

Sekcja

5

ORGANIZER

BARDZO PROFESJONALNY KALENDARZ

Dokumentacja

Skład sekcji:

Alan Biegun

Dawid Ryl

Konrad Sładkowski

Jan Snopek

Piotr Zuber

13-06-2016

Spis treści

1.	Wprowadzenie	3
a)	Wstęp	3
b)	Cel i założenia projektu	3
c)	Podział prac	3
2.	Dokumentacja użytkownika	4
a)	O programie	4
b)	Wymagania techniczne	4
c)	Instalacja	4
d)	Korzystanie z programu	5
e)	Opis funkcji programu	7
3.	Dokumentacja techniczna	11
a)	Podział programu	11
b)	Działanie programu	12
c)	Algorytmy i ważniejsze metody	14
d)	Zastosowane narzędzia i biblioteki	16
4.	Podsumowanie	17
a)	Harmonogram prac	17
b)	Wnioski	17
c)	Spis literatury	17

1. Wprowadzenie

a) Wstęp

Program „Organizer” jest naszym pierwszym projektem realizowanym w ramach toku nauczania na naszym kierunku. Jest to aplikacja konsolowa z przyjaznym dla użytkownika interfejsem graficznym. Pozwala ona na zarządzanie w prosty i wygodny sposób czasem użytkownika poprzez organizację jego wydarzeń. Program pozwala na wykorzystywanie go przez wielu użytkowników.

b) Cel i założenia projektu

Celem tego projektu jest stworzenie za pomocą programowania obiektowego w języku C++ programu użytkowego, który pozwala na komfortowe zarządzanie i planowanie wydarzeń użytkownika. Przyjęta struktura organizera pozwala na intuicyjną obsługę programu, a interfejs graficzny wpasowuje się w ogólnie przyjęty schemat wyglądu organizatorów.

Zrealizowane założenia:

- obiektowa realizacja projektu
- wykorzystanie systemu kontroli wersji GIT-SCM
- obsługa wielu użytkowników
- wyświetlanie kalendarza, bieżącej daty oraz godziny
- dodawanie, usuwanie oraz edycja wydarzeń
- zaznaczenie ważniejszych dni oraz wydarzeń
- możliwość sortowania wydarzeń wg. wybranych przez użytkownika kryteriów
- obsługa cykliczności wydarzeń

c) Podział prac

W trakcie planowania budowy aplikacji, została ona podzielona na pięć klas i każda z osób jest odpowiedzialna za jedną. W trakcie prowadzenia prac wiele razy zdarzało się jednak, iż wzajemnie uzupełnialiśmy nasze klasy pomagając sobie radą oraz konstruktywną krytyką. Poniżej przedstawiono funkcje każdego z członków zespołu oraz zrealizowane przez nich zadania.

Alan Biegun – główny programista, projektant klasy MainMenu, zaimplementował algorytm wiecznego kalendarza.

Dawid Ryl – główny tester, projektant klasy User, implementacja funkcji hashujących, poprawa drobnych błędów w programie oraz przygotowywanie prezentacji.

Konrad Śladkowski – projektant klasy File_Guardian, opracowanie struktur katalogów oraz formatów przechowywania danych użytkowników.

Jan Snopek – projektant klasy Callendar, projektant aktualnego interfejsu graficznego aplikacji, odpowiedzialny za poprawne wywołania metod z pozostałych klas.

Piotr Zuber – główny planista, projektant klasy Event, odpowiedzialny za stworzenie i implementacji funkcji sortujących, prezentowanie programu.

2. Dokumentacja użytkownika

a) O programie

Organizer jest to aplikacja konsolowa z przyjaznym dla użytkownika interfejsem graficznym, która pozwala poza takimi podstawowymi funkcjami jak wyświetlanie aktualnej godziny i daty, oraz przeglądanie kalendarza na tworzenie wydarzeń i ich podgląd według określonych kryteriów (np. według Nazwy, Daty czy Priorytetu).

Dodatkowo w programie zaimplementowana została obsługa wielu użytkowników, dzięki czemu informacje zapisane w programie chronione są przed dostępem niepowołanych osób. Zapominasz o swojej rocznicy ślubu? Planujesz wyjazd i potrzebujesz optymalnie zaplanować swój czas? To nie problem dla naszego Organizera!

Dodatkowo nasz Organizer otrzymał od ekspertów najwyższe noty w kategorii prezentacja projektu. Nie czekaj! Zmień teraz swój notatnik na profesjonalny Organizer stworzony przez sekcję piątą.

b) Wymagania techniczne

- Windows XP lub wyżej.
- 256 MB ramu.
- Jednordzeniowy procesor 1.2 Ghz.

c) Instalacja

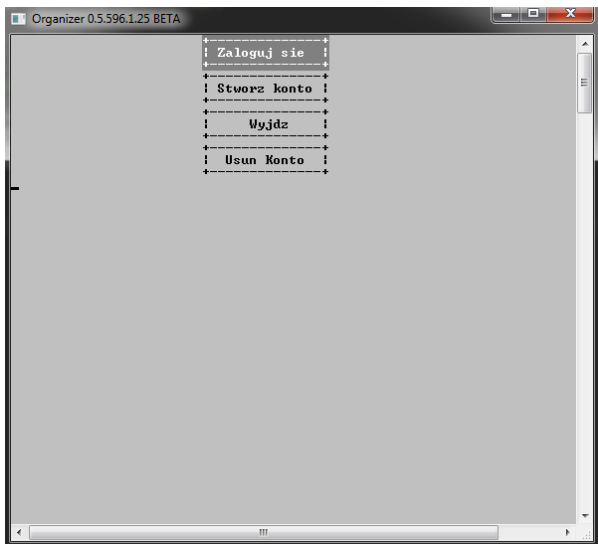
Aby instalacja programu nastąpiła pomyślnie wymagane jest przeprowadzenie jej w kilku określonych krokach:

- Utworzenie katalogu w którym będzie przechowywany program.
- Przeniesienie pliku exe Organizera do tego folderu.
- Przeniesienie folderu w którym znajduje się plik exe Organizera do lokacji w której chcemy przechowywać nasz program.

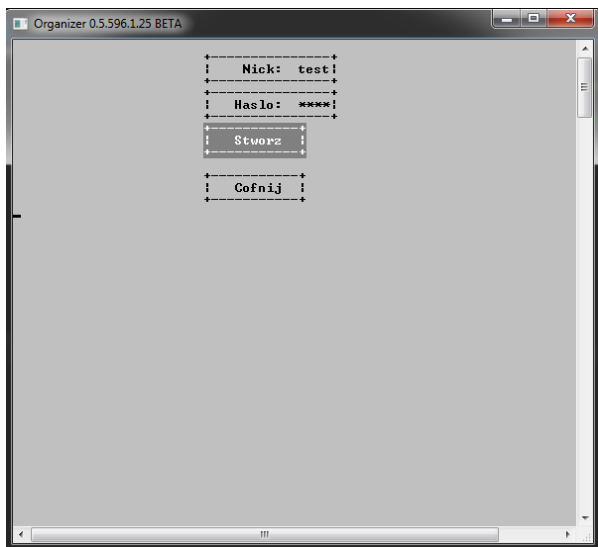
d) Korzystanie z programu

Pierwsze Kroki

Po włączeniu programu pokaże się nam menu główne, po którym poruszamy się strzałkami góra i dół. Aby dostać się do części praktycznej programu niezbędne jest utworzenie użytkownika. Aby to nastąpiło wybieramy opcję Stworz konto.



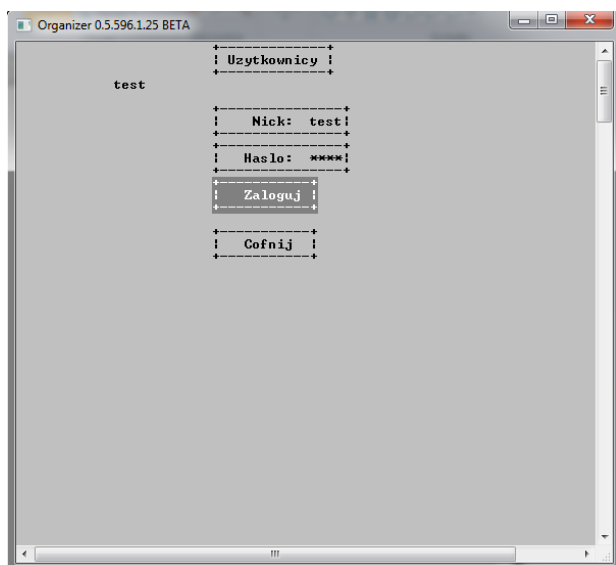
Po wybraniu odpowiedniej opcji pokaże się nam takie oto okno:



Aby utworzenie nowego użytkownika nastąpiło pomyślnie należy przestrzegać kilku podstawowych zasad:

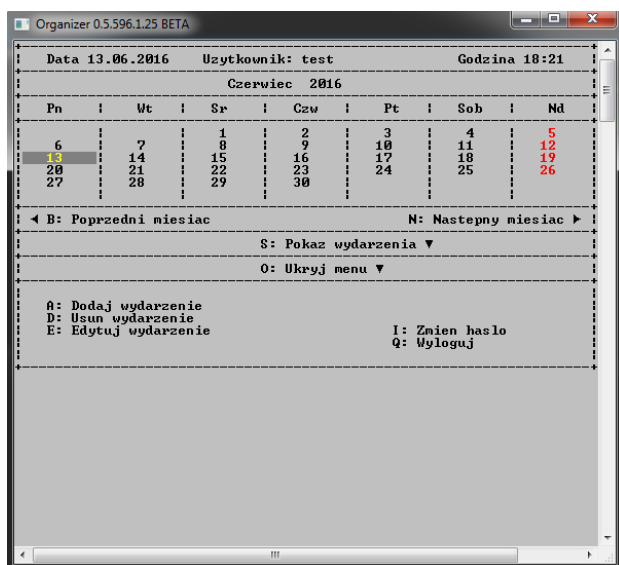
- Nick i hasło muszą mieć 4 - 15 znaków.
- Nick i hasło nie mogą zawierać Polskich znaków, oraz innych znaków specjalnych (np. [],',./).
- Nie może być stworzonych dwóch użytkowników o takim samym nicku.

Po stworzeniu nowego użytkownika wybieramy w menu głównym opcję Zaloguj się, a następnie po uzupełnieniu wymaganych pól wybieramy opcję Zaloguj.



Główne okno programu

Gdy proces logowania został przeprowadzony pomyślnie pojawi się nam główne okno programu które wygląda następująco:



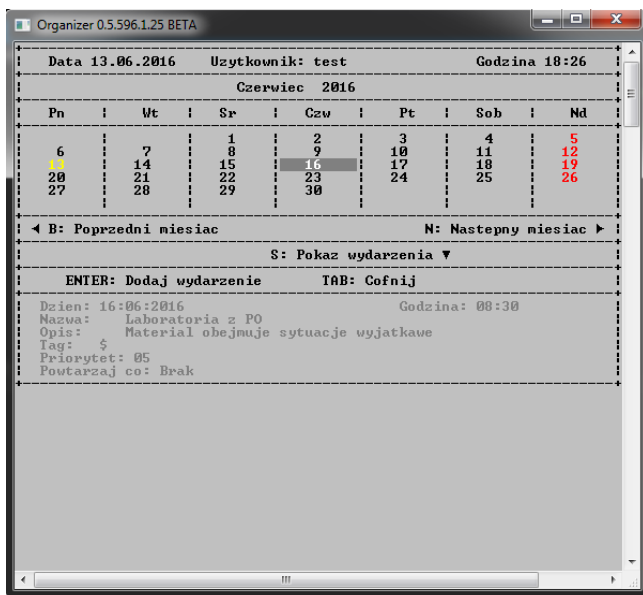
Jak widać na powyższej ilustracji okno programu jest podzielone na fragmenty które odpowiadają za wyświetlanie różnego rodzaju informacji:

- W pierwszej linii wyświetlona jest aktualna data, godzina oraz aktualnie zalogowany użytkownik.
- Kalendarz po którym możemy się poruszać przyciskami funkcyjnymi (B i N) jak i strzałkami. Dzień na kalendarzu wyświetlony na żółto jest dniem dzisiejszym.
- Menu Funkcyjne w którym znajduje się między innymi obsługa wydarzeń (dodawanie, edycja i ich usuwanie), oraz zmiana hasła użytkownika.

e) Opis funkcji programu

Tworzenie wydarzeń

Aby utworzyć nowe wydarzenie wybieramy w głównym oknie programu opcję Dodaj wydarzenie (Klikamy na klawiaturze przycisk A)

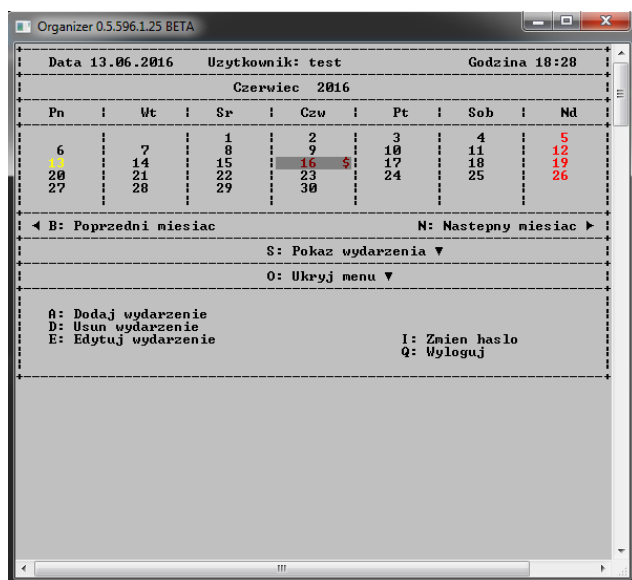


Na wydarzenie w programie składają się:

- Dzień - wybierany jest na kalendarzu. Ciemne pole to dzień który mamy aktualnie wybrany.
- Godzina - Strzałkami prawo - lewo wybieramy godzinę, natomiast strzałkami góra - dół wybieramy minuty.
- Nazwa - może mieć maksymalnie 20 znaków. Polskie znaki oraz znaki specjalne nie są dozwolone.
- Opis - może mieć maksymalnie 60 znaków. Polskie znaki oraz znaki specjalne nie są dozwolone.
- Tag - pojedynczy znak reprezentujący wydarzenie na kalendarzu. Wybieramy go z pośród określonej puli.
- Priorytet - Określa wagę wydarzenia. Występuje on w skali od 0 do 10.
- Powtarzaj co - Określa cykliczność wydarzenia. Dostępne są 3 rodzaje cykliczności: Brak, Co tydzień, Co miesiąc.

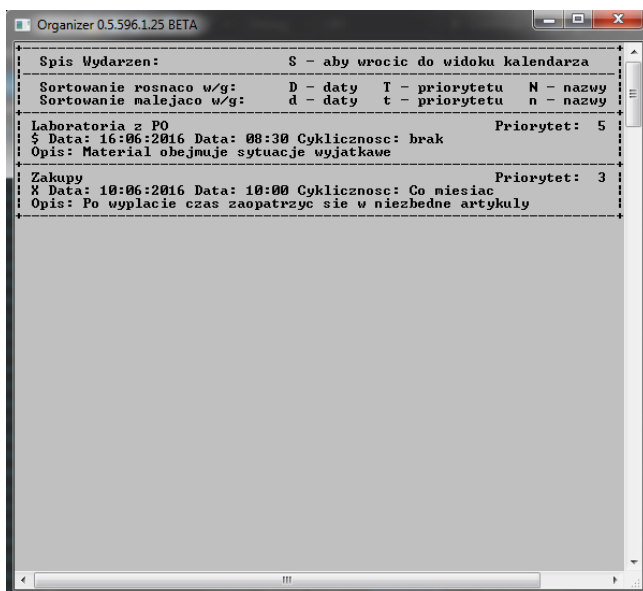
Po poprawnym wypełnieniu pola akcję zatwierdzamy klawiszem enter. Jeżeli chcemy wprowadzić poprawki klawisz tab cofa nas do ostatnio wypełnionego pola.

Po stworzeniu wydarzenia zostanie zaznaczone ono na kalendarzu. Znak znajdujący się przy numerze dnia to tag wydarzenia.



Wyświetlanie listy wydarzeń

Aby przeglądać listę wydarzeń wybieramy w głównym oknie programu opcję Pokaz wydarzenia (Klikamy na klawiaturze przycisk S)



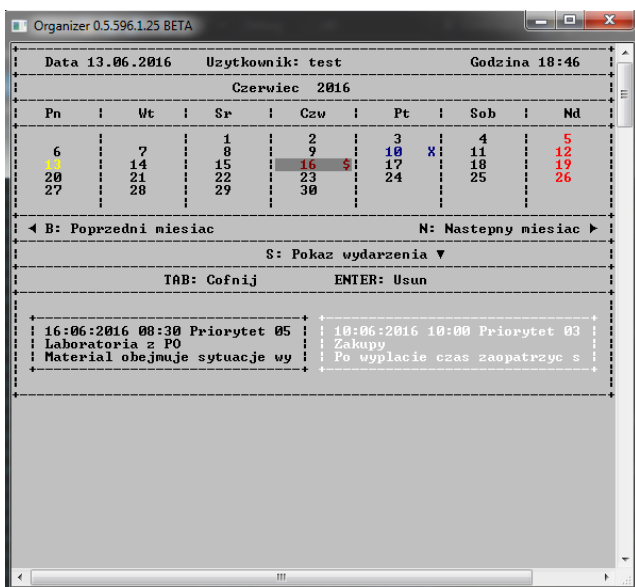
Wydarzenia znajdujące się na liście można posortować w celu lepszej organizacji. W programie występują 3 kryteria sortowania:

- Sortowanie według daty - (rosnąco D, malejąco d)
- Sortowanie według priorytetu - (rosnąco T, malejąco t)
- Sortowanie według nazwy - (rosnąco N, malejąco n)

Aby powrócić do głównego okna programu wybieramy przycisk S.

Usuwanie wydarzeń

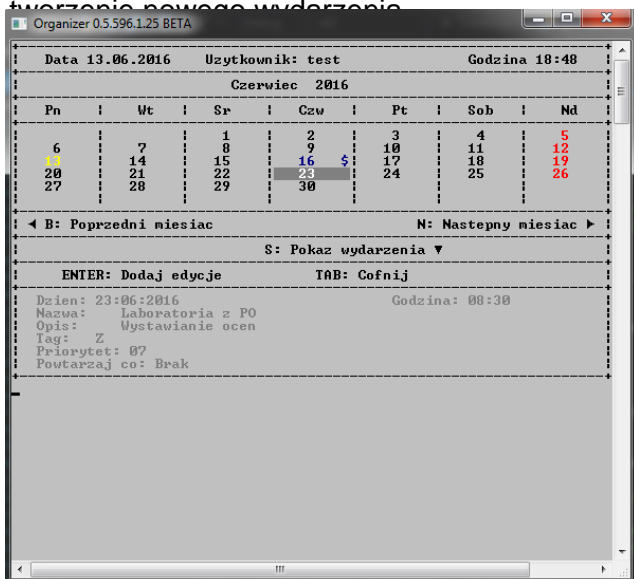
Aby usunąć wydarzenie wybieramy w głównym oknie programu opcję Usun wydarzenie (Klikamy na klawiaturze przycisk D)



Aby wybrać wydarzenie które chcemy usunąć używamy strzałek góra dół, a następnie zatwierdzamy klawiszem enter. Jeżeli nie jesteśmy zdecydowani na usunięciu wydarzenia możemy wyjść z menu usuwania wydarzeń klawiszem tab.

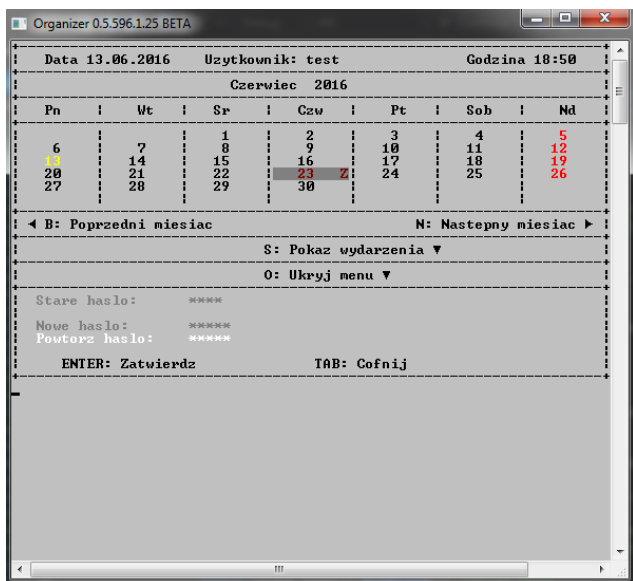
Edycja wydarzeń

Gdy chcemy edytować jakieś wydarzenie wybieramy z menu głównego opcję Edytuj wydarzenie (Za którą odpowiada klawisz E). Wydarzenie które chcemy edytować wybieramy za pomocą strzałek góra dół, a następnie dokonujemy jego edycji na takiej samej zasadzie jak tworzenie nowego wydarzenia.



Zmiana hasła użytkownika

Aby zmienić hasło użytkownika wybieramy w głównym oknie programu opcję Zmien hasło (Klikamy na klawiaturze przycisk I)

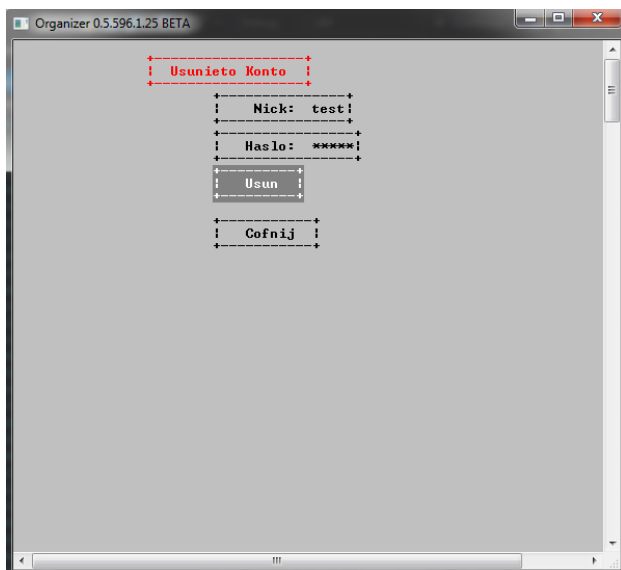


Aby zmienić hasło wymagane jest poprawne wpisanie starego hasła, a następnie po zatwierdzeniu klawiszem enter dwukrotne wpisanie nowego hasła. Nowe hasło musi spełniać takie same wymagania jak hasło które zamierzamy zmienić:

- Długość pomiędzy 4 - 15 znaków.
- Nie może zawierać Polskich znaków, oraz innych znaków specjalnych (np. [],',./).

Usuwanie użytkownika

Usuwanie użytkownika jest możliwe tylko z poziomu menu głównego. Aby to mogło nastąpić wymagana jest znajomość nicku użytkownika, oraz jego hasła.



3. Dokumentacja techniczna

a) Podział programu

Program składa się z pięciu uzupełniających się klas, plik z definicjami oraz plik zawierający funkcje sortujące.

Wyróżniamy klasy:

- File_Guardian – klasa odpowiadająca za obsługę plików oraz wszelkie wykonywane na nich operacje ,między innymi:
 - Tworzenie i usuwanie użytkowników.
 - Obsługa zapisu oraz edycji nazwy i za haszowanego hasła użytkownika.
 - Obsługa zapisu, edycji i usuwania wydarzeń.
 - Tworzenie i usuwanie struktury katalogów.
 - Sprawdzenie poprawności (zgodności) danych wprowadzanych z klawiatury z danymi przechowywanymi w plikach. * poprawić.
- MainMenu – odpowiada za wyświetlanie menu w oknie po uruchomieniu programu. Klasa jest rozszerzana przez dwie klasy: MenuModel i MenuView. Zastosowano podział odpowiednio na obsługę przetwarzania danych oraz samo wyświetlanie. Udostępnia następujące możliwości:
 - Stworzenie indywidualnego konta z unikalną nazwą.
 - Zalogowanie się do konta wraz z przejściem do widoku kalendarza.
 - Wyświetlanie nazw kont użytkowników.
 - Usuwanie konta.
 - Wyjście z programu.
- User – klasa pośrednicząca pomiędzy kalendarzem a klasą pliku(File_Guardian). Jej ważniejsze funkcje to:
 - Przechowuje tablicę wydarzeń oraz nazwę użytkownika.

- Tworzy i edytuje wydarzenia poprzez modyfikacje poszczególnych elementów tablicy wydarzeń.
 - Sprawdza czy danego dnia występuje wydarzenie.
 - Aktualizacja tablicy wydarzeń poprzez metodę w klasie File_Guardian.
- Callendar – obsługuje główny widok zaraz po zalogowaniu się użytkownika. Rozszerza ją Callendar_Model odpowiedzialna za przetwarzania danych. Główne zadania klasy:
 - Wyświetlanie kalendarza oraz wydarzeń.
 - Umożliwia edycje hasła z poziomu kalendarza.
 - Zapewnia możliwość dodawania, usuwania i edycji wydarzeń na poziomie kalendarza.
 - Wyświetlanie listy wydarzeń oraz ich sortowanie.
 - Wyświetlanie aktualnej daty i godziny.
 - Wyróżnienie dni z zaplanowanym wydarzeniem.
 - Event – Obiekt tej klasy przechowuje informację na temat pojedynczego wydarzenia. Składa się z:
 - Kilku prywatnych pól określających: nazwę, opis, datę, cykliczność i jej rodzaj, tag oraz priorytet.
 - Dostęp do pól zapewniają odpowiednie metody get i set .

