**Dokumentacja Projektu - System Zarządzania Szkołą**

*Jakub Michalik*

**1. Wprowadzenie**

System Zarządzania Szkołą to aplikacja Java służąca do kompleksowego zarządzania danymi szkoły, w tym informacjami o klasach, uczniach, nauczycielach, przedmiotach i ocenach. Projekt składa się z dwóch części:

* **Moduł Core** - logika biznesowa aplikacji
* **Moduł GUI** - interfejs użytkownika oparty na bibliotece Swing

**2. Architektura Systemu**

**Struktura pakietów:**

src/

├── Core/ # Logika biznesowa

├── GuiApp/ # Interfejs użytkownika

└── Main.java # Klasa startowa

**Diagram klas (uproszczony):**

[School] 1--\* [Class]

[Class] 1--1 [Teacher] (wychowawca)

[Class] 1--\* [Student]

[Student] 1--\* [Subject]

[Subject] 1--1 [Teacher] (nauczyciel)

[Subject] 1--\* [Grade]

[Grade] 1--1 [Teacher] (wystawiający ocenę)

**3. Opis Klasy Głównej - Main.java**

Klasa startowa aplikacji zawierająca:

* Dwie możliwe ścieżki uruchomienia: tryb konsolowy (zakomentowany) i tryb GUI
* Metodę main() inicjującą aplikację Swing
* Pomocnicze metody do:
  + Wyświetlania statystyk (showStatisticsPanel())
  + Inicjalizacji danych testowych (initializeSchoolData())
  + Testowania funkcjonalności (test())

**4. Moduł Core - Klasy Biznesowe**

**4.1 Person.java**

* Klasa abstrakcyjna reprezentująca osobę
* Wspólne atrybuty:
  + name (imię)
  + surname (nazwisko)
  + age (wiek)
  + peselNumber (numer PESEL)

**4.2 Student.java**

* Rozszerza Person
* Dodatkowe atrybuty:
  + subjects (lista przedmiotów)

**4.3 Teacher.java**

* Rozszerza Person
* Dodatkowe atrybuty:
  + salary (wynagrodzenie)

**4.4 Class.java**

* Reprezentuje klasę szkolną
* Atrybuty:
  + name (nazwa klasy)
  + students (lista uczniów)
  + supervisingTeacher (wychowawca)
* Kluczowe metody:
  + getStudentByPeselNumber() - wyszukiwanie ucznia po PESEL
  + getAverageGrade() - obliczanie średniej klasy z przedmiotu

**4.5 Subject.java**

* Reprezentuje przedmiot szkolny
* Atrybuty:
  + name (nazwa przedmiotu)
  + grades (lista ocen)
  + leadingTeacher (nauczyciel prowadzący)
* Metoda getAverageGrade() oblicza średnią ocen z przedmiotu

**4.6 Grade.java**

* Reprezentuje pojedynczą ocenę
* Atrybuty:
  + grade (wartość oceny 1-6)
  + weight (waga oceny)
  + description (opis oceny)
  + author (nauczyciel wystawiający)
* Metoda getGradeWithWeight() zwraca ocenę pomnożoną przez wagę

**4.7 School.java**

* Reprezentuje szkołę
* Atrybuty:
  + name (nazwa szkoły)
  + classes (lista klas)
  + teachers (lista nauczycieli)
* Metody zarządzania:
  + addClass(), removeClass()
  + addTeacher(), removeTeacher()
  + getTeacherByPesel() - wyszukiwanie nauczyciela po PESEL

**4.8 SchoolManager.java**

* Główna klasa zarządzająca logiką biznesową
* Funkcjonalności:
  + Zarządzanie klasami (tworzenie, usuwanie)
  + Zarządzanie nauczycielami (zatrudnianie, zwalnianie)
  + Zarządzanie uczniami (zapisywanie, usuwanie, przenoszenie)
  + Zarządzanie przedmiotami i ocenami
  + Generowanie statystyk (średnie ocen)
  + Wyświetlanie informacji o szkole

**5. Moduł GUI - Interfejs Użytkownika**

**5.1 ClassManagementPanel.java**

* Panel do zarządzania klasami szkolnymi
* Elementy interfejsu:
  + Tabela z listą klas
  + Panel szczegółów klasy
  + Przyciski: Odśwież, Dodaj klasę, Usuń klasę
* Funkcjonalności:
  + Wyświetlanie listy klas z wychowawcami
  + Wyświetlanie szczegółów klasy (uczniowie, wychowawca)
  + Dodawanie nowych klas
  + Usuwanie istniejących klas

**6. Funkcjonalności Systemu**

**6.1 Zarządzanie klasami**

* Tworzenie nowych klas z przypisanym wychowawcą
* Usuwanie istniejących klas
* Przeglądanie szczegółów klas (lista uczniów, wychowawca)

**6.2 Zarządzanie nauczycielami**

* Dodawanie nowych nauczycieli (dane osobowe + wynagrodzenie)
* Usuwanie istniejących nauczycieli
* Przeglądanie listy nauczycieli

**6.3 Zarządzanie uczniami**

* Zapisywanie uczniów do klas
* Przenoszenie uczniów między klasami
* Usuwanie uczniów z klas
* Przeglądanie szczegółów ucznia (oceny, przedmioty)

**6.4 Zarządzanie przedmiotami**

* Dodawanie przedmiotów do klas
* Przypisywanie nauczycieli do przedmiotów

**6.5 Zarządzanie ocenami**

* Dodawanie ocen uczniom
* Określanie wagi i opisu ocen
* Automatyczne obliczanie średnich

**6.6 Statystyki**

* Obliczanie średniej ocen ucznia
* Obliczanie średniej ocen z przedmiotu w klasie
* Wyświetlanie ogólnych statystyk szkoły

**7. Uruchomienie Aplikacji**

1. Wymagania:
   * Java JDK 8 lub nowsza
2. Kompilacja i uruchomienie:

bash

Copy

Download

javac Main.java

java Main

1. Tryby pracy:
   * Tryb GUI (domyślny) - uruchamia interfejs graficzny
   * Tryb konsolowy (wymaga odkomentowania odpowiedniego kodu w Main.java)

**8. Testowanie**

Aplikacja zawiera rozbudowaną metodę test() w klasie Main.java, która:

1. Tworzy szkołę z przykładowymi danymi
2. Dodaje nauczycieli, klasy, uczniów i przedmioty
3. Przypisuje oceny uczniom
4. Demonstruje kluczowe funkcje:
   * Przenoszenie uczniów między klasami
   * Wyświetlanie szczegółów klas i uczniów
   * Obliczanie średnich ocen
   * Usuwanie uczniów

**9. Możliwości Rozbudowy**

1. Rozszerzenie modułu GUI o dodatkowe panele:
   * Zarządzanie nauczycielami
   * Zarządzanie uczniami
   * Zarządzanie ocenami
2. Implementacja eksportu danych do formatów (PDF, Excel)
3. Dodanie systemu uwierzytelniania użytkowników
4. Rozbudowa modułu statystyk o zaawansowane raporty

**10. Ograniczenia Systemu**

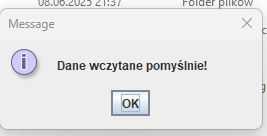
* Brak trwałego przechowywania danych (brak bazy danych)
* Ograniczona walidacja danych wejściowych
* Brak obsługi równoległych użytkowników

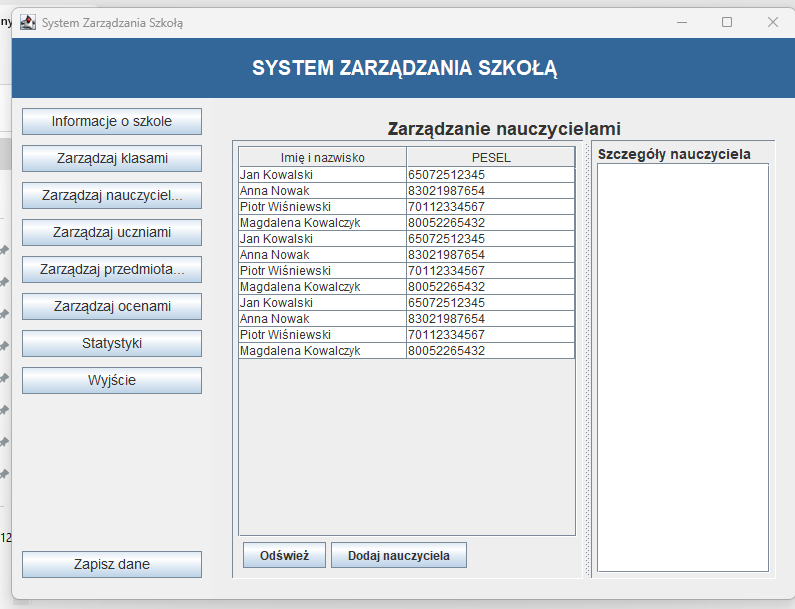
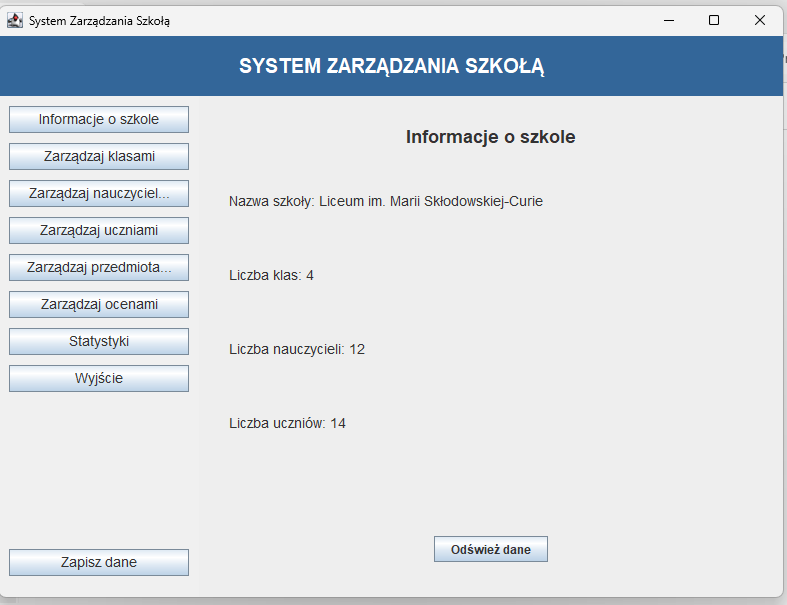
**11. Wnioski**

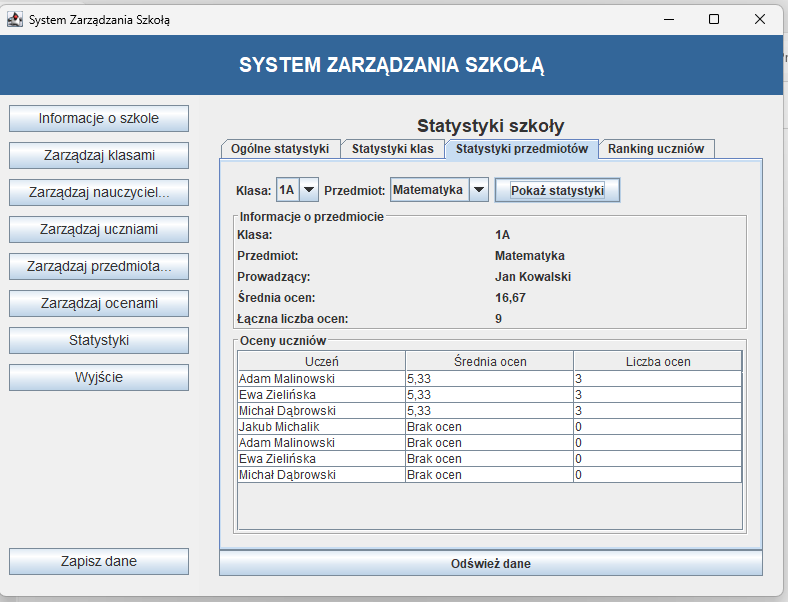
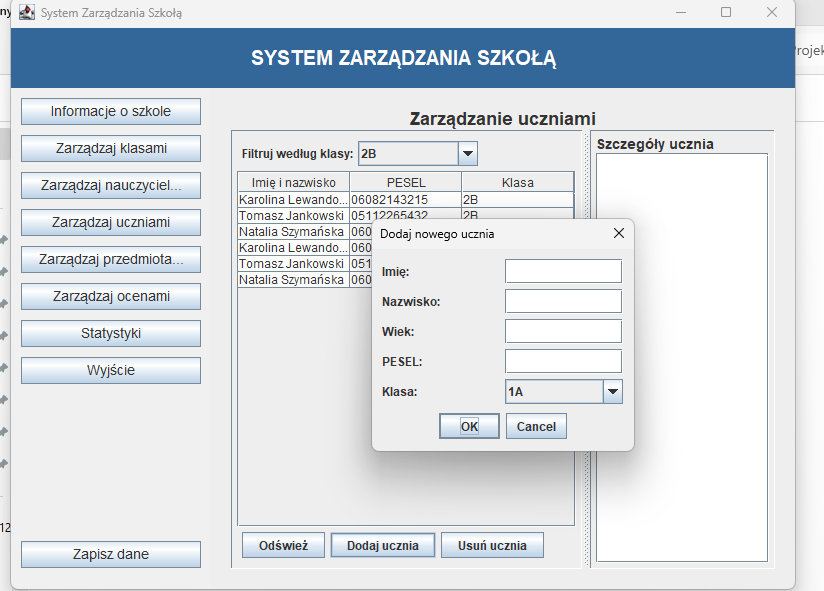
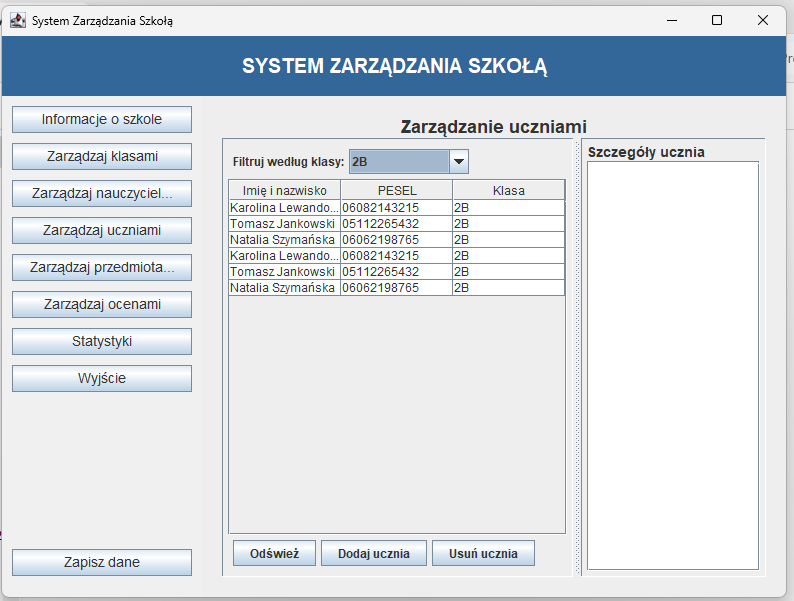
System Zarządzania Szkołą stanowi solidną podstawę do dalszego rozwoju. Jego modułowa architektura pozwala na łatwe rozszerzanie funkcjonalności. Obecna implementacja skutecznie rozwiązuje podstawowe problemy zarządzania danymi szkolnymi, a dzięki warstwowej strukturze umożliwia proste dodawanie nowych modułów w przyszłości.

**Przykładowe zrzuty ekranu:**

Tryb GUI:







Tryb konsolowy:

