

Projekt: akademicka baza danych studentów

Zadanie

Zadaniem jest stworzenie od podstaw aplikacji, która będzie prostą akademicką "bazą danych", przechowującą informacje o studentach. Program powinien umożliwiać wykonywanie podstawowych operacji na danych, takich jak dodawanie, wyświetlanie, sortowanie, czy usuwanie rekordów.

Aplikację można napisać w dowolnym języku programowania (np. Python, Java, C#, C++, itp.).

Minimalne wymagania funkcjonalne:

Przechowywanie informacji o studentach

Każdy student powinien mieć następujące dane (wymyślone na użytek zadania):

- Imię
- Nazwisko
- Adres zamieszkania
- Numer indeksu
- Numer PESEL
- Płeć

Funkcjonalność aplikacji

Aplikacja musi posiadać następujące funkcjonalności:

- Dodawanie nowych studentów do bazy danych
- Wyświetlanie wszystkich studentów
- Wyszukiwanie studentów:
 - według nazwiska
 - według numeru PESEL
- Sortowanie bazy danych:
 - według numeru PESEL
 - według nazwiska
- Usuwanie studentów według numeru indeksu
- Walidacja poprawności numeru PESEL (należy upewnić się, że wprowadzony PESEL jest zgodny z zasadami numeracji)

Przechowywanie danych:

Program powinien umożliwiać zapis bazy danych do pliku oraz wczytywanie danych z pliku przy ponownym uruchomieniu aplikacji.

Wymagania dodatkowe

Modularność i rozszerzalność:

Program powinien być zorganizowany w taki sposób, aby jego rozwój w przyszłości był prosty. Przykładami rozszerzeń mogą być:

- Dodanie dodatkowych danych dla studentów, takich jak kierunek studiów, oceny lub zapisane przedmioty.
- Dodanie innych operacji na danych, np. aktualizacja informacji o studentach.

Dobra praktyka programistyczna

Należy stosować zasady modularności i enkapsulacji. Proszę zadbać o to, aby kod był dobrze zorganizowany, czytelny i łatwy do utrzymania. Na przykład:

- Każda funkcjonalność (dodawanie, sortowanie, wyszukiwanie itp.) powinna być umieszczona w osobnych modułach/funkcjach/metodach.
- Unikanie pisania kodu, który trudno będzie rozszerzyć o nowe funkcje.
- Należy zadbać o odpowiednią strukturę danych, aby w przyszłości łatwo było ją rozbudowywać.

Wskazówki techniczne

Kilka wskazówek i zasad:

- Walidacja PESEL - numer PESEL (wymyślony na użytek zadania – przykładowy generator online: <https://pesel.cstudios.pl/o-generatorze/generator-on-line>) ma ustaloną długość (11 cyfr) oraz określone zasady wyliczania sumy kontrolnej. Należy wykorzystać algorytm do weryfikacji poprawności numeru PESEL i zaimplementować go w programie.
- Pliki do przechowywania bazy danych możesz wykorzystać prosty format tekstowy (np. pliki CSV, JSON, XML). Proszę zweryfikować czy wczytywanie i zapis danych są poprawnie obsługiwane.
- Sortowanie i wyszukiwanie, można użyć wbudowanych funkcji sortowania i wyszukiwania dostępnych w wybranym języku programowania lub zaimplementować własne algorytmy.

Ocena projektu:

Projekt będzie oceniany na podstawie:

- Poprawności działania aplikacji (czy spełnia wszystkie wymagania funkcjonalne)
- Organizacji kodu (czy jest modularny, dobrze zaprojektowany i łatwy do rozbudowy)
- Jakości i czytelności kodu (czy kod jest dobrze napisany i zgodny z zasadami dobrej praktyki programistycznej)
- Zastosowania walidacji PESEL i obsługi plików
- Dodatkowe punkty zostaną przyznane za implementację funkcji wykraczających poza wymagania, np. dodanie nowych pól, funkcji lub zastosowanie bardziej zaawansowanych technik przechowywania danych (bazy danych).