b) Działanie programu

Po uruchomieniu programu następuje ustalenie domyślnych kolorów, rozmiaru okna i jego nazwy. Realizowane jest to za pomocą odpowiednich funkcji:

```
SetConsoleTitle(TEXT("Organizer")); // Nazwa okna
SET_COLOR(DEFAULT_TEXT_COLOR); // Domyślne kolory konsoli

SMALL_RECT windowSize = { 0, 0, CONSOLE_SIZE_X, CONSOLE_SIZE_Y }; //
Rozmiar oknaSetConsoleWindowInfo(GetStdHandle(STD_OUTPUT_HANDLE),
TRUE, &windowSize);
```

Następnie tworzone są instancje obiektów File_Guardian, MainMenu, User oraz Callendar, po czym program przechodzi do wykonywania głównej pętli. Wewnątrz niej wykonywana jest główna metoda wyświetlająca menu, zwracająca nazwę użytkownika w razie poprawnego zalogowania lub pusty string. W drugim przypadku program zakańcza swoje działanie, usuwa utworzone wcześniej instancje obiektów. W razie poprawnego zalogowania, do klasy użytkownika przekazywana jest nazwa użytkownika i wcześniej utworzony obiekt File_Guardian. Na podstawie tak wypełnionej klasy User jest inicjalizowana klasa Calendar po czym uruchamiany jest właściwy widok programu.

W razie wyjścia z kalendarza pętla uruchamia się na nowo aż do momentu wyjścia poprzez opcję „Wyjdź” dostępną z menu głównego lub poprzez „X” w prawym górnym rogu okna.

Po uruchomieniu metody Menu(), program tworzy zmienne buforowe przechowujące login, hasło i przechodzi do wykonywania pętli, w której na początku następuje czyszczenie konsoli za pomocą zdefiniowanego słowa „CLEAR”. Następnie wywoływana jest metoda PrintMenu(), która odpowiedzialna jest za wyrysowanie i odpowiednie wyświetlenie menu głównego. Executelnput() realizuje poruszanie się po programie za pomocą strzałek, zwraca true jeśli wybrana jest opcja wyjścia lub użytkownik poprawnie się zalogował.

```
do {
    CLEAR; // Czyszczenie ekranu
    PrintMenu(sLoginBuffer, sPasswordBuffer, bWritingStatus, fFile-
>NumberOfUsers(), fFile->getUsers()); // Rysowanie menu na podstawie
danych

    bExit = Executelnput(Getlnput(), iMenuState, iMenuSelectedOption,
sErrorMessage); // Pobranie przycisku i wykonanie akcji
} while (!bExit);
```

Po zainicjowaniu instancji obiektu Callendar jest wywoływana metoda RunCalendar() powodująca przejście do pętli głównej Callendar. Na początku pętli znajduje się metoda odświeżająca „MOVETO”. Przenosi ona kursor na początek konsoli.

```
while (!_ExitProgram) {

    MOVETO; //// Wraca na górę konsoli i nadpisuje wszystko.

    FindFirstLastDays();
```

```

et),    MainView(getDate("%d.%m.%G"),getDate("%H:%M"),(iMonthOffset),(iYearOffs
        getDate("%d"));

        ExecuteEvent(ReadUserInput());

}

```

FindFirstLastDays() jest metodą odpowiedzialną za pobranie dzisiejszej daty z systemu, określenie liczby dni miesiąca, obliczenie przesunięcia miesiąca podczas przewijania kalendarza.

MainView() wyświetla na podstawie podanych parametrów dany miesiąc w kalendarzu.

ExecuteEvent() odbiera dane wejściowe z klawiatury (poruszanie się po menu, obsługa wszystkich przycisków).

c) Algorytmy i ważniejsze metody

```

int User::DayOfWeek(int iDay, int iMonth, int iYear) {
    static int t[] = { 0, 3, 2, 5, 0, 3, 5, 1, 4, 6, 2, 4 };
    iYear -= iMonth < 3;
    return ((iYear + iYear / 4 - iYear / 100 + iYear / 400 + t[iMonth - 1] + iDay) % 7);
}

```

Algorytm określający jakim dniem tygodnia jest dana data.

```

string HashString(string Text) {
    return to_string(hash<string>()(Text));
}

```

Metoda haszująca wprowadzany tekst

```

string getDate(char* Mod) {
    time_t Time;
    struct tm * Date;
    char Result[80];

    time(&Time);
    Date = localtime(&Time);

    strftime(Result, 80, Mod, Date);
    return Result;
}

