Obraz zawierający tekst, logo, Czcionka, wizytówka

Opis wygenerowany automatycznie

Kierunek studiów: **Informatyka**

Nazwisko i Imię: Szpytka Jakub, Knap Bartłomiej

Numer albumu: 118847, 116758

Semestr: IV

Grupa: 1

Studia Inżynierskie - Zaoczne

**Języki i Paradygmaty Programowania – sem. 4**

**Sprawozdanie z Projektu**

Prowadzący zajęcia: mgr inż. Krystian Gumiński

**Łódź 2024**

Spis treści

[Projekt 3](#_Toc165656307)

[Paszportyzacja Projektu 4](#_Toc165656308)

[Szkielet Programu 4](#_Toc165656309)

[Wymaganie nr 1 4](#_Toc165656310)

[Wymaganie nr 2 4](#_Toc165656311)

[Wymaganie nr 3 5](#_Toc165656312)

[Stworzenie Interfejsu Użytkownika 5](#_Toc165656313)

# Projekt

Zwrócił się do klient ze zleceniem stworzenia systemu archiwizacji dokumentów, który ma pomóc w katalogowaniu zbiorów i śledzeniu ich stanu.

Projekt powinien być oparty na paradygmacie programowania obiektowego. Wykorzystaj znane techniki jak klasy, interfejsy, metody, dziedziczenie itd. Wykorzystaj wzorce projektowe np.: budowniczy.

Wymagania:

1. System powinien być w stanie śledzić informacje o dokumencie: tytuł, rok, UUID, kategoria, miejsce przechowywania oraz liczba egz.
2. System powinien umożliwiać dodanie nowych dokumentów do spisu, usuwanie, modyfikowanie.
3. System powinien zawierać szukajkę po roku, tytule lub miejscu przechowywania
4. System powinien śledzić historię zmian dokumentu: kto ostatnio zmieniał dokument, kto wypożyczył i kiedy wróci do archiwów.
5. System powinien umożliwiać przeglądanie listy wszystkich dokumentów oraz użytkowników

## Paszportyzacja Projektu

### Szkielet Programu

Utworzenie szkieletu programu:

- Stworzenie klas **Dokument** i **Archiwum**.

- Klasa **Dokument** przechowuje informacje o dokumencie: **tytuł, rok, UUID, kategoria, miejsce przechowywania oraz liczba egzemplarzy**.

- Klasa **Archiwum** zarządza listą dokumentów i zawiera **metody dodawania, usuwania, modyfikowania i wyszukiwania dokumentów.**

### Wymaganie nr 1

Dodanie możliwości śledzenia informacji o dokumencie:

- Klasa **Dokument** została stworzona z atrybutami: **tytul, rok, kategoria, miejsce\_przechowywania, liczba\_egz oraz uuid.**

- Użyto modułu **uuid** do generowania **unikalnych** identyfikatorów **UUID dla każdego dokumentu.**

### Wymaganie nr 2

Implementacja funkcji **dodawania, usuwania i modyfikowania** dokumentów:

- Metoda **dodaj\_dokumen** w klasie **Archiwum** pozwala na dodanie nowego dokumentu do listy dokumentów.

- Metoda **usun\_dokument** usuwa dokument **na podstawie podanego UUID.**

- Metoda **modyfikuj\_dokument** modyfikuje informacje o dokumencie **na podstawie podanego UUID oraz nowych danych.**

### Wymaganie nr 3

Implementacja funkcji wyszukiwania dokumentów:

- Metoda **szukaj\_dokumentu** w klasie **Archiwum** wyszukuje dokumenty **na podstawie podanego parametru, takiego jak tytuł, rok lub miejsce przechowywania.**

### Stworzenie Interfejsu Użytkownika

Stworzenie interfejsu użytkownika w funkcji main:

- Użytkownik może wybrać jedną z dostępnych opcji: dodanie nowego dokumentu, usunięcie dokumentu, aktualizację informacji o dokumencie, wyszukanie dokumentu lub wyjście z programu.

- W zależności od wybranej opcji, użytkownik zostaje poproszony o podanie odpowiednich informacji lub wybranie dokumentu do usunięcia/modyfikacji.

- Na koniec programu wyświetlany jest komunikat pożegnalny, a użytkownik musi nacisnąć Enter, aby zakończyć działanie programu.