Politechnika Śląska w Gliwicach

Wydział Automatyki, Elektroniki i Informatyki

Podstawy Programowania Komputerów

„Dziennik uczniów”

autor	Kamil Lewandowski
prowadzący	dr inż. Maciej Długosz
rok akademicki	2016/2017
kierunek	informatyka
rodzaj studiów	SSI
semestr	1
termin laboratorium / ćwiczeń	wtorek, 12:00-13:30
grupa	6
sekcja	1
termin oddania sprawozdania	2016-12-02
data oddania sprawozdania	2016-12-01

1 Treść zadania

Napisać program pełniący funkcję dziennika ocen. Powinien on przechowywać listę osób z przyporządkowanymi podstawowymi danymi osobowymi (imię, nazwisko, data i miejsce urodzenia) oraz listą uzyskanych przez daną osobę ocen. Każda ocena jest opisana wartością oraz krótką informacją tekstową. Program powinien mieć możliwość zapisu i odczytu osób wraz z ocenami z pliku binarnego, a także dodawania (według kolejności alfabetycznej nazwisk i imion), wyszukiwania, edycji i usuwania osób, dodawania ocen, a także wyświetlania całej bazy danych.

2 Analiza zadania

Zagadnienie przedstawia problemy operowania dynamicznymi strukturami danych. Należy opracować odpowiednie dodawanie usuwanie oraz edytowanie elementów. Istotne jest również odpowiednie sortowanie tych elementów.

2.1 Struktury danych

W programie wykorzystano strukturę listy. W każdym jej elemencie znajduje się głowa kolejnej listy. Ta pierwsza obsługuje dane uczniów. Kolejna z kolei przechowuje informacje o ocenach każdego z uczniów.

2.2 Algorytmy

W programie pojawiły się algorytmy pozwalające w sposób alfabetyczny dodawać uczniów, dodawać oceny oraz w odpowiedni sposób zapisywać dane i odczytywać je z pliku binarnego.

3 Specyfikacja zewnętrzna

Program może być uruchamiany z konsoli. Żadne dane nie są wymagane. Cała interakcja z użytkownikiem odbywa się podczas działania programu. W programie zostało wykorzystane proste menu. Za jego pomocą użytkownik może komunikować się z programem. Program

został stworzony z pewnymi założeniami, miedzy innymi takimi że imię i nazwisko zostanie wprowadzone przez użytkownika z dużej litery.

4. Specyfikacja wewnętrzna

Program został zrealizowany zgodnie z paradygmatem strukturalnym. W programie rozdzielono interfejs (komunikację z użytkownikiem) od logiki aplikacji(operowania na listach).

4.1 Typy zdefiniowane w programie

W programie zdefiniowano następujący typ:

```
//-----------------------------------------------------------------------------
```

```
1. struct Ocena  //zawiera informacje o ocenie
2. {
3. int    stopien;
4. string opis;
5. };
```

```
//-----------------------------------------------------------------------------
```

Typ ten przechowuje wartości opisujące oceny uczniów.

```
//-----------------------------------------------------------------------------
```

```
1. struct LOcena //element listy ocen
2. {
3. Ocena  ocena;
4. LOcena* nastO = nullptr; //wskaźnik do nastepnego elementu
5. };
```

```
//-----------------------------------------------------------------------------
```

Ten typ przechowuje strukturę typu ocena oraz wskaźnik na następny element listy. Umożliwia on stworzenie listy ocen.

```
//-----------------------------------------------------------------------------
```

```
1. struct Uczen //zawiera inforamcje o uczniu
2. {
3. string imie;
4. string nazwisko;
5. string data;
6. string miejsce;
7. int          iloscOcen = 0; //zmienna zawiera informacje pomocne przy
   operowaniu na liście ocen danego ucznia
8. LOcena* glowaOcen = nullptr; //wskazuje wierwszy element listy ocen
9. };
```

```
//-----------------------------------------------------------------------------
```