```

Metoda pobierająca czas systemowy.

```

int findLastDay(int iMonth, int iYear) {
    if (iMonth == 2) {
        if ((iYear % 400 == 0 || iYear % 100 != 0) && (iYear % 4 == 0))
            return 29;
        else return 28;
    }
    else if (iMonth == 4 || iMonth == 6 || iMonth == 9 || iMonth == 11) {
        return 30;
    }
    else {
        return 31;
    }
}

```

Algorytm określający liczbę dni w miesiącu.

```

bool User::isEvent(int iDayNumber, int iMonthOffset, int iYearOffset, int *counter) {
    SetEvent();

    for (int i = 0; i < fFile->getNoOfElements(); i++) {
        if (iYearOffset == eEventList[i].getYear()) {
            if (iMonthOffset == eEventList[i].getMonth()) {
                if (iDayNumber == eEventList[i].getDay()) {
                    *counter = i; return true; // Danego dnia jest event
                }
            }
        }

        if (DayOfWeek(iDayNumber, iMonthOffset, iYearOffset) ==
            (DayOfWeek(eEventList[i].getDay(), eEventList[i].getMonth(), eEventList[i].getYear())))
        {
            if (eEventList[i].getPeriod() == co_tydzien) {
                *counter = i; return true;
            }
        }
        else if (eEventList[i].getPeriod() == co_miesiac) {
            if (iDayNumber == eEventList[i].getDay()) {
                *counter = i; return true;
            }
        }
    }
    return false;
}

```

Algorytm sprawdzający czy jest w danym dniu wydarzenie, następnie sprawdzający czy jest cykliczny.

```
int findFirstDay(int iDayOfMonth, int iDayN) { // kiedy jest pierwszy dzien miesiaca
    while (iDayOfMonth - 7 > 1)
        iDayOfMonth -= 7;
    while (iDayOfMonth - 1 != 0) {
        iDayOfMonth--;
        iDayN--;
    }

    return iDayN;
}
```

Algorytm sprawdzający kiedy przypada pierwszy dzień miesiąca.

d) Zastosowane narzędzia i biblioteki

- Biblioteki:
 - <string>
 - <iostream>
 - <Windows.h>
 - <conio.h>
 - <string.h>
 - <fstream>
- Oprogramowanie:
 - Microsoft Visual Studio Enterprise 2015
 - Git/GitHub

4. Podsumowanie

a) Harmonogram prac

- 3.03 – spotkanie organizacyjne w ramach wyboru tematu projektu
- 7.03 – konferencja internetowa, tworzenie pierwszej prezentacji oraz założeń
- 14.04 – spotkanie w ramach podziału pracy i zaprojektowania budowy programu, metodyki działań oraz stworzenie schematów blokowych obrazujących działanie programu
- 18.04 – rozpoczęcie pracy w systemie kontroli wersji GIT-SCM oraz przygotowań do kolejnej prezentacji
- 6.05 – połączenie wszystkich klas w działający program, rozpoczęcie prac nad rozbudową programu
- 30.05 – przygotowania do finalnej prezentacji
- 3.06 – spotkanie dotyczące planowania niezbędnych modyfikacji do programu, ponowne rozplanowanie zadań
- 6.06 – rozpoczęcie fazy testów oraz poprawia błędów
- 10.06 – rozpoczęcie prac nad dokumentacją
- 12.06 – zakończenie prac nad programem
- 13.06 – zakończenie prac nad dokumentacją

b) Wnioski

Realizacja tego projektu uświadomiła nam, jak wiele możliwości otwiera podejście obiektowe w projektowaniu. Szczególny nacisk położono na staranne rozplanowanie prac oraz komunikacje w zespole, co mimo wielu trudności z czasem zaowocowało.

Projekt przyniósł wraz z wieloma wyzwaniami tyle samo doświadczeń. Co prawda nie tylko tych przyjemnych aczkolwiek na pewno przydatnych w przyszłej pracy zawodowej. Przejrzysty podział obowiązków i ról, ustalanie terminów „deadline” czy wzajemna pomoc nauczyły nas, że komunikacja w zespole jest równie ważna, jak umiejętności programistyczne.

c) Spis literatury

- Radosław Sokół, „Microsoft Visual Studio 2012. Programowanie w C i C++”
- Jerzy Grębosz, „Symfonia C++ Standard”
- Serwis internetowy <http://cpp0x.pl/>