

# **Dziennik Elektroniczny**

Giza Artur

# 1.Charakterystyka programu i instrukcja obsługi.

Dziennik Elektroniczny to program mający za zadanie umożliwić w szybki i wygodny sposób zarządzanie grupą uczniów. Pozwala nam utworzyć pełną listę osób, zawierającą numer przydzielany każdemu z osobna, dane osobowe (Nazwisko,Imię) oraz uzyskane oceny.Program ponadto posiada kilka przydatnych funkcji jak:

- Szybkie wyszukiwanie ucznia - w sytuacjach nagłej potrzeby dostępu do wyników i informacji ucznia możemy wykorzystać tę opcję. Po wprowadzeniu imienia i nazwiska w wymaganym polu zwrócone zostają wszelkie informacje dotyczące danej osoby.
- Losowanie ucznia do odpowiedzi- opcja ta pozwala nam na szybkie wylosowanie jednego ucznia spośród wszystkich osób znajdujących się na liście.
- Dodaj ocenę- W przypadku uzyskania coraz to nowszych ocen w trakcie prowadzenia zajęć jesteśmy w stanie je dopisywać (jedną lub kilka za jednym razem) do innych wartości w rubryce Oceny.Nie muszą być one wyłącznie liczbami całkowitymi.
- Usuń ucznia z dziennika- umożliwia skreślenie osoby z listy wraz z jej wynikami i numerem. Następnie program na nowo przydziela numery dla kolejnych osób.
- Zapis listę w pliku- funkcja ta pozwala nam zapisać utworzoną listę w pliku tekstowym w formie jaką możemy zobaczyć wywołując funkcję "Wyświetl dziennik".

## Obsługa programu:

```
*****DZIENNIK*****
[1].Skonfiguruj dziennik
[2].Wyświetl dziennik
[3].Wprowadz ocene
[4].Szybkie wyszukiwanie ucznia
[5].Losuj ucznia do odpowiedzi
[6].Usun ucznia z dziennika
[7].Zapisz liste w pliku
[8].Wyjście.
Wybor:
```

Po uruchomieniu programu ukazuje się nam menu dziennika. Aby rozpocząć konfigurację naszej listy należy w polu "Wybor:" wprowadzić interesującą nas funkcję numerowaną od [1] do [8]. Po wybraniu funkcji podążamy za poleceniami wydawanymi przez program. Po zakończeniu funkcji zostaną wyświetlone komunikaty informujące nas o zachodzących procesach.

## 2.Specyfikacja techniczna.

Program został napisany w języku c++ w środowisku Microsoft Visual Studio Community 2017 wersji 4.7.03062. Do kompilacji zaleca się używać najnowszej wersji Visual Studio. Kod projektu został podzielony na trzy pliki:

- **main.cpp**

Znajduje się w niej menu, do którego stworzenia została wykorzystana pętla while odpowiedzialna za zakończenie programu i powrót do menu po każdej wykonanej akcji. Składnia **switch ... case** umożliwia nam wybór dowolnej akcji z menu, które z kolei zawierają w sobie wywołania poszczególnych funkcji odpowiedzialnych za przebieg żądanego procesu.

- **funkcje.cpp**

Zawiera definicje funkcji :

- drukuj

Za pomocą **setw()** zarządzamy strumieniem, tak aby nadać odpowiednią strukturę wyświetlanego dziennika. Po wyświetleniu numeru, nazwiska i imienia, pętla for zwraca kolejne oceny ucznia.

```
void uczen::drukuj()
{
    cout << setw(2)<< numer_ucznia << ". " << setw(21) << nazwisko << setw(21) << imie << " " << " | ";
    for(int i=0;i < oceny_ucznia.size();i++) // Wyświetlanie ocen
    {
        cout << oceny_ucznia[i] << " | ";
    }
    cout << endl;
    cout << endl;
}
```

- dodaj

Pozwala dodać nowe obiekty do listy jednokierunkowej.

```

void Lista::dodaj()
{
    Uczeń Nowy = new uczen;
    cout << "Imie:"; //
    string imie, nazwisko; //Wprowadzanie danych nowego ucznia.
    cin >> imie; //
    cout << "Nazwisko:"; //
    cin >> nazwisko; //
    Nowy->imie = imie; //
    Nowy->nazwisko = nazwisko; //
    cout << "Ilosc ocen ucznia :"; //
    int ilosc_ocen;
    cin >> ilosc_ocen;
    for (int i = 0; i < ilosc_ocen; i++) // Wprowadzanie dowolnej ilosci ocen dla nowego ucznia.
    {
        double oceny;
        cout << "Ocena nr." << i+1<<":";
        cin >> oceny;
        Nowy->oceny_ucznia.push_back(oceny);
    }
    //sprawdzamy czy cos juz jest na liscie;
    if (pierwszy == nullptr)
    {
        pierwszy = Nowy;
        Nowy->kolejny = nullptr;
    }
    else
    {
        Uczeń Biezacy = pierwszy;
        while (Biezacy->kolejny)
            Biezacy = Biezacy->kolejny;
        Biezacy->kolejny = Nowy;
    }

    Nowy->kolejny = nullptr;
}
getchar();
getchar();
system("cls");
}

```

- drukujListe

Odpowiada za wyświetlanie dziennika. Dla każdej osoby listy jednokierunkowej zostaje wywołana funkcja dodaj.

```

void Lista::drukujListe()
{
    Uczeń biezacy = pierwszy;
    if(biezacy != nullptr)
    {
        cout << "Lp." << "      Nazwisko"<< "      Imie      " << "      Oceny" << endl;
        cout << "-----" << endl;
        while (biezacy != nullptr)
        {
            biezacy->drukuj();
            biezacy = biezacy->kolejny;
        }
    }
    else
        cout << "**** Lista jest pusta ****" << endl;
}

```

- szukajj

Przeszukuje listę jednokierunkową w poszukiwaniu ucznia o wskazanym imieniu i nazwisku, aby wyświetlić jego dane.

```
void Lista::szukaj()
{
    Uczeń biezacy = new uczen;
    biezacy = pierwszy;
    string nazwisko_szukane, imie_szukane;
    cout << "Nazwisko:"; cin >> nazwisko_szukane; //Wprowadzenie danych szukanej osoby.
    cout << "Imie:"; cin >> imie_szukane; //
    system("cls");
    if (biezacy->nazwisko == nazwisko_szukane && biezacy->imie == imie_szukane)
    {
        cout << "Lp." << "      Nazwisko" << "      Imie      " << "      Oceny" << endl;
        cout << "-----" << endl;
        biezacy->drukuj();
    }
    else
    {
        while (biezacy->kolejny != nullptr)
        {
            biezacy = biezacy->kolejny;
            if (biezacy->nazwisko == nazwisko_szukane && biezacy->imie == imie_szukane)
            {
                cout << "Lp." << "      Nazwisko" << "      Imie      " << "      Oceny" << endl;
                cout << "-----" << endl;
                biezacy->drukuj();
                break;
            }
        }
    }
    getchar();
    getchar();
    system("cls");
}
```

- Losuj ucznia do odpowiedzi

```
void Lista::losuj_ucznia_do_odpowiedzi()
{
    Uczeń biezacy = pierwszy;
    int liczba_osob_w_liscie = 0;
    while (biezacy != nullptr) //sprawdzanie ilosci osob w liscie
    {
        biezacy = biezacy->kolejny;

        liczba_osob_w_liscie++;
    }

    srand(time(NULL));
    int losowa_liczba = (rand() % liczba_osob_w_liscie) + 1; //losowanie liczby dowolnej z zakresu od 1 do liczby uczniow wprowadzonych do dziennika
    biezacy = pierwszy; //
    if (losowa_liczba != 1)
    {
        biezacy = biezacy->kolejny; //
        for (int i = 2; i < losowa_liczba; i++) //ustawienie wskaznika na osobe o pozycji w liscie wylosowanej w powyzzszym kroku.
        {
            biezacy = biezacy->kolejny; //
        }
    }
    cout << "Do odpowiedzi zostal wywolany uczen:" << biezacy->nazwisko << ' ' << biezacy->imie << endl; //Wywołanie osoby wylosowanej do odpowiedzi
}
```

- usun

Usuwa ucznia o określonym numerze.

```

void Lista::usun(int nr)
{
    Uczeń wczesniejszy, biezacy;
    int ind = 1;
    if (pierwszy == nullptr) return;
    if (nr == 1) {
        biezacy = pierwszy->kolejny;
        delete pierwszy;
        pierwszy = biezacy;
    }
    else {
        biezacy = pierwszy;
        wczesniejszy = pierwszy;
        while (biezacy->kolejny != nullptr && ind < nr) {
            wczesniejszy = biezacy;
            biezacy = biezacy->kolejny;
            ind++;
        }
        if (biezacy != nullptr) {
            wczesniejszy->kolejny = biezacy->kolejny;
            delete biezacy;
        }
    }
}

```

- przydzielanie numeru ucznia

```

void Lista::przydzielanie_numeru_ucznia()
{
    int numer = 1; //ustawienie początkowego numeru
    Uczeń biezacy = pierwszy;
    while (biezacy != nullptr) //Petla w ktorej kazdemu nastepnemu uczniowi przydziela sie numer o 1 wiekszy od poprzedniego.
    {
        biezacy->numer_ucznia = numer;
        numer++;
        biezacy = biezacy->kolejny;
    }
}

```

- dodaj ocene

```

    {
        Uczeń biezacy = new uczen;
        biezacy = pierwszy;
        string nazwisko_szukane, imie_szukane;           //Szukanie określonego ucznia w liście
        cout << "Nazwisko:"; cin >> nazwisko_szukane;    //
        cout << "Imie:"; cin >> imie_szukane;           //
        system("cls");
        if (biezacy->nazwisko == nazwisko_szukane && biezacy->imie == imie_szukane)
        {
            int ilosc_ocen;
            cout << "Ile ocen chcesz dopisać:";         //Wprowadzanie kolejnych ocen
            cin >> ilosc_ocen;                             //
            for (int i = 0; i < ilosc_ocen; i++)           //
            {                                               //
                double oceny;                             //
                cout << "Ocena nr." <<(i + 1) << ":";    //
                cin >> oceny;                             //

                biezacy->oceny_ucznia.push_back(oceny);    // Dopisanie kolejnych ocen ucznia do klasy wektor.
            }
        }
        else
        {
            while (biezacy->kolejny != nullptr)
            {
                biezacy = biezacy->kolejny;
                if (biezacy->nazwisko == nazwisko_szukane && biezacy->imie == imie_szukane)
                {
                    int ilosc_ocen;
                    cout << "Ile ocen chcesz dopisać:";
                    cin >> ilosc_ocen;
                    for (int i = 0; i < ilosc_ocen; i++)
                    {
                        double oceny;
                        cout << "Ocena nr." << + (i+1) << ":";
                        cin >> oceny;
                        biezacy->oceny_ucznia.push_back(oceny);
                    }
                    break;
                }
            }
        }
    }
}

```

- zapisz do pliku



```

void Lista::zapisz_do_pliku()
{
    fstream plik;
    plik.open("Lista_uczniow.txt", ios::out); //Utworzenie pliku "Lista_uczniow.txt" i otworzenie go z trybem otwarcia umożliwiającym zapis danych do pliku.
    Uczeń biezacy = pierwszy;
    plik << "Lp." << "      Nazwisko" << "      Imie      " << "      Oceny" << endl;
    plik << "-----" << endl;

    while (biezacy != nullptr) //Wpisywanie kolejnych osob z listy wraz z jej numerem i ocenami, w wyniku czego otrzymujemy pełna liste zapisana w pliku tekstowym.
    {
        plik<< setw(2) << biezacy->numer_ucznia << ". " << setw(21) << biezacy->nazwisko << setw(21) << biezacy->imie << "      " << " | ";
        for (int i = 0; i < biezacy->oceny_ucznia.size(); i++)
        {
            plik << biezacy->oceny_ucznia[i] << " | ";
        }
        plik << endl;
        biezacy = biezacy->kolejny;
    }
    plik.close(); // Zamkniecie pliku po operacji zapisu listy.
    cout << "*** Lista uczniow zostala zapisana ***";
    getchar();
    getchar();
    system("cls");
}

```

- **naglowek.h**

Zawiera deklaracje struktur:

```

struct uczen {
    string imie;
    string nazwisko;
    uczen * kolejny;
    vector<double> oceny_ucznia;
    int numer_ucznia;
    void drukuj();
};

using Uczeń = uczen * ;

struct Lista {
    Uczeń pierwszy;
    void dodaj();
    void drukujListe();
    void szukaj();
    void losuj_ucznia_do_odpowiedzi();
    void usun(int);
    void przydzielanie_numeru_ucznia();
    void zapisz_do_pliku();
    void dodaj_ocene();
};

```

**Wykorzystywane biblioteki:**

```

#include <iostream>
#include <string>
#include <windows.h>
#include <cstdlib>

```



```
#include <ctime>
#include <vector>
#include <fstream>
#include <stdio.h>
#include <iomanip>
```