****

**Programowanie obiektowe**

*Baza danych pojazdów*

Prowadzący: Autor:

mgr inż. Ewa Żesławska *Arkadiusz Haznar*

nr albumu: 125120

Kierunek: Informatyka, grupa lab 1

Rzeszów, 27.06.2023

Spis treści

[**1.** **Opis założeń projektu** 3](#_Toc128654617)

[**2.** **Specyfikacja wymagań** 3](#_Toc128654618)

[**3.** **Diagram przypadków użycia** 4](#_Toc128654619)

[**4.** **Harmonogram realizacji projektu (diagram Gantta)** 4](#_Toc128654620)

[**5.** **Opis techniczny projektu** 5](#_Toc128654621)

[**6.** **Prezentacja warstwy użytkowej projektu** 6](#_Toc128654622)

[**7.** **System kontroli wersji** 7](#_Toc128654623)

[**8.** **Literatura** 8](#_Toc128654624)

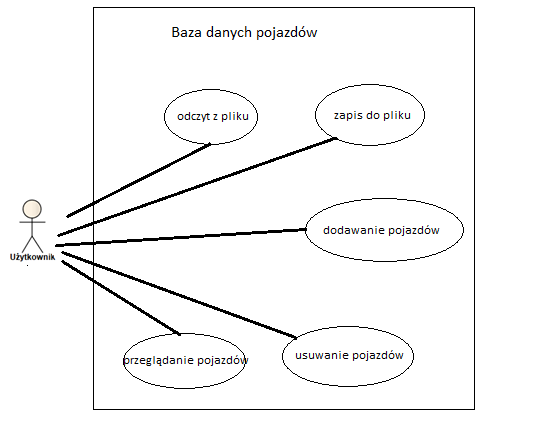
1. **Opis założeń projektu**

Użytkownik może dodawać do pliku nowe pojazdy. Każdy pojazd jest reprezentowany przez następujące dane: marka, model, kolor, rok produkcji, przebieg i (dodatkowo w zależności od wyboru pojazdu (samochód): rodzaj nadwozia, ilość drzwi, (motocykl): rodzaj motoru, spalanie, (ciężarowy): waga, długość, (autobus): maksymalna liczba osób, długość, (ciągnik): maksymalny załadunek. Użytkownik może dodawać dane dotyczące różnych pojazdów do pliku (najpierw musi dodać do listy, później może zapisać dane do pliku). Może również usuwać wcześniej wprowadzone dane na przykład w przypadku pomyłki, aby to zrobić musi podać numer linii tekstu zaczynając od zera, które dane chce usunąć. W każdej chwili może zobaczyć dane, które są zapisane w liście i zapisać je w pliku. W dowolnym momencie może zobaczyć jakie dane zapisuje do pliku i jakie informacje posiada w pliku. Każdy rodzaj pojazdu zapisuje się w oddzielnym pliku, co upraszcza odfiltrowanie pojazdów ze względu na typ. Pliki z danymi mogą posłużyć jako tabele w bazie danych. Zaleca się język Java.

1. **Specyfikacja wymagań**
   1. **Wymagania funkcjonalne**

* Użytkownik może dodawać pojazdy z rozróżnieniem na rodzaje pojazdów .
* Użytkownik może usuwać wprowadzone pojazdy przed zapisem do pliku.
* Użytkownik ma dostęp do danych wprowadzanych jak i do zapisanych w plikach.
* Użytkownik może dodawać i usuwać dowolną liczbę pojazdów (przeglądając ich zawartość).
* Użytkownik posiada uprawnienia do przeglądania wszystkich plików jak i modyfikacji dodawanych danych, zawartość plików kasuje się po wyjściu z programu i ponownym otworzeniu programu i dodaniu nowych danych w celu bezpieczeństwa danych.
* Nie można wprowadzać danych, które są niezgodne ze wzorcem.
  1. **Wymagania niefunkcjonalne**
* Możliwość dodawania, usuwania rekordów pliku, oraz odczyt i zapis rekordów pliku (wszystko z poziomu użytkownika).
* Aplikacja jest przyjazna dla klienta i jego rodziny, oraz jest bardzo prosta w użyciu.
* Aplikacja działa na systemach, które posiadają środowisko programistyczne, np. IntelliJ IDEA.
* Aplikacja tworzona jest w języku Java.
* Aplikacja zapisuje i odczytuje dane do/z plików.

1. **Diagram przypadków użycia**



Rysunek 1. Diagram przypadków użycia

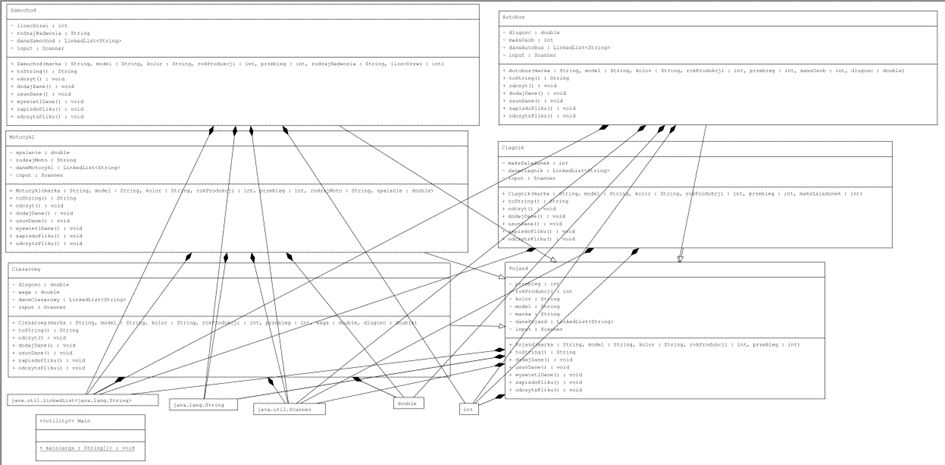
1. **Harmonogram realizacji projektu (diagram Gantta)**



Rysunek 2. Diagram Gantta

1. **Opis techniczny projektu**

* Środowisko programistyczne Javy: Java JDK Kit 7u79
* Ze względów na liczbę szczegółów aplikacja była projektowana na urządzenia typu laptop ‘Lenovo G570’.

