpecjalizacje ( )
```

Metoda odpowiedzialna za zwracanie stringa, który przechowuje specjalizacje lekarza

Zwraca

specjalizacja - string, w którym przechowana jest specjalizacja lekarza

4.2.3.4 pobierzZmianeZDnia()

```
Zmiana Lekarz::pobierzZmianeZDnia (
    std::string data )
```

Metoda odpowiedzialna za zwracanie zmiennej typu [Zmiana](#)

Metoda zwraca zmianę lekarza z danego dnia (zmiana tj godziny pracy w danym dniu)

Parametry

<i>data</i>	- string, w którym do metody podajemy datę
-------------	--

Zwraca

zwraca zmianę lekarza z danej daty

4.2.3.5 pobierzZmiany()

```
std::vector< Zmiana > Lekarz::pobierzZmiany ( )
```

Metoda odpowiedzialna za zwracanie zmian (godzin pracy z dniami) danego lekarza

Zwraca

zmiany - wektor zmian danego lekarza

Dokumentacja dla tej klasy została wygenerowana z plików:

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

4.3 Dokumentacja klasy Osoba

Klasa opisująca osobę, podstawowe informacje personalne każdego bywalca przychodni.

```
#include <Osoba.h>
```

Diagram dziedziczenia dla Osoba

Metody publiczne

- [Osoba](#) (std::string imie, std::string nazwisko, std::string numerTelefonu)
- virtual std::string [pobierzDaneOsoby](#) ()
- std::string [pobierzNazwisko](#) ()

4.3.1 Opis szczegółowy

Klasa opisująca osobę, podstawowe informacje personalne każdego bywalca przychodni.

4.3.2 Dokumentacja konstruktora i destruktora

4.3.2.1 Osoba()

```
Osoba::Osoba (
    std::string imie,
    std::string nazwisko,
    std::string numerTelefonu )
```

4.3.3 Dokumentacja funkcji składowych

4.3.3.1 pobierzDaneOsoby()

```
std::string Osoba::pobierzDaneOsoby ( ) [virtual]
```

Metoda, która ma za zadanie zwrócić dane osoby

Metoda, która łączy poszczególne stringi w jeden - imię, nazwisko oraz numer telefonu

Zwraca

Metoda zwraca string, który zawiera imię, nazwisko oraz numer telefonu osoby

Reimplementowana w [Pacjent](#) i [Lekarz](#).

4.3.3.2 pobierzNazwisko()

```
std::string Osoba::pobierzNazwisko ( )
```

Metoda, która ma za zadanie zwrócić zmienną nazwisko z klasy [Osoba](#)

Zwraca

Zwraca zmienną typu string - nazwisko z klasy [Osoba](#)

Dokumentacja dla tej klasy została wygenerowana z plików:

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

4.4 Dokumentacja klasy Pacjent

Klasa, która dziedziczy z klasy [Osoba](#), mająca za zadanie obrazować Pacjenta w przychodni.

```
#include <Pacjent.h>
```

Diagram dziedziczenia dla Pacjent

Diagram współpracy dla Pacjent:

Metody publiczne

- [Pacjent](#) (const std::string &imie, const std::string &nazwisko, const std::string &numerTelefonu, const std::string &pesl, const std::string &dataUrodzenia)
- std::string [pobierzDaneOsoby](#) ()
- std::string [pobierzPesel](#) ()

4.4.1 Opis szczegółowy

Klasa, która dziedziczy z klasy [Osoba](#), mająca za zadanie obrazować Pacjenta w przychodni.

4.4.2 Dokumentacja konstruktora i destruktor

4.4.2.1 Pacjent()

```
Pacjent::Pacjent (
    const std::string & imie,
    const std::string & nazwisko,
    const std::string & numerTelefonu,
    const std::string & pesl,
    const std::string & dataUrodzenia )
```

4.4.3 Dokumentacja funkcji składowych

4.4.3.1 pobierzDaneOsoby()

```
std::string Pacjent::pobierzDaneOsoby ( ) [virtual]
```

Metoda odpowiedzialna za zwracanie stringa złożonego z wyrazów

Zwraca

zwraca string w tym przypadku zmienną o nazwie retVal, która przechowuje dane pacjenta

Reimplementowana z [Osoba](#).

4.4.3.2 pobierzPesel()

```
std::string Pacjent::pobierzPesel ( )
```

Metoda odpowiedzialna za zwracanie stringa, który przechowuje numer pesel pacjenta

Zwraca

pesel - string, w którym przechowany jest numer pesel pacjenta

Dokumentacja dla tej klasy została wygenerowana z plików:

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

4.5 Dokumentacja klasy Wizyta

Klasa mająca za zadanie obrazowanie pojedynczej wizyty.

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

Metody publiczne

- [Wizyta](#) (const std::string &peslPacjenta, unsigned int numerLekrza, const std::string &data, int godzina)
- int [pobierzNumerLekarza](#) ()
- std::string [pobierzDate](#) ()
- std::string [pobierzPeselPacjenta](#) ()
- int [pobierzGodzine](#) ()
- bool [czySieOdbyla](#) ()

Atrybuty publiczne

- std::shared_ptr< [Wizyta](#) > [kolejnaWizyta](#)

4.5.1 Opis szczegółowy

Klasa mająca za zadanie obrazowanie pojedynczej wizyty.

4.5.2 Dokumentacja konstruktora i destruktora

4.5.2.1 Wizyta()

```
Wizyta::Wizyta (
    const std::string & peslPacjenta,
    unsigned int numerLekrza,
    const std::string & data,
    int godzina )
```

4.5.3 Dokumentacja funkcji składowych

4.5.3.1 czySieOdbyla()

```
bool Wizyta::czySieOdbyla ( )
```

Metoda odpowiedzialna za sprawdzanie czy dana wizyta już miała miejsce

Zwraca

true jeli wizyta jest z przeszłości lub false, jeli wizyta się jeszcze nie odbyła i jest zaplanowana

4.5.3.2 pobierzDate()

```
std::string Wizyta::pobierzDate ( )
```

Metoda odpowiedzialna za zwracanie stringa, który przechowuje date wizyty

Zwraca

daya - string, w którym przechowana jest data wizyty

4.5.3.3 pobierzGodzine()

```
int Wizyta::pobierzGodzine ( )
```

Metoda odpowiedzialna za zwracanie zmiennej typu int, który przechowuje godzinę wizyty

Zwraca

godzina - int, w którym przechowana jest godzina

4.5.3.4 pobierzNumerLekarza()

```
int Wizyta::pobierzNumerLekarza ( )
```

Metoda odpowiedzialna za zwracanie zmiennej typu int, który przechowuje numer lekarza

Zwraca

godzina - int, w którym przechowany jest numer lekarza

4.5.3.5 pobierzPeselPacjenta()

```
std::string Wizyta::pobierzPeselPacjenta ( )
```

Metoda odpowiedzialna za zwracanie stringa, który przechowuje numer pesel pacjenta zapisanego na daną wizytę

Zwraca

peselPacjenta - zmienna typu string przechowująca pesel pacjenta zapisanego na daną wizytę

4.5.4 Dokumentacja atrybutów składowych

4.5.4.1 kolejnaWizyta

```
std::shared_ptr<Wizyta> Wizyta::kolejnaWizyta
```

Inteligentny wskanik na obiekt klasy [Wizyta](#), ma wskazywać na kolejną wizytę jaka została wprowadzona do programu

Dokumentacja dla tej klasy została wygenerowana z plików:

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

4.6 Dokumentacja klasy Wizyty

Klasa mająca symbolizować spis Wizyt np dla danego lekarza.

```
#include <Wizyty.h>
```

Metody publiczne

- `Wizyty ()`
- `std::vector< Wizyta > pobierzDlaDanegoLekarza (unsigned int numerLekarza)`
- `void dodaj (std::shared_ptr< Wizyta > wizytaPtr)`
- `std::vector< Wizyta > pobierzDlaDanegoPacjenta (std::string peselPacjenta)`
- `void usun (std::string data, std::string godzina)`
- `std::vector< Wizyta > pobierzWszystkieWizyty ()`

4.6.1 Opis szczegółowy

Klasa mająca symbolizować spis Wizyt np dla danego lekarza.

4.6.2 Dokumentacja konstruktora i destruktora

4.6.2.1 `Wizyty()`

```
Wizyty::Wizyty ( )
```

4.6.3 Dokumentacja funkcji składowych

4.6.3.1 `dodaj()`

```
void Wizyty::dodaj (
    std::shared_ptr< Wizyta > wizytaPtr )
```

Metoda służy do doawania wizyt do wektora wizyt

Parametry

<i>wizytaPtr</i>	- wskaźnik na wizytę, którą dodajemy do pamięci
------------------	---

4.6.3.2 `pobierzDlaDanegoLekarza()`

```
std::vector< Wizyta > Wizyty::pobierzDlaDanegoLekarza (
    unsigned int numerLekarza )
```

Metoda odpowiedzialna za zwrócenie wizyt danego lekarza

Zwraca

wizyty - zwraca wektor wizyt danego lekarza

4.6.3.3 pobierzDlaDanegoPacjenta()

```
std::vector< Wizyta > Wizyty::pobierzDlaDanegoPacjenta (
    std::string peselPacjenta )
```

Metoda odpowiedzialna za zwrócenie wizyt danego pacjenta

Zwraca

wizyty - zwraca wektor wizyt danego pacjenta

4.6.3.4 pobierzWszystkieWizyty()

```
std::vector< Wizyta > Wizyty::pobierzWszystkieWizyty ( )
```

Metoda która zwraca wszystkie wizyty jakie istnieją

Zwraca

wizyty - zwraca wektor wszystkich wizyt

4.6.3.5 usun()

```
void Wizyty::usun (
    std::string data,
    std::string godzina )
```

Metoda służąca do usuwania konkretnej wizyty z wektora wizyt

Parametry

<i>data</i>	- zmienna typu string, za pomocą której podajemy datę wizyty
<i>godzina</i>	- zmienna typu int, za pomocą której podajemy godzinę wizyty

Dokumentacja dla tej klasy została wygenerowana z plików:

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

4.7 Dokumentacja klasy Zmiana

Klasa obrazująca poszczególne zmiany lekarza.

```
#include <Zmiana.h>
```

Metody publiczne

- [Zmiana](#) ([Dzien](#) dzien, unsigned short godzinaRozpoczecia, unsigned short godzinaZakonczenia)
- [Dzien pobierzDzien](#) ()
- int [pobierzGodzineRozpoczecia](#) ()
- int [pobierzGodzineZakonczenia](#) ()

4.7.1 Opis szczegółowy

Klasa obrazująca poszczególne zmiany lekarza.

4.7.2 Dokumentacja konstruktora i destruktor

4.7.2.1 Zmiana()

```
Zmiana::Zmiana (
    Dzien dzien,
    unsigned short godzinaRozpoczecia,
    unsigned short godzinaZakonczenia )
```

4.7.3 Dokumentacja funkcji składowych

4.7.3.1 pobierzDzien()

```
Dzien Zmiana::pobierzDzien ( )
```

Metoda odpowiedzialna za zwracanie zmiennej typu Dzień

Zwraca

dzien - zwraca dzien z danej zmiany

4.7.3.2 pobierzGodzineRozpoczecia()

```
int Zmiana::pobierzGodzineRozpoczecia ( )
```

Metoda odpowiedzialna za zwracanie godziny rozpoczęcia lekarza dla danej zmiany

Zwraca

godzinaRozpoczecia - zmienna typu int, godzina rozpoczęcia pracy przez lekarza

4.7.3.3 pobierzGodzineZakonczenia()

```
int Zmiana::pobierzGodzineZakonczenia ( )
```

Metoda odpowiedzialna za zwracanie godziny zakończenia lekarza dla danej zmiany

Zwraca

godzinaZakonczenia - zmienna typu int, godzina zakończenia pracy przez lekarza

Dokumentacja dla tej klasy została wygenerowana z plików:

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

Rozdział 5

Dokumentacja plików

5.1 Dokumentacja pliku Dzień.h

Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

Wyliczenia

- enum `Dzien` {
 `Poniedzialek` = 0 , `Wtorek` = 1 , `Sroda` = 2 , `Czwartek` = 3 ,
 `Piatek` = 4 , `Sobota` = 5 , `Niedziela` = 6 , `Wolne` = 7 }

Pomocniczy plik nagłówkowy tygodnia, gdzie zdefiniowane s  nazwy dni tygodnia.

5.1.1 Dokumentacja typ w wyliczanych

5.1.1.1 Dzień

enum `Dzien`

Pomocniczy plik nagłówkowy tygodnia, gdzie zdefiniowane s  nazwy dni tygodnia.

Warto ci wylicze 

Poniedzialek	
Wtorek	
Sroda	
Czwartek	
Piatek	
Sobota	
Niedziela	
Wolne	

5.2 Dokumentacja pliku Konsola.cpp

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <limits>
#include "string"
#include "Konsola.h"
```

Wykres zależności załączania dla Konsola.cpp:

5.3 Dokumentacja pliku Konsola.h

```
#include <vector>
#include "Lekarz.h"
#include "Pacjent.h"
#include "Wizyty.h"
```

Wykres zależności załączania dla Konsola.h: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

Komponenty

- class [Konsola](#)

Klasa ta jest odpowiedzialna za komunikację z użytkownikiem oraz wyświetlanie danych w konsoli i za wczytanie danych.

5.4 Dokumentacja pliku Lekarz.cpp

```
#include "Lekarz.h"
#include <utility>
#include <ctime>
```

Wykres zależności załączania dla Lekarz.cpp:

5.5 Dokumentacja pliku Lekarz.h

```
#include <vector>
#include "Osoba.h"
#include "Zmiana.h"
```

Wykres zależności załączania dla Lekarz.h: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

Komponenty

- class [Lekarz](#)

Klasa, która dziedziczy z klasy [Osoba](#), mająca za zadanie obrazować Lekarza w przychodni.

5.6 Dokumentacja pliku Osoba.cpp

```
#include "Osoba.h"  
#include <utility>
```

Wykres zależności załączania dla Osoba.cpp:

5.7 Dokumentacja pliku Osoba.h

```
#include "string"
```

Wykres zależności załączania dla Osoba.h: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

Komponenty

- class [Osoba](#)

Klasa opisująca osobę, podstawowe informacje personalne każdego bywalca przychodni.

5.8 Dokumentacja pliku Pacjent.cpp

```
#include "Pacjent.h"  
#include <utility>
```

Wykres zależności załączania dla Pacjent.cpp:

5.9 Dokumentacja pliku Pacjent.h

```
#include "Osoba.h"  
#include "chrono"
```

Wykres zależności załączania dla Pacjent.h: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

Komponenty

- class [Pacjent](#)

Klasa, która dziedziczy z klasy [Osoba](#), mająca za zadanie obrazować Pacjenta w przychodni.

5.10 Dokumentacja pliku Projektsem2Przychodnia.cpp

```
#include "Konsola.h"
```

Wykres zależności załączania dla Projektsem2Przychodnia.cpp:

Funkcje

- int [main](#) ()

5.10.1 Dokumentacja funkcji

5.10.1.1 main()

```
int main ( )
```

5.11 Dokumentacja pliku Wizyta.cpp

```
#include "Wizyta.h"  
#include <utility>  
#include "chrono"
```

Wykres zależności załączania dla Wizyta.cpp:

5.12 Dokumentacja pliku Wizyta.h

```
#include <string>  
#include <iostream>  
#include <memory>  
#include <thread>  
#include <chrono>  
#include <mutex>
```

Wykres zależności załączania dla Wizyta.h: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

Komponenty

- class [Wizyta](#)

Klasa mająca za zadanie obrazowanie pojedynczej wizyty.

5.13 Dokumentacja pliku Wizyty.cpp

```
#include <iostream>  
#include <fstream>  
#include <utility>  
#include <vector>  
#include "Wizyty.h"
```

Wykres zależności załączania dla Wizyty.cpp:

5.14 Dokumentacja pliku Wizyty.h

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

Wykres zależności załączania dla Wizyty.h: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

Komponenty

- class [Wizyty](#)

Klasa mająca symbolizować spis Wizyt np dla danego lekarza.

5.15 Dokumentacja pliku Zmiana.cpp

```
#include "Zmiana.h"
```

Wykres zależności załączania dla Zmiana.cpp:

5.16 Dokumentacja pliku Zmiana.h

```
#include "Dzien.h"
```

Wykres zależności załączania dla Zmiana.h: Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:

Komponenty

- class [Zmiana](#)

Klasa obrazująca poszczególne zmiany lekarza.

