

# Laboratorium programowania komputerów

Temat:  
Program do zarządzania bazą ocen uczniów

Autor: Krzysztof Stępień

Informatyka, Rok I, Grupa 1, Sekcja 2

Prowadzący: mgr inż. Tomasz Wojdyła

# 1. Temat

Napisać program obsługi szkolnej bazy danych umożliwiający dodawanie, usuwanie, przeglądanie oraz wyszukiwanie rekordów zawierających informacje o uczniach. Program ma posiadać możliwość logowania się na konta o różnych prawach dostępu do danych.

## 2. Analiza, projektowanie

### Algorytmy, struktury danych, ograniczenia specyfikacji.

Program zawiera dwie listy dwukierunkowe. Pierwsza lista zawiera rekord z danymi ucznia oraz wskaźniki na poprzedni i następny element listy. W drugiej liście znajduje się rekord przechowujący dane konta użytkownika oraz tak jak w przypadku pierwszej listy wskaźniki na poprzedni i następny element listy.

W programie wykorzystywane są także dwa typy tablic dynamicznych. Pierwszy przechowuje informacje o ocenach ucznia. Drugi przechowuje rekordy zawierające nazwę przedmiotu oraz informacje o ocenach z przedmiotu.

Wszystkie informacje wykorzystywane w bazie danych zawierają się w czterech plikach. W pierwszym pliku o nazwie „Dziennik” zapisywane są rekordy zawierające Id ucznia oraz informacje o nim (Imię, Nazwisko, Klasa). W pliku „Login” zapisywane są rekordy zawierające dane konta użytkownika(Login, Hasło, Imię, Nazwisko, Uprawnienie) oraz Id. W trzecim pliku o nazwie „Oceny” zapisane są informacje o ocenach ucznia(Ilość Ocen, z czego ocena została wystawiona 7 – 9 oraz ocena 1 – 6). W pliku czwartym „Przedmioty” zapisane są nazwy przedmiotów z których można wystawiać uczniowi oceny.

## 3. Specyfikacja zewnętrzna

### Obsługa programu.

Po uruchomieniu programu należy podać Login i Hasło konta na które chcemy się zalogować. To jakie menu zostanie wyświetlone i jakie operacje będzie mógł wykonywać użytkownik zależy od tego jakie uprawnienie posiada konto na które się logujemy. Istnieją trzy różne prawa dostępu do danych (Administradora, Nauczyciela i Rodzica). W przypadku gdy żadne konto jeszcze nie istnieje, zostanie utworzone automatycznie konto o loginie „admin” i hasle podanym przez użytkownika. Dane odczytywane są z plików zaraz po uruchomieniu programu, a zapisywane po wylogowaniu się użytkownika.

Operacje możliwe do wykonania po zalogowaniu się na konto administratora to: wypisanie danych uczniów, dodanie ucznia, wypisanie danych do logowania, dodanie konta nauczyciela, wyszukanie uczniów spełniających kryteria, edytowanie danych do logowania, usunięcie konta, dodanie konta rodzica, dodawanie przedmiotów szkolnych, edytowanie danych ucznia, usunięcie ucznia oraz wylogowanie.

Operacje możliwe do wykonania po zalogowaniu się na konto nauczyciela to: wypisanie danych uczniów, dodanie ucznia, wypisanie danych do logowania, wyszukiwanie uczniów spełniających kryteria, edytowanie danych ucznia, usunięcie ucznia oraz wylogowanie.

Operacje możliwe do wykonania po zalogowaniu się na konto rodzica to: wyświetlenie danych ucznia, wylogowanie.

Po wylogowaniu się użytkownika możemy podjąć decyzję o ponownym zalogowaniu się.

Po dodaniu nowego ucznia musimy stworzyć dla niego konto dla rodzica, nie ma możliwości stworzenia dwóch kont o takim samym loginie. Konto rodzica jest powiązane z danymi ucznia za pomocą Id. Do danych jednego ucznia można mieć dostęp logując się z różnych kont, jednak dodatkowe konto musi zostać utworzone przez administratora.

Po wyszukaniu ucznia spełniającego podane kryteria możemy usunąć lub edytować jego dane.

## Sterowanie:

- cyfry – wybieranie opcji z menu (np. od 0 – 9 w przypadku menu dla administratora).
- „p” – przerwanie przeglądania danych uczniów lub danych do zalogowania.
- „e” – edytowanie danych ucznia.
- „u” – usunięcie ucznia.
- „t” i „n” – podejmowanie decyzji.

## 4. Specyfikacja wewnętrzna

### 4.1 Zmienne globalne

- Uprawnienie – zmienna typu integer. Przechowuje numer uprawnienia zalogowanego użytkownika.
- NumerUcznia – zmienna typu integer. Przechowuje numer ucznia zalogowanego użytkownika.
- Id – zmienna typu integer, przechowuje najwyższy numer Id.
- GlowaDanychOsob – zmienna typu WskListaDanychOsob.
- GlowaDanychDoLog – zmienna typu WskListaDanychDoZalog.

## 4.2 Procedury

**procedure OdczytaniePrzedmiotowZPliku(var ListaPrzedmiotow: Przedmioty);**

Procedura odczytująca nazwy przedmiotów z pliku „Przedmioty” i dodające odczytane nazwy do dynamicznej tablicy rekordów.

**procedure ZapisaniePrzedmiotowDoPliku(ListaPrzedmiotow: Przedmioty);**

Procedura zapisująca nazwy przedmiotów z dynamicznej tablicy rekordów do pliku „Przedmioty”.

**procedure SprawdzenieCzyPrzedmiotIstnieje(PodanyPrzedmiot: string;**

Procedura sprawdza czy przedmiot który chcemy dodać już istnieje.

**procedure DodanieNowegoPrzedmiotu(var GlowaOsob: WskListaDanychOsob);**

Dodanie nowego przedmiotu kolejno wszystkim uczniom oraz zapisanie nazwy dodanego przedmiotu na końcu pliku „Przedmioty”.

**procedure UsunieciePrzedmiotu(var GlowaOsob: WskListaDanychOsob);**

Usuniecie podanego przedmiotu. Usuniecie przedmiotu powoduje skasowanie wszystkich ocen z usuniętego przedmiotu.

**procedure WyświetlenieOpcjiEdytowaniaPrzedmiotow;**

Wyświetlenie menu w którym znajdują się opcje dodania i usunięcia przedmiotu.

**procedure ZapisanieDanychDoLogowania(GlowaLog: WskListaDanychDoZalog);**

Zapisanie rekordów z informacjami o kontach użytkowników z listy do pliku „login”.

**procedure SprCzyPodanyLogIstnieje(sprlogin: string; GlowaLog: WskListaDanychDoZalog;**

Sprawdzenie czy podany przez użytkownika login znajduje na liście z danymi do zalogowania.

**procedure Logowanie;**

Procedura umożliwia zalogowanie się. Po zalogowaniu zostają zapamiętane Uprawnienie i NumerUcznia.

**procedure DodanieDanychDoZalogowania(NrUcznia: integer; Imie: string; Nazwisko: string; UprawnienieDoDodania: integer; var GlowaLog: WskListaDanychDoZalog);**

Utworzenie konta do logowania. Dodanie elementu na koniec listy.

**procedure WybranieOceny(var wybor: integer);**

Decydujemy czy podana ocena ma być z sprawdzianu, kartkowi lub odpowiedzi.

**procedure PodanieOcen(var Przedmiot: Oceny);**

Procedura dodaje lub usuwa ocenę z tablicy dynamicznej.

**procedure PodanieDanychOsoby(var tmp: WskListaDanychOsob; var przerwano: boolean);**

Podawanie danych o uczniu z możliwością podjęcia decyzji o przerwaniu wykonywania tej czynności.

**procedure DodanieOsobyDoListy(var GlowaOsob: WskListaDanychOsob);**  
Dodawanie ucznia na koniec listy dwukierunkowej.

**procedure WypisanieOcen(Przedmiot: Oceny);**  
Wypisywanie ocen z przekazanej tablicy dynamicznej.

**procedure WypisanieDanychOsoby(DaneUcznia: DaneOsoby);**  
Wypisanie danych ucznia z przekazanego rekordu.

**procedure WypisanieDanychZListy(GlowaOsob: WskListaDanychOsob);**

**procedure ZapisanieDanychOsobDoPliku(GlowaOsob: WskListaDanychOsob);**  
Zapisanie wszystkich informacji o uczniach z listy dwukierunkowej do plików „Dziennik” i „Oceny”.

**procedure OdczytanieDanychOsobZPliku(var GlowaOsob: WskListaDanychOsob);**  
Wczytanie danych z plików „Dziennik” i „Oceny” do listy dwukierunkowej zawierającej rekord z danymi ucznia i tablicą dynamiczną z ocenami.

**procedure UtworzenieKontaAdministratora;**  
Zapisanie rekordu z danymi konta administratora do pliku „login”. Login(„admin”) i Id podawane są automatycznie. Hasło musi zostać wprowadzone przez użytkownika.

**procedure OdczytanieDanychDoLogowania(var GlowaLog: WskListaDanychDoZalog);**  
Wczytanie rekordów z pliku „login” do kolejnych elementów listy dwukierunkowej.

**procedure UsuniecieDanychDoZalog(NrDoUsuniecia: integer; var GlowaLog: WskListaDanychDoZalog);**  
Usuniecie z listy danych do zalogowania elementu o Id = NrDoUsuniecia.

**procedure UsuniecieOsobyZListy(NrDoUsuniecia: integer; var GlowaOsob: WskListaDanychOsob);**  
Procedura usuwa z listy z danymi o uczniu element o Id = NrDoUsuniecia.

**procedure PodanieNowychDanych(var DaneDoEdycji: DaneOsoby; var DaneDoLog: DaneDoZalogowania);**  
Użytkownik zmienia wybrane dane w przekazanym do procedury rekordzie. W przypadku zmiany imienia i nazwiska automatycznie zostaje zmienione imię i nazwisko w koncie do logowania powiązanym z zmienianym rekordem.

**procedure SzukOsobySpelWarunek(nrdoedycji: integer; var GlowaOsob: WskListaDanychOsob; GlowaLog: WskListaDanychDoZalog);**  
Procedura wyszukuje w liście z danymi o uczniach osoby o Id = NrDoEdycji i wywołuje procedurę **PodanieNowychDanych**.

**procedure PodanieNowychDanychDoLogowania(var DaneDoEdycji: DaneDoZalogowania);**

Użytkownik zmienia wybrane dane w przekazanym do procedury rekordzie.

**procedure EdytowanieDanychDoLogowania(login: string; var GlowaLog: WskListaDanychDoZalog);**

Procedura wyszukuje w liście z danymi do zalogowania konta o podanym loginie i wywołuje procedurę **PodanieNowychDanychDoLogowania**.

**procedure EdytowanieAlboUsowanie(NrUczniaDoEdycji: integer);**

Użytkownik decyduje czy chce edytować lub usunąć z listy dane ucznia o Id = NrUczniaDoEdycji. Edytowanie lub usunięcie danych nie jest konieczne.

**procedure EdytowanieLubUsuwanieDanychDoLog(var KontoDoEdycji: DaneDoZalogowania);**

Użytkownik decyduje czy chce edytować lub usunąć z listy element z rekordem KontoDoEdycji. Edytowanie lub usunięcie danych nie jest konieczne.

**procedure WyszukiwanieOsob(WyszukiwanyTekst: string; SprawdzanyTekst: string; var Znalezione: boolean);**

Procedura sprawdza czy kolejne litery wyszukiwanego i sprawdzanego tekstu są sobie równe. W przypadku gdy wyszukiwany tekst jest dłuższy od tekstu sprawdzanego porównywanie nie jest konieczne.

**procedure WyświetlenieMenuWyszukiwania(GlowaOsob: WskListaDanychOsob);**

Użytkownik decyduje czy chce wyszukać osobę o podanym fragmencie imienia, nazwiska klasy, lub o podanym Id. Dla każdego elementu listy wywoływana jest procedura **WyszukiwanieOsob**.

**procedure WyświetlenieMenuAdministradora;**

**procedure WyświetlenieMenuDlaNauczyciela;**

**procedure WyświetlenieMenuDlaRodzica(GlowaOsob: WskListaDanychOsob;**

Zalogowany użytkownik decyduje jakie operacje chce wykonać.

**procedure WybranieMenu(Uprawnienie: integer);**

Procedura sprawdza uprawnienie zalogowanego użytkownika i wyświetla menu odpowiadające jego prawu dostępu do danych.

**procedure ZwolnieniePamieci;**

Procedura usuwa kolejne elementy listy z danymi o uczniach oraz z danymi do zalogowania.

## 4.3 Program główny

**begin**

clrscr;

GlowaDanychOsob := nil;

GlowaDanychDoLog := nil;

OdczytanieDanychDoLogowania(GlowaDanychDoLog);

OdczytanieDanychOsobZPliku(GlowaDanychOsob);

**repeat**

logowanie;

```
WybranieMenu(Uprawnienie);  
Writeln('Jesli chcesz zalogowac sie ponownie wcisnij T');  
until readkey <> 't';  
ZwolnieniePamieci;  
end.
```

## 5. Testowanie

Testowanie programu polegało na sprawdzeniu czy dane są odczytywane i zapisywane prawidłowo, czy program nie generuje błędów, czy jest odporny na podawanie nieprawidłowych danych oraz czy komunikaty wyświetlane są prawidłowo.

## 6. Wnioski

Zastosowanie listy dwukierunkowej okazało się bardzo dobrym wyborem przy tworzeniu prostej bazy danych, gdy nie jesteśmy w stanie przewidzieć ilości potrzebnych nam elementów. W odróżnieniu od tablic możemy w prosty sposób usuwać elementy z środka listy oraz dodawać nowe w dowolnym miejscu. Stosowanie list nie zapewni jednak szybkiego dostępu do wybranego elementu, w przypadku wyszukiwania konkretnego elementu musimy przejść z jednego końca listy na drugi, co przy liście zawierającej wiele elementów może okazać się czasochłonne.