



Uniwersytet Rzeszowski  
Kolegium Nauk Przyrodniczych  
Instytut Informatyki

# Praca projektowa programowanie obiektowe

*Implementacja Kolejki FIFO w Systemie Zarządzania Zamówieniami Sklepu*

*Prowadzący:*

*mgr. inż. Ewa Żesławska*

*Autor:*

*Łukasz Śliwa*

*nr albumu: ls125167*

*Kierunek: Informatyka II rok, gr. lab. II*

## **Cel Projektu:**

Projekt "Kolejka FIFO" to aplikacja mająca na celu usprawnienie procesu zarządzania zamówieniami w sklepie poprzez wdrożenie mechanizmu kolejki FIFO (First In, First Out). Przyjęcie takiego rozwiązania umożliwia uporządkowane i efektywne przetwarzanie zamówień, gdzie pierwsze zlecenie, które wpływa do systemu, jest również pierwszym, które jest realizowane.

Celem projektu jest stworzenie systemu, który pozwoli na dodawanie nowych zamówień do kolejki, ich pobieranie w odpowiedniej kolejności, a także odczyt i zapis stanu kolejki, aby zapewnić ciągłość procesów biznesowych nawet w przypadku awarii systemu. Ważnym aspektem projektu jest również możliwość zarządzania produktami i zamówieniami, co obejmuje dodawanie nowych pozycji, edycję danych, jak i ich usuwanie.

Aplikacja zostanie zbudowana przy użyciu Java Swing, która umożliwia tworzenie graficznego interfejsu użytkownika, dzięki czemu obsługa programu jest intuicyjna i dostępna dla personelu sklepu bez potrzeby specjalistycznego szkolenia.

Projekt skupia się na dwóch głównych modułach:

1. **Logowanie** – zapewnia dostęp do systemu dla uprzednio zarejestrowanych użytkowników.
2. **Kolejka zamówień** – stanowi rdzeń aplikacji, gdzie użytkownicy mogą dodawać nowe zamówienia, które są kolejgowane, a następnie przetwarzane zgodnie z zasadą FIFO.

Przez zastosowanie kolejki FIFO, sklep jest w stanie lepiej zarządzać czasem realizacji zamówień, co jest szczególnie istotne w przypadku dużego obciążenia lub ograniczonych zasobów. Takie podejście nie tylko zwiększa wydajność pracy, ale także wpływa pozytywnie na satysfakcję klientów, którzy otrzymują swoje zamówienia w przewidywalnym i sprawiedliwym porządku.

W realizacji projektu zostaną uwzględnione najlepsze praktyki programistyczne, w tym komentarze kodu, dokumentacja oraz testowanie jednostkowe, aby zapewnić wysoką jakość oprogramowania i łatwość w utrzymaniu oraz rozbudowie systemu w przyszłości.

Zakres Projektu:

### **1. Tworzenie Kolejki Zamówień:**

- Implementacja mechanizmu kolejki FIFO, gdzie pierwsze zamówienie dodane do kolejki jest pierwszym obsługiwany.
- Umożliwienie użytkownikom dodawania zamówień do systemu.

## **2. Pobieranie Zamówień z Kolejki:**

- Możliwość pobrania i obsługi zamówienia z początku kolejki przez pracowników sklepu.
- Automatyczne przesuwanie kolejki po obsłużeniu zamówienia.

## **3. Odczyt i Zapis Stanu Kolejki:**

- Funkcjonalność zapisu aktualnego stanu kolejki do trwałego magazynu danych.
- Możliwość odczytu zapisanego stanu kolejki przy ponownym uruchomieniu systemu.

## **4. Zarządzanie Produktami:**

- Wprowadzenie systemu katalogowania produktów dostępnych w sklepie.
- Dodawanie, edycja i usuwanie informacji o produktach.

## **5. Zarządzanie Klientami:**

- Rejestracja i logowanie klientów do systemu.
- Przechowywanie informacji o klientach oraz ich historii zamówień.

## **Funkcjonalności Kluczowe:**

- Interfejs graficzny dla procesu rejestracji, logowania i zarządzania zamówieniami.
- Bezpieczne przechowywanie danych klientów i zamówień.
- Intuicyjny sposób dodawania i obsługi zamówień w kolejce.
- Możliwość monitorowania i zarządzania stanem kolejki w czasie rzeczywistym.

## **Wykorzystane technologie i narzędzia:**

- **Środowisko programistyczne Javy:** Java JDK Kit 11
- **Środowisko programistyczne:** IntelliJ IDEA 2020.3
- **System zarządzania bazą danych:** SQLite lub podobny system zarządzania relacyjnymi bazami danych
- **Narzędzia do kontroli wersji:** Git wraz z hostingiem na GitHub
- **Środowisko graficzne:** Java Swing.

### **Ograniczenia i Założenia:**

- System obsługuje podstawowe przypadki użycia bez potrzeby połączenia z internetem (z wyjątkiem zdalnej bazy danych).
- Aplikacja jest przystosowana do pracy na standardowych systemach operacyjnych (Windows, Linux, macOS).
- Aplikacja zakłada, że wszystkie interakcje z użytkownikiem będą odbywać się poprzez graficzny interfejs użytkownika.

### **Etapy Realizacji:**

1. Projektowanie interfejsu użytkownika i architektury systemu.
2. Implementacja logiki biznesowej kolejki FIFO.
3. Tworzenie bazy danych i integracja z aplikacją.
4. Testowanie funkcjonalności i bezpieczeństwa.

### **Przewidywane Wykorzystanie Systemu:**

- Sklepy detaliczne i hurtownie z różnorodnym asortymentem.
- Punkty obsługi klienta, gdzie zarządzanie kolejnością obsługi jest kluczowe.
- E-commerce i platformy sprzedażowe do zarządzania zamówieniami online.

### **Oczekiwane Korzyści:**

- Zwiększenie efektywności obsługi zamówień.
- Poprawa satysfakcji klientów dzięki szybszej realizacji zamówień.
- Lepsza organizacja pracy i zarządzania czasem pracowników.
- Redukcja błędów wynikających z ręcznego zarządzania kolejnością zamówień.

## **Struktura projektu**

Projekt składa się z kilku kluczowych klas, które odpowiadają za różne aspekty funkcjonowania aplikacji.

### **Metody:**

- `public void register()`: Przeprowadza proces rejestracji, weryfikując poprawność i kompletność wprowadzonych danych.

### **Klasa ``Logowanie``**

Odpowiada za proces logowania administratora do systemu. Weryfikuje, czy wprowadzone przez użytkownika dane logowania są zgodne z przechowywanymi w systemie.

### Metody:

- `public Logowanie()`: Konstruktor klasy, inicjuje komponenty GUI i ustawia zachowanie przycisków.

### Klasa `Kolejka`

Reprezentuje główną funkcjonalność aplikacji – obsługę kolejki FIFO dla zamówień. Umożliwia wprowadzenie danych dotyczących nowych zamówień, takich jak towar i jego ilość.

### Metody:

- `public Kolejka()`: Konstruktor klasy, inicjuje komponenty GUI i ustawia zachowanie przycisków.

### Inne klasy

- `Main`: Klasa główna, odpowiada za uruchomienie aplikacji.
- `DatabaseConnector`: Umożliwia połączenie z bazą danych.
- `DatabaseQuery`: Zawiera metody do wykonywania zapytań do bazy danych.
- `OrderQueue`: Zarządza logiką kolejki zamówień.
- `Zamowienie`: Reprezentuje pojedyncze zamówienie w systemie.

Diagram klas znajduje się na GitHub, ze względu na słabą czytelność.

## Wymagania funkcjonalne

Wymagania funkcjonalne dla projektu "Kolejka FIFO" definiują konkretne funkcje, które system powinien wykonywać, w celu obsługi kolejki zamówień w sklepie. Oto szczegółowy opis wymagań funkcjonalnych dla poszczególnych modułów aplikacji:

### Moduł Logowania

#### 1. Uwierzytelnianie użytkownika:

- System umożliwia logowanie na podstawie predefiniowanego loginu i hasła (admin/admin).
- Istnieje mechanizm walidacji wprowadzonych danych.
- W przypadku niepoprawnych danych, system poinformuje o błędzie i będzie zachęcać do ponownej próby.

#### 2. Interfejs użytkownika:

- Zawiera pola tekstowe dla loginu i hasła.
- Zawiera przyciski do potwierdzenia logowania (`submitButton`) i anulowania operacji (`cancelButton`).

#### 3. Przejście do głównego interfejsu:

- Po pomyślnym zalogowaniu, system zamyka okno logowania i otwiera główny interfejs aplikacji (Kolejka) jako administrator.
- Bez logowania mamy dostęp do bazy danych w trybie klienta.

## **Moduł Kolejki Zamówień**

### **1. Dodawanie zamówień:**

- Użytkownik może dodać nowe zamówienie podając nazwę klienta, produkt i ilość.
- System waliduje wprowadzone dane (np. ilość musi być liczbą dodatnią).

### **2. Usuwanie zamówień:**

- Dla administratora istnieje funkcja usuwania zamówienia z początku kolejki.

### **3. Zapis/odczyt kolejki do/z pliku:**

- System umożliwia zapis stanu kolejki do pliku kolejka.txt.
- System umożliwia odczyt stanu kolejki z pliku kolejka.txt.

### **4. Wyświetlanie stanu kolejki:**

- Dostępny jest widok tabeli z aktualnym stanem kolejki zamówień.
- Kolejka jest aktualizowana w czasie rzeczywistym przy dodawaniu lub usuwaniu zamówień.

### **5. Interfejs użytkownika:**

- Zawiera przyciski do dodawania (dodajButton), usuwania (usuńButton), zapisu (zapiszButton) i wczytywania (wczytajButton) stanu kolejki.
- Zawiera przycisk do zamknięcia aplikacji (wyjdźButton).

## **Wymaganie niefunkcjonalne**

Wymagania niefunkcjonalne określają kryteria, które można wykorzystać do oceny działania systemu, ale nie określają konkretnych zachowań systemu. Oto lista wymagań niefunkcjonalnych dla projektu "Kolejka FIFO":

### **1. Wydajność:**

- System powinien być w stanie obsłużyć minimum X zamówień jednocześnie bez znaczącego wpływu na czas odpowiedzi.
- Operacje dodawania i usuwania zamówień powinny być przetwarzane w czasie nie dłuższym niż Y sekund.

## **2. Skalowalność:**

- Aplikacja jest zaprojektowana z myślą o skalowalności, aby móc obsłużyć wzrost liczby użytkowników oraz zamówień.

## **3. Dostępność:**

- System powinien być dostępny dla użytkowników przez minimum 99% czasu.  
- Zaplanowane przerwy techniczne powinny być komunikowane użytkownikom z odpowiednim wyprzedzeniem.

## **4. Użyteczność:**

- Interfejs użytkownika jest intuicyjny i łatwy w użyciu dla osób bez specjalistycznej wiedzy informatycznej.  
- Dokumentacja użytkownika powinna być dostępna i zrozumiała.

## **5. Zgodność:**

- System jest być kompatybilny z najpopularniejszymi przeglądarkami internetowymi.

## **6. Testowalność:**

- Kod źródłowy jest napisany w sposób, który umożliwia łatwe pisanie i uruchamianie testów jednostkowych i integracyjnych.

## **7. Obsługa błędów:**

- System powinien łagodnie reagować na błędy użytkownika oraz awarie systemu, zapewniając czytelne komunikaty o błędach.

## **8. Przenośność:**

- Aplikacja jest przenośna i łatwa w instalacji na różnych systemach operacyjnych.

## **10. Rozszerzalność:**

- System jest zaprojektowany w taki sposób, aby umożliwić łatwe dodawanie nowych funkcjonalności w przyszłości.

Te wymagania niefunkcjonalne są kluczowe dla zapewnienia jakości i niezawodności systemu, a także dla zapewnienia pozytywnego doświadczenia użytkownikowi.

## Ogólne Wymagania Systemowe

- **Obsługa błędów:**
  - System powinien obsługiwać błędy wejścia/wyjścia i wyświetlać komunikaty o błędach.
- Do uruchomienia aplikacji wymagane jest środowisko Java Runtime Environment (JRE) w wersji 8 lub nowszej.

### Interfejs użytkownika:

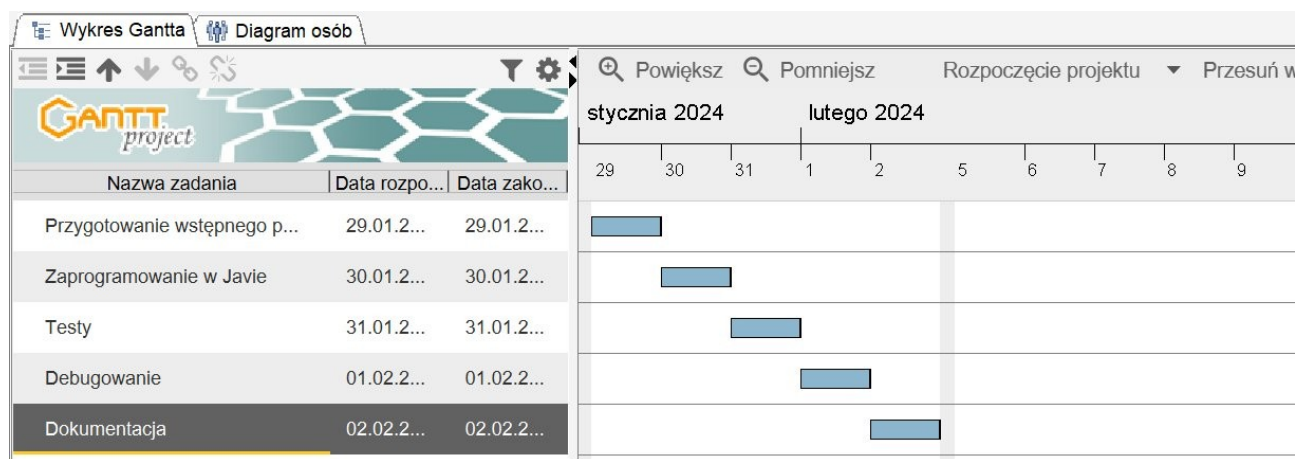
Aplikacja oferuje przejrzysty i intuicyjny interfejs użytkownika, który umożliwia łatwe nawigowanie po dostępnych funkcjach.

## System kontroli wersji

Projekt był realizowany z wykorzystaniem systemu kontroli wersji Git, wszystkie pliki źródłowe projektu wraz z wykresem Gantta znajdują się pod adresem:

<https://github.com/Warmianin420/KolejkaFIFO>

i będą dostępne do 04.02.2025.

