

Uniwersytet Przyrodniczo-Humanistyczny

Kalendarz (Przypominacz)

Dokumentacja projektu zaliczeniowego

Autor: Kornel Domeradski

Kierunek: Informatyka

Semestr: Semestr II

Minsk Mazowiecki, 2018r.

Cel Projektu

Projekt implementacji systemu wspomagającego zarządzanie wydarzeniami powiązanymi z mundialem. 4 Pliki przechowujące dane: wydarzenia (Events), konta (accounts), konta z przywilejami administracyjnymi (administrators) oraz ustawienia (settings).

Założenia Projektu

Aplikacja ma możliwość zalogowania różnych użytkowników z których jedynie administratorzy mogą dodawać/odejmować wydarzenia. Aplikacja przechowuje dane o wydarzeniach w pliku tekstowym. Możliwość usuwania konkretnych wpisów. Możliwość edytowania konkretnych wpisów. Aplikacja ładuje domyślne ustawienia okien, czcionki itp. z pliku.

Funkcjonalności

- Logowanie za pomocą hasła oraz loginu
- Wybór czy użytkownik chce się zalogować jako użytkownik normalny lub administrator w przypadku posiadania owych uprawnień
- Przeglądanie kalendarza w celu uzyskania informacji nt. Wydarzeń
- Dodawanie nowych wydarzeń z użyciem kalendarza
- Edytowanie wcześniej dodanych wydarzeń z użyciem kalendarza
- Usuwanie wcześniej dodanych wydarzeń z użyciem kalendarza
- Zmiana ustawień wizualnych programu
- Możliwość zmiany trybu z trybu administracyjnego na tryb dostępny dla każdego użytkownika

Narzędzia

Program został napisany w języku Java a dokładniej na Java 10 z użyciem JDK 10. Systemem operacyjnym na którym pisana była aplikacja jest dystrybucja GNU/Linux nazywana Arch Linux. Do edycji kodu używane były IDE IntelliJ IDEA oraz edytor tekstu atom. W projekcie został także użyty Git, jako narzędzie ułatwiające zarządzanie kodem. Użyty został także GitHub w celu przechowywania kodu źródłowego.

Struktura Aplikacji

Ze względu na rodzaj aplikacji tj. CRUD (ang. skrót od „Create, Read, Update, Delete”), czyli aplikacji skupiającej się na przetwarzaniu informacji o ustandaryzowanej strukturze uznałem, że wskazane jest wyizolowanie modelu danych i mechanizmów do jