****

**Programowanie obiektowe**

*Program do zarządzania przychodnią*

Prowadzący: Autor:

mgr inż. Ewa Żesławska Tomasz Lula

w67033

Kierunek: 3 IID/2021, grupa GP01

Rzeszów,r.a. 2022/2023

Spis treści

[**1.** **Cele projektu** 3](#_Toc125467490)

[**2.** **Opis techniczny projektu** 3](#_Toc125467491)

[**3.** **Harmonogram realizacji projektu** 3](#_Toc125467492)

[**4.** **Prezentacja warstwy użytkowej projektu** 4](#_Toc125467493)

[**5.** **Testy jednostkowe** 6](#_Toc125467494)

[**6.** **Repozytorium, system kontroli wersji** 7](#_Toc125467495)

[**7.** **Podsumowanie** 7](#_Toc125467496)

[**8.** **Literatura** 7](#_Toc125467497)

1. **Cele projektu**

Celem projektu jest skonstruowanie prostej aplikacji pozwalającej zarządzać pacjentami oraz personelem przychodni. Istotną funkcją jest zapisywanie się na wizytę, gdzie wybieramy dzień i godzinę oraz lekarza który nam odpowiada.

**Wymagania funkcjonalne i niefunkcjonalne**

**Wymagania funkcjonalne**

* Dodawanie oraz usuwanie pacjentów i personelu z bazy danych.
* Wyświetlanie danych osobowych oraz dat wizyt.
* Wyświetlanie wszystkich danych pacjentów.
* Możliwość zaktualizowania danych.
* Walidacja danych aby zapobiec wpisaniu danych w złym formacie lub wybraniu danych spoza zakresu bazy danych.
* Nawiązywanie połączenia z bazą danych.

**Wymagania niefunkcjonalne**

* Program jest prosty w obsłudze dla użytkownika.
* Program jest wydajny i szybki podczas operacji na bazie.
* Program stworzony w języku Java.
* Program nawiązuje połączenie z bazą danych i używa rekordów w niej zapisanych.

1. **Opis techniczny projektu**

* Środowisko programistyczne Javy: Java OpenJDK 19, IntelliJ IDEA.
* Narzędzia: Maven.
* Baza danych: SQLite.
* Dodatkowe biblioteki: mysql-connector-j-8.0.31.
* Urządzenie z systemem operacyjnym Windows 10.
* JUnit 5

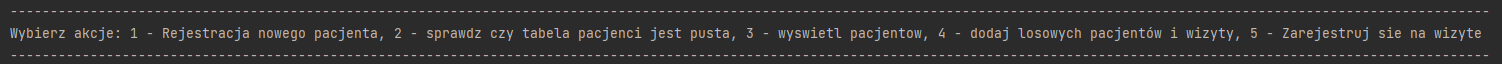
1. **Harmonogram realizacji projektu**

Poniżej zamieszono harmonogram realizacji projektu.

Rysunek . Diagram Gantta

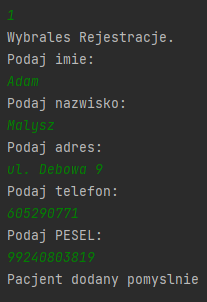
1. **Prezentacja warstwy użytkowej projektu**

Na rysunku 2 przedstawiono menu główne aplikacji, które zostaje wyświetlone po uruchomieniu:



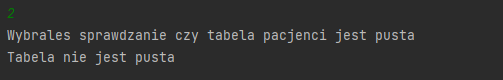
Rysunek . Menu główne

Wybierając rejestracje nowego pacjenta uruchomi się menu:



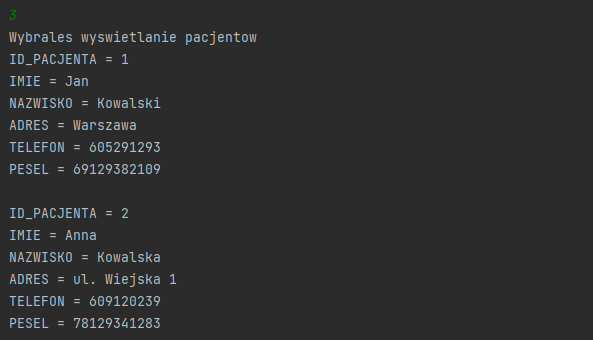
Rysunek . Rejestracja nowego pacjenta

Wybierając sprawdzenie czy tabela pacjenci jest pusta wykona się następujący program:



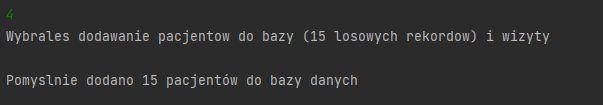
Rysunek . Sprawdzanie wypełnienia tabeli pacjenci

Wybierając wyświetlanie pacjentów otrzymamy wypis wszystkich danych dotyczących pacjentów:



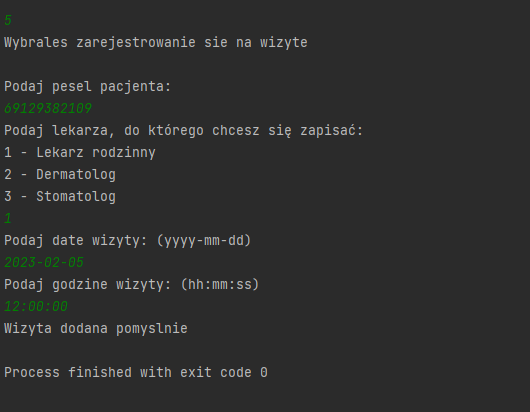
Rysunek . Wyświetlanie pacjentów

Wybierając dodawanie pacjentów do bazy, uzupełniamy po prostu bazę aby nie była pusta.



Rysunek . Uzupełnianie bazy danymi

Menu pracownika jest znacznie bardziej rozbudowane ponieważ możemy zarządzać wszystkimi danymi:

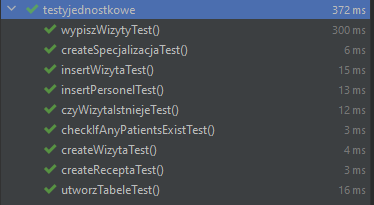


Rysunek . Dodawanie wizyty

1. **Testy jednostkowe**

Testy jednostkowe dla programu zostały wykonane z użyciem biblioteki JUnit, która pozwala na łatwe i efektywne pisanie testów jednostkowych w języku Java. Testy jednostkowe pozwoliły na szybkie wykrycie i naprawienie występujących błędów w kodzie oraz umożliwiły zwiększenie pewności co do poprawności działania poszczególnych klas.

W ramach programu realizowane były testy dla klas dotyczących tabel w bazie danych. Umożliwiły one weryfikację poprawności zapisywania, odczytywania i aktualizacji danych w bazie, a także weryfikację zachowania tych klas w różnych sytuacjach i wariantach testowych.



Rysunek . Testy

1. **Repozytorium, system kontroli wersji**

Projekt został zrealizowany z wykorzystaniem systemu kontroli wersji Git. Dokumentacja oraz projekt został umieszczony w repozytorium dostępnym pod adresem:

[https://github.com/sveddah/ProjektPrzychodnia](https://github.com/sveddah/ProjektPrzychodnia%20)

1. **Podsumowanie**

Dokumentacja ma na celu przedstawienie w sposób zrozumiały dla odbiorcy działanie projektu oraz jego stronę techniczna. Aplikacja spełnia wszystkie założenia oraz jest łatwa do rozbudowy lub jej przeróbki.

1. **Literatura**

1. <https://stackoverflow.com> (data dostępu: 19.01.2023)
2. <https://docs.oracle.com/javase/tutorial/index.html> (data dostępu: 20.01.2023)
3. <https://github.com> (data dostępu 20.01.2023)
4. https://www.youtube.com/ (data dostępu 15.01.2023)
5. <https://www.w3schools.com/java/> (data dostępu 20.01.2023)