

Dokumentacja Projektu

System Zarządzania Baza Danych Studentów

| | |
|---------|-------------------------|
| Kurs: | Programowanie Obiektowe |
| Autor: | Mateusz Konachowicz |
| Wersja: | 1.0 |
| Data: | 23.06.2025 |
| Jezyk: | Java |
| GUI: | Swing |
| Temat: | Baza Danych Studentow |

Spis Tresci

1. Wprowadzenie
2. Opis Zadania
3. Architektura Systemu
4. Implementacja Mechanizmow OOP
5. Dane Testowe w Systemie
6. Interfejs Uzytkownika
7. Instrukcja Uzytkownika
8. Podsumowanie

1. Wprowadzenie

Niniejszy dokument zawiera kompletną dokumentację projektu "System Zarządzania Bazą Danych Studentów", zrealizowanego w ramach kursu Programowanie Obiektowe. Projekt stanowi implementację systemu zarządzania informacjami o studentach, kursach i ocenach na uczelni.

Głównym celem projektu było stworzenie funkcjonalnej aplikacji z graficznym interfejsem użytkownika, która demonstruje praktyczne zastosowanie paradygmatów programowania obiektowego poznanych podczas kursu.

System realizuje następujące funkcjonalności:

- Zarządzanie studentami (dodawanie, edycja, usuwanie, wyszukiwanie)
- Zarządzanie kursami akademickimi
- System ocen z różnymi typami sprawdzianów
- Zapisywanie i wypisywanie studentów na kursy
- Obliczanie średniej ocen (GPA)
- Persystencja danych w plikach

2. Opis Zadania

2.1. Założenia Funkcjonalne

System realizuje następujące główne funkcjonalności:

Zarządzanie studentami:

- Dodawanie, edycja, usuwanie i wyszukiwanie studentów
- Przechowywanie danych osobowych i akademickich
- Sortowanie i filtrowanie według różnych kryteriów
- Zarządzanie zapisami studentów na kursy

Zarządzanie kursami akademickimi:

- Tworzenie i edycja kursów z punktami ECTS
- Ustawianie limitów studentów
- Przeglądanie listy zapisanych studentów

System ocen:

- Różne typy ocen (egzamin, kolokwium, projekt)
- Obliczanie średniej ocen (GPA)
- Historia ocen z datami i opisami

2.2. Ograniczenia Techniczne

Projekt został zrealizowany z następującymi ograniczeniami:

- Implementacja w języku Java
- Wykorzystanie biblioteki Swing do GUI
- Persystencja danych poprzez serializację
- Brak wykorzystania zewnętrznych bibliotek

3. Architektura Systemu

System został zaprojektowany zgodnie z zasadami programowania obiektowego, wykorzystując wzorzec Model-View-Controller (MVC). Architektura systemu jest modułowa, co zapewnia łatwość utrzymania i rozbudowy.

3.1. Struktura Pakietów

| Pakiet | Zawartosc | Opis |
|----------------------|--------------------------------|-----------------------|
| studentdb.models | Person, Student, Course, Grade | Modele danych |
| studentdb.services | StudentService, CourseService | Logika biznesowa |
| studentdb.gui | MainFrame, panele GUI | Interfejs użytkownika |
| studentdb.exceptions | Własne klasy wyjątków | Obsługa błędów |
| studentdb.utils | FileManager | Narzędzia pomocnicze |

3.2. Diagram Klas UML

Struktura klas systemu opiera się na hierarchii dziedziczenia z klasą abstrakcyjną Person na szczycie oraz specjalizowanymi klasami dla różnych typów użytkowników.

| Klasa | Atrybuty | Metody |
|----------------------------|---|-------------------------------|
| Person (abstrakcyjna) | firstName, lastName, email, birthDate | getFullName(), getRole() |
| Student extends Person | studentId, enrolledCourses, yearOfStudy, studyProgram | getProgram(), getYear() |
| Course | courseCode, courseName, description, credits | getDetails(), getCourseCode() |
| Grade | value, description, gradeType, date, letterGrade | getters/setters |
| StudentManagementException | message | konstruktor |
| StudentNotFoundException | extends StudentManagementException | konstruktor |
| CourseNotFoundException | extends StudentManagementException | konstruktor |

4. Implementacja Mechanizmow OOP

4.1. Realizacja Zalozen Projektowych

| Mechanizm OOP | Implementacja | Status |
|--------------------|--|--------------|
| Enkapsulacja | Prywatne pola z publicznymi metodami dostepowymi | Zrealizowane |
| Dziedziczenie | Hierarchia klas Person → Student | Zrealizowane |
| Polimorfizm | Metody toString() i getRole() | Zrealizowane |
| Klasy abstrakcyjne | Klasa abstrakcyjna Person | Zrealizowane |
| Interfejsy | Implementacja interfejsu Serializable | Zrealizowane |
| Kolekcje | Map, Set, List i operacje na nich | Zrealizowane |
| Wlasne wyjatki | Hierarchia wyjatkov systemu | Zrealizowane |
| Zasady SOLID | SRP i DIP w architekturze | Zrealizowane |
| GUI Swing | Kompletny interfejs uzytkownika | Zrealizowane |
| Persystencja | Serializacja obiektow do plikow | Zrealizowane |

5. Dane Testowe w Systemie

5.1. Studenci

| ID | Imie | Nazwisko | Email | Sem | Typ | Srednia |
|------|-----------|-------------|------------------------------|-----|----------|---------|
| S001 | Jan | Kowalski | j.kowalski@student.edu.pl | 3 | Stac. | 4.2 |
| S002 | Anna | Nowak | a.nowak@student.edu.pl | 5 | Stac. | 4.7 |
| S003 | Piotr | Wisniewski | p.wisniewski@student.edu.pl | 1 | Niestac. | 3.8 |
| S004 | Maria | Kaminska | m.kaminska@student.edu.pl | 7 | Stac. | 4.5 |
| S005 | Tomasz | Lewandowski | t.lewandowski@student.edu.pl | 2 | Niestac. | 4.0 |
| S006 | Katarzyna | Dabrowska | k.dabrowska@student.edu.pl | 4 | Stac. | 4.3 |

5.2. Kursy

| Kod | Nazwa | ECTS | Max | Opis |
|--------|------------------------------|------|-----|--|
| PO001 | Programowanie Obiektowe | 6 | 50 | Podstawy programowania w Javie |
| BD001 | Bazy Danych | 5 | 45 | Projektowanie i implementacja baz danych |
| ALG001 | Algorytmy i Struktury Danych | 6 | 40 | Zaawansowane algorytmy |
| MAT001 | Matematyka Dyskretna | 4 | 60 | Podstawy matematyki dla informatykow |
| WEB001 | Technologie Webowe | 5 | 35 | HTML, CSS, JavaScript |

6. Interfejs Uzytkownika

6.1. Glowne Okno Aplikacji

Glowne okno aplikacji zawiera pasek menu oraz panel z zakladkami dla poszczegolnych funkcjonalnosci:

- Zakladka 'Studenci' - do zarzadzania studentami
- Zakladka 'Kursy' - do zarzadzania kursami
- Zakladka 'Oceny' - do zarzadzania ocenami

6.2. Panel Zarzadzania Studentami

Panel studentow zawiera:

- Tabele studentow wyswietlajaca dane wszystkich studentow
- Formularz do dodawania i edycji studentow
- Przyciski do zarzadzania studentami (Dodaj, Edytuj, Usun)
- Pole wyszukiwania do filtrowania studentow
- Przycisk 'Zarzadzaj kursami' do zarzadzania zapisami studenta na kursy

6.3. Panel Zarzadzania Kursami

Panel kursow zawiera:

- Liste kursow
- Formularz do dodawania i edycji kursow
- Przyciski do zarzadzania kursami (Dodaj, Edytuj, Usun)
- Przycisk 'Pokaz studentow' do wyswietlania listy studentow zapisanych na kurs

6.4. Panel Zarzadzania Ocenami

Panel ocen zawiera:

- Pola wyboru studenta i kursu
- Tabele ocen wyswietlajaca oceny wybranego studenta z wybranego kursu
- Formularz do dodawania nowej oceny
- Przycisk do obliczania sredniej ocen (GPA)

7. Instrukcja Użytkownika

7.1. Uruchomienie Aplikacji

Aby uruchomic aplikacje, nalezy:

1. Upewnic sie, ze zainstalowana jest Java w wersji 8 lub nowszej
2. Uruchomic plik JAR lub skompilowac i uruchomic klase Main
3. Po uruchomieniu aplikacji zostanie wyswietlone glowne okno z trzema zakladkami

7.2. Zarzadzanie Studentami

Dodawanie nowego studenta:

1. Przejdz do zakladki 'Studenci'
2. Wypelnij formularz danymi studenta
3. Kliknij przycisk 'Dodaj'

Edycja danych studenta:

1. Zaznacz studenta w tabeli
2. Kliknij przycisk 'Edytuj'
3. Zmodyfikuj dane w formularzu
4. Kliknij przycisk 'Zapisz'

7.3. Zarzadzanie Kursami

Dodawanie nowego kursu:

1. Przejdz do zakladki 'Kursy'
2. Wypelnij formularz danymi kursu
3. Kliknij przycisk 'Dodaj'

7.4. Zarzadzanie Ocenami

Dodawanie nowej oceny:

1. Przejdz do zakladki 'Oceny'
2. Wybierz studenta z listy rozwijalnej
3. Wybierz kurs z listy rozwijalnej
4. Wypelnij formularz danymi oceny (wartosc, typ oceny, data)
5. Kliknij przycisk 'Dodaj ocene'

8. Podsumowanie

8.1. Realizacja Założeń Projektowych

Projekt "System Zarządzania Baza Danych Studentów" spełnia wszystkie założenia i wymagania postawione w ramach kursu Programowanie Obiektowe. System w pełni realizuje koncepcje programowania obiektowego, wykorzystuje bibliotekę Swing do utworzenia funkcjonalnego interfejsu użytkownika oraz implementuje persystencję danych poprzez serializację.

8.2. Możliwości Dalszego Rozwoju

System może być rozbudowany o dodatkowe funkcjonalności, takie jak system raportowania, powiadomienia czy też integracja z zewnętrzną bazą danych. Obecna architektura umożliwia łatwą rozbudowę i modyfikację bez naruszania istniejącego kodu.

8.3. Wnioski Końcowe

Projekt stanowi kompletną implementację systemu zarządzania danymi studenckimi z wykorzystaniem najważniejszych mechanizmów programowania obiektowego. Wszystkie zaplanowane funkcjonalności zostały zrealizowane zgodnie z wymaganiami kursu.