Typ ten zawiera wszystkie dane o uczniu oraz głowę listy ocen danego ucznia.

```
//-----------------------------------------------------------------------------
```

```
1.  struct LUczen //element listy uczniów
2.  {
3.  Uczen  uczen;
4.  LUczen*       nastU; //wskaźnik do nastepnego elementu
5.  };
```

```
//-----------------------------------------------------------------------------
```

Kolejny typ będący elementem listy. Przechowuje on informacje o uczniu oraz wskaźnik do następnego elementu listy.

```
//-----------------------------------------------------------------------------
```

```
1.  struct Dziennik //struktura przechowujaca wszystkie dane potrzebne do
    poprawnego funkcjonowania programu
2.  {
3.  LUczen* glowa = nullptr;
4.  int iloscUczniow = 0; //zmienna przechowujaca informacje o ilości uczniów
5.  int stan = 0;
6.  };
```

```
//-----------------------------------------------------------------------------
```

Typ przechowuje wszystkie informacje potrzebne do funkcjonowania programu.

4.2 Ogólna struktura programu

W głównej funkcji programu jest stworzona pętla ,która obsługuje wszystkie funkcjie. Stan w którym znajduje się program definiuje pole struktury Dziennik o nazwie stan.

```
//-----------------------------------------------------------------------------
```

```
1.  int main()
2.  {
3.  Dziennik dziennik = Dziennik();
4.  do
5.  {
6.  switch (dziennik.stan)
7.  {
8.  case 0:
9.  {
10. system("cls");
11. dziennik.stan = menuGlowne();
12. break;
```

```
13. }
14. case 1:
15. {
16. system("cls");
17. wyswietlDziennik(dziennik.glowa, 1);
18. dziennik.stan = 0;
19. break;
20. }
21. case 2:
22. {
23. system("cls");
24. dziennikDodajUcznia(dziennik.glowa);
25. dziennik.iloscUczniow++;
26. dziennik.stan = 0;
27. break;
28. }
29. case 3:
30. {
31. system("cls");
32. dziennikWybierzUcznia(dziennik.glowa);
33. dziennik.stan = 0;
34. break;
35. }
36. case 4:
37. {
38. wczytajDziennik(dziennik.glowa, dziennik.iloscUczniow);
39. dziennik.stan = 0;
40. break;
41. }
42. case 5:
43. {
44. zapiszDziennik(dziennik.glowa, dziennik.iloscUczniow);
45. dziennik.stan = 0;
46. break;
47. }
48. }


49. } while (dziennik.stan != 6);

50. usunDziennik(dziennik.glowa);

51. return 0;
52. }
```

```
//--------------------------------------------------------------------------------



//--------------------------------------------------------------------------------
```

```
1.  int menuGlowne(); // pelni funkcje menu
```

```
//--------------------------------------------------------------------------------
```

Ta funkcja obsługuje system menu. Za jej pośrednictwem użytkownik wybiera odpowiednią opcję. Z uwagi na specyfikację poszczególnych funkcji zostały one podzielone na takie które wymagają wskaźnika odpowiedniego ucznia oraz na takie które tego

wskaźnika nie potrzebują. Dlatego zostało stworzone jeszcze jedno menu oraz odpowiednia funkcja menuWyboru która je obsługuje.

4.3 Szczegółowy opis implementacji funkcji

W programie można znaleźć takie funkcje jak:

```
//-----------------------------------------------------------------------------
1. void dziennikDodajUcznia(LUczen*& glowa); //funkcja nadzorujaca dodawanie ucznia

//-----------------------------------------------------------------------------
```

Funkcja wczytuje z klawiatury dane i wysyła je do funkcji:

```
//-----------------------------------------------------------------------------
1.  void dodajUcznia(LUczen*& glowa, const Uczen uczen); //dodaje ucznia
    uwzgledniajac kolejnosc alfabetyczna

//-----------------------------------------------------------------------------
```