Indeks

Czwartek
 Dzien.h, [21](#)
czySieOdbyla
 Wizyta, [14](#)

dodaj
 Wizyty, [16](#)

Dzien
 Dzien.h, [21](#)

Dzien.h, [21](#)
 Czwartek, [21](#)
 Dzien, [21](#)
 Niedziela, [21](#)
 Piatek, [21](#)
 Poniedzialek, [21](#)
 Sobota, [21](#)
 Sroda, [21](#)
 Wolne, [21](#)
 Wtorek, [21](#)

kolejnaWizyta
 Wizyta, [15](#)

Konsola, [7](#)
 start, [7](#)

Konsola.cpp, [22](#)
Konsola.h, [22](#)

Lekarz, [8](#)
 Lekarz, [8](#)
 pobierzDaneOsoby, [8](#)
 pobierzNumerLekarza, [9](#)
 pobierzSpecjalizacje, [9](#)
 pobierzZmianeZDnia, [9](#)
 pobierzZmiany, [10](#)
Lekarz.cpp, [22](#)
Lekarz.h, [22](#)

main
 Projektsem2Przychodnia.cpp, [24](#)

Niedziela
 Dzien.h, [21](#)

Osoba, [10](#)
 Osoba, [10](#)
 pobierzDaneOsoby, [11](#)
 pobierzNazwisko, [11](#)

Osoba.cpp, [23](#)
Osoba.h, [23](#)

Pacjent, [11](#)
 Pacjent, [12](#)
 pobierzDaneOsoby, [12](#)
 pobierzPesel, [12](#)
Pacjent.cpp, [23](#)
Pacjent.h, [23](#)
Piatek
 Dzien.h, [21](#)
pobierzDaneOsoby
 Lekarz, [8](#)
 Osoba, [11](#)
 Pacjent, [12](#)
pobierzDate
 Wizyta, [14](#)
pobierzDlaDanegoLekarza
 Wizyty, [16](#)
pobierzDlaDanegoPacjenta
 Wizyty, [17](#)
pobierzDzien
 Zmiana, [18](#)
pobierzGodzine
 Wizyta, [14](#)
pobierzGodzineRozpoczenia
 Zmiana, [18](#)
pobierzGodzineZakonczenia
 Zmiana, [19](#)
pobierzNazwisko
 Osoba, [11](#)
pobierzNumerLekarza
 Lekarz, [9](#)
 Wizyta, [14](#)
pobierzPesel
 Pacjent, [12](#)
pobierzPeselPacjenta
 Wizyta, [15](#)
pobierzSpecjalizacje
 Lekarz, [9](#)
pobierzWszystkieWizyty
 Wizyty, [17](#)
pobierzZmianeZDnia
 Lekarz, [9](#)
pobierzZmiany
 Lekarz, [10](#)
Poniedzialek
 Dzien.h, [21](#)
Projektsem2Przychodnia.cpp, [23](#)
 main, [24](#)

Sobota
 Dzien.h, [21](#)

Sroda

- Dzien.h, [21](#)
- start
 - Konsola, [7](#)
- usun
 - Wizyty, [17](#)
- Wizyta, [13](#)
 - czySieOdbyla, [14](#)
 - kolejnaWizyta, [15](#)
 - pobierzDate, [14](#)
 - pobierzGodzine, [14](#)
 - pobierzNumerLekarza, [14](#)
 - pobierzPeselPacjenta, [15](#)
 - Wizyta, [13](#)
- Wizyta.cpp, [24](#)
- Wizyta.h, [24](#)
- Wizyty, [15](#)
 - dodaj, [16](#)
 - pobierzDlaDanegoLekarz, [16](#)
 - pobierzDlaDanegoPacjenta, [17](#)
 - pobierzWszystkieWizyty, [17](#)
 - usun, [17](#)
 - Wizyty, [16](#)
- Wizyty.cpp, [24](#)
- Wizyty.h, [24](#)
- Wolne
 - Dzien.h, [21](#)
- Wtorek
 - Dzien.h, [21](#)
- Zmiana, [18](#)
 - pobierzDzien, [18](#)
 - pobierzGodzineRozpoczecia, [18](#)
 - pobierzGodzineZakonczenia, [19](#)
 - Zmiana, [18](#)
- Zmiana.cpp, [25](#)
- Zmiana.h, [25](#)