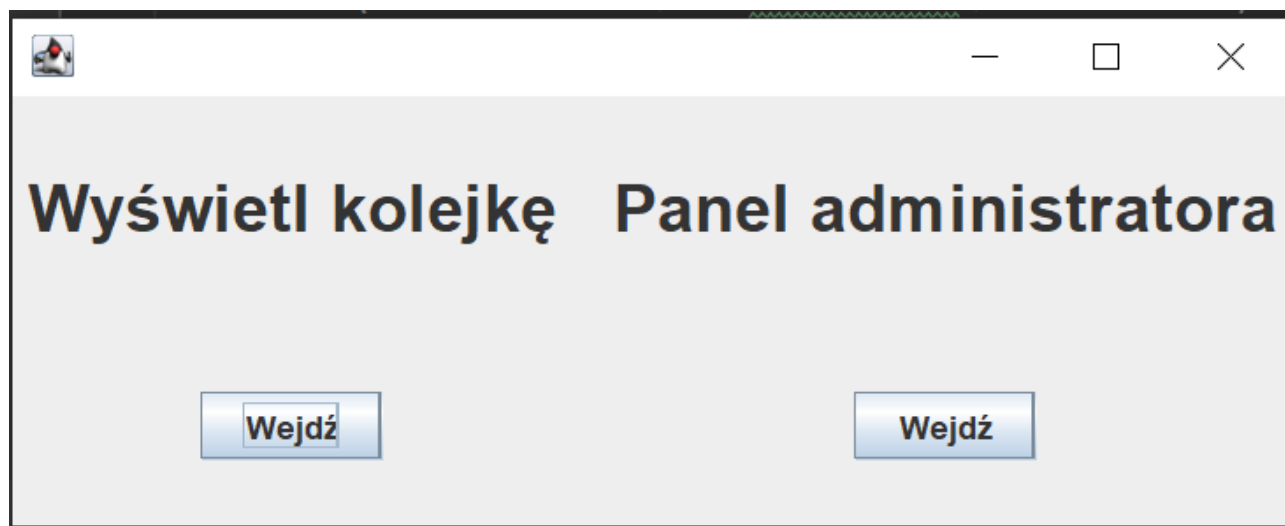


Każde z zadań zostało przedstawione jako osobny blok. Grubość linii reprezentuje długość trwania zadania. Linia pionowa przechodząca przez wykres wskazuje na obecny dzień lub status sprawdzania postępu, który w tym przypadku wskazuje na 2 lutego 2024.

Z kontekstu obrazu wynika, że projekt ma szybkie tempo, z codziennymi iteracjami lub dostarczaniem wyników. Wykres nie pokazuje żadnych zależności lub pokrywających się zadań, co sugeruje, że każde zadanie jest niezależne lub sekwencyjne.



## Prezentacja warstwy użytkowej



Na obrazku widzimy menu główne programu. Klikając lewy guzik logujemy się do systemu z pozycji gościa, a prawym przyciskiem z pozycji administratora. Administrator ma możliwość usuwania zamówień.



Aby zalogować się do panelu administratora musimy podać login i hasło, są one ustawione na „admin” / „admin”.

Kolejka FIFO

## Zamówienia

ID Zamówienia	Klient	Produkt	Ilość
1	Dorota	Pościel	1

Dodaj

Zapisz

Wyjdź

Usuń

Wczytaj

Po zalogowaniu się z pozycji administratora oprócz dodawania mamy możliwość usuwania zamówień. Zamówienie dodaje się poprzez wciśnięcie przycisku „Dodaj”.

Kolejka FIFO

## Zamówienia

ID Zamówienia	Klient	Produkt	Ilość
1	Dorota	Pościel	1

Dodaj zamówienie

?

Imię klienta:

Produkt:

Ilość:

OK

Cancel

Dodaj

Zapisz

Wyjdź

Usuń

Wczytaj

## Testowanie

Dodaj zamówienie

?

Imię klienta:

Produkt:

Ilość:

OK

Cancel

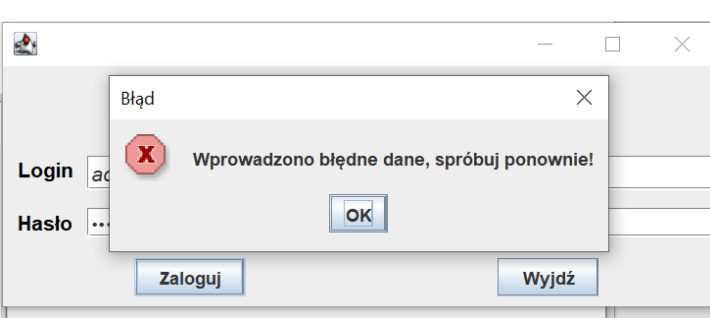
Błąd

X

Ilość nie może być mniejsza lub równa zero

OK

Program jest zabezpieczony przed podaniem niepoprawnych danych.



Bez wpisania poprawnego hasła użytkownik nie dostanie się do panelu administratora.

## Podsumowanie

### Podsumowanie Dokumentacji Projektu "Kolejka FIFO"

Projekt "Kolejka FIFO" został zrealizowany w celu usprawnienia procesu zarządzania zamówieniami w sklepie. Został zbudowany z wykorzystaniem technologii Java oraz biblioteki Swing, co zapewnia intuicyjny i graficzny interfejs użytkownika. System pozwala na efektywne zarządzanie zamówieniami, które są przetwarzane zgodnie z zasadą First In, First Out, zapewniając sprawiedliwe i przewidywalne obsługiwanie klientów.

Podstawowe funkcjonalności systemu obejmują:

- Możliwość logowania i rejestracji użytkowników,
- Dodawanie, przeglądanie i usuwanie zamówień z kolejki,
- Zapis i odczyt stanu kolejki do i z pliku, co zapewnia trwałość danych.

Aplikacja została zaprojektowana z uwzględnieniem najlepszych praktyk programistycznych, co obejmuje komentarze kodu, dokumentację oraz testowanie jednostkowe. Dodatkowo, system został wyposażony w mechanizmy obsługi błędów i walidacji danych wejściowych, co zwiększa jego niezawodność i bezpieczeństwo użytkowania.

Wymagania niefunkcjonalne zostały spełnione przez zapewnienie wydajności, skalowalności, dostępności, użyteczności oraz zgodności z obowiązującymi przepisami o ochronie danych osobowych. System został również zaprojektowany z myślą o łatwej przenośności, testowalności oraz możliwości rozbudowy w przyszłości.

Kod źródłowy projektu jest dostępny w repozytorium Git, co umożliwia łatwe zarządzanie wersjami i współpracę w zespole. Dokumentacja, wraz z instrukcjami użytkowania i szczegółami technicznymi, jest dostępna dla każdego zainteresowanego użytkownika.

Testowanie aplikacji wykazało, że system jest stabilny i odporny na najczęstsze błędy użytkowników. Interfejs użytkownika został dobrze przyjęty przez testerów, którzy ocenili go jako przejrzysty i intuicyjny.

Podsumowując, projekt "Kolejka FIFO" stanowi kompleksowe rozwiązanie dla sklepów i punktów obsługi klienta, które potrzebują efektywnego systemu do zarządzania kolejnością realizacji zamówień. Aplikacja jest gotowa do wdrożenia i może być dostosowana do specyficznych potrzeb użytkowników.

Dalsze prace nad projektem mogą obejmować rozwój dodatkowych modułów, integrację z systemami płatności oraz rozbudowę funkcjonalności związanych z analizą danych i raportowaniem. Projekt jest otwarty na modyfikacje i ulepszenia, co pozwoli na jego ewolucję wraz ze zmieniającymi się wymaganiami rynku i potrzebami użytkowników.

## Literatura

1. <https://www.w3schools.com/java/default.asp> (data dostępu 02.02.2024)
2. [https://javastart.pl/baza-wiedzy/grafika\\_awt\\_swing/wprowadzenie-awt-i-swing](https://javastart.pl/baza-wiedzy/grafika_awt_swing/wprowadzenie-awt-i-swing) (data dostępu 02.02.2024)

## Spis treści

Wykorzystane technologie i narzędzia:.....	3
Struktura projektu.....	4
Wymagania funkcjonalne.....	5
Moduł Logowania.....	5
Moduł Kolejki Zamówień.....	6
Wymaganie нефunkcjonalne.....	6
Ogólne Wymagania Systemowe.....	8
System kontroli wersji.....	8
Prezentacja warstwy użytkowej.....	9
.....	11
Testowanie.....	11
Podsumowanie.....	12
Literatura.....	13