Ta z kolei dodaje kolejny element do listy w kolejności alfabetycznej biorąc pod uwagę nazwiska i imiona uczniów.

```
//-----------------------------------------------------------------------------
1. void dziennikWybierzUcznia(LUczen*& glowa);  // nadzoruje funkcje operujace na
   danym uczniu

//-----------------------------------------------------------------------------
```

Jest to funkcja która daje użytkownikowi możliwość wyboru odpowiedniego ucznia, za pomocą funkcji wybierzUcznia, po czym przenosi użytkownika do drugiego menu. Daje ono możliwość użycia tych funkcji.

```
//-----------------------------------------------------------------------------
1.  void dziennikEdytujUcznia(LUczen* uczen);

//-----------------------------------------------------------------------------
```

Funkcja zmieniająca dane wybranego wcześniej użytkownika za pomocą funkcji:

```
//-----------------------------------------------------------------------------
1.  void edytujUcznia(LUczen* u, Uczen ucz); //zmienia dane i umozliwia usuwanie i
    edycje ocen

//-----------------------------------------------------------------------------

//-----------------------------------------------------------------------------
1.  void dziennikDodajOcene(LUczen*& uczen); //
    nadzoruje dodawanie ocen

//-----------------------------------------------------------------------------
```

Funkcja umożliwiająca dodanie oceny wcześniej wybranemu uczniowi za pomocą funkcji:

```
//-----------------------------------------------------------------------------
1. void dodajOcene(LUczen*& uczen, Ocena ocena); // dodaje ocene podanemu uczniowi

//-----------------------------------------------------------------------------

//-----------------------------------------------------------------------------
1. void usunUcznia(LUczen*& glowa, LUczen* u);  //usuwanie
   wyłacznie po indeksie by nie bylo pomyłki

//-----------------------------------------------------------------------------
```

Funkcja usuwająca wybranego wcześniej ucznia, pod warunkiem że nie jest on jedynym w bazie danych.

```
//-----------------------------------------------------------------------------
1. void wyswietlDziennik(LUczen* glowa, int a); // wyswietla
   cala baze danych

//-----------------------------------------------------------------------------
```

Funkcja wyświetlająca wszystkich uczniów i ich oceny.

```
//-----------------------------------------------------------------------------
1. void usunDziennik(LUczen*& glowa); // usuwa wszystkie dane

//-----------------------------------------------------------------------------
```

Funkcja usuwająca wszystkie dane.

```
//-----------------------------------------------------------------------------
1. void zapiszDziennik(LUczen* glowa, int ilosc); // zapisuje
   dziennik

//-----------------------------------------------------------------------------
```

Funkcja zapisująca dane do pliku binarnego.

```
//-----------------------------------------------------------------------------
1. void wczytajDziennik(LUczen*& glowa, int& ilosc); //
   wczytuje dziennik

//-----------------------------------------------------------------------------
```

Funkcja wczytująca dane z pliku binarnego.

Testowanie

Do programu wprowadzane były różne dane. Pozwoliło to na wykrycie paru błędów W działaniu funkcji dodającej uczniów w kolejności alfabetycznej.

Kluczowe dla programu było odpowiednie obsługiwanie wskaźników, co sprawiało problemy ze względu na nie intuicyjność. Na szczęście po wielu testach udało się usunąć wszystkie błędy.

Największe problemy sprawiły funkcje zapisu i odczytu z pliku binarnego. Udało się wyeliminować błędy z funkcji zapisu. Niestety w funkcji odpowiedzialnej za odczyt z pliku binarnego zdarza się wyrzucanie wyjątków.

Wnioski

Stworzenie programu, który obsługuje prostą bazę danych wymaga wiedzy o dynamicznych strukturach danych. Ważne jest również umiejętne obsługiwanie plików aby gdzieś te dane zapisać. Stworzenie tego programu zwiększyło moją znajomość tych aspektów. Praca nad projektem o takiej złożoności wymaga sporo czasu. Dużo uwagi należy zwrócić na testowanie aplikacji w celu wyeliminowania błędów.