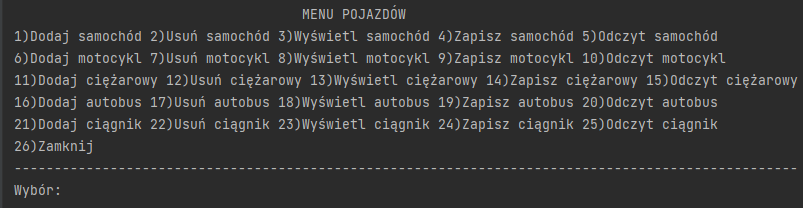


Rysunek 3. Diagram klas projektowanej aplikacji

1. **Prezentacja warstwy użytkowej projektu**

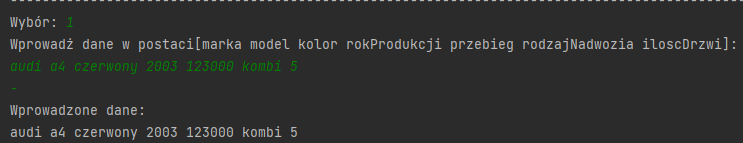
Na rysunku 4 przedstawiono główne okno aplikacji. Użytkownikowi po uruchomieniu aplikacji ukazuje się ekran startowy na którym może:

* Dodać dowolny pojazd (z rozróżnieniem na typy pojazdów).
* Usunąć dowolny pojazd (z rozróżnieniem na typy pojazdów).
* Wyświetlić dowolny pojazd, który trzeba oczywiście wcześniej dodać, aby móc zobaczyć jego zawartość.
* Zapisać dane do pliku (z rozróżnieniem na typy pojazdów).
* Odczytać dane z pliku, jeśli jakieś się w nim znajdują (jeśli nie ma danych to trzeba je pierwsze dodać).



Rysunek 4. Główne okno aplikacji

Przy wyborze opcji dodania pojazdu ukazuje się okno, w którym musimy podać dane w przedstawionym formacie (wprowadzanie danych kończy znak ‘-’), po wprowadzeniu danych możemy przejść do usunięcia danych podając numer linii, którą chcemy usunąć (zaczynając od 0 - to pierwsza linia), Następnie możemy zobaczyć jakie dane aktualnie posiadamy w liście. Jeżeli jesteśmy pewni, że takie dane chcemy wprowadzić do pliku to wybieramy opcję zapisu do pliku. Dodatkowo możemy zobaczyć czy dane poprawnie zapisały się w pliku.



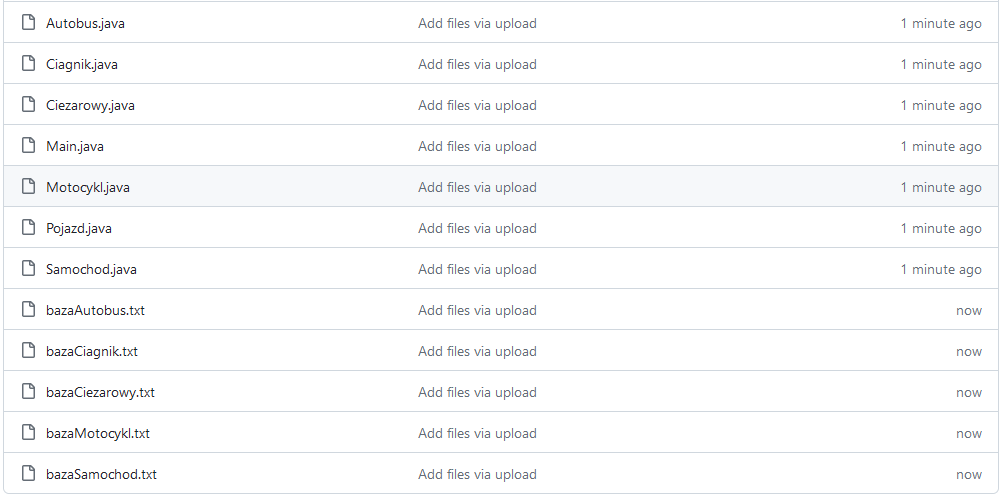
Rysunek 5. Przykład dodania pojazdu do pliku



Rysunek 6. Przykład zapisu danych do pliku

1. **System kontroli wersji**

Projekt realizowany był z wykorzystaniem systemu kontroli wersji Github, wszystkie pliki źródłowe projektu znajdują się pod adres: **https://github.com/arek123456/Programowanie-obiektowe.git**. Na rysunku 7 przedstawiono zrzut ekranu pokazujący zawartość folderu projekt.



Rysunek 7. Zawartość folderu projekt

1. **Literatura**

Należy podać literaturę książki/podręczniki/artykuły lub link do stron www.

1. <https://www.samouczekprogramisty.pl>
2. <https://javappa.com/kurs-java>
3. <https://javastart.pl/baza-wiedzy/java-podstawy-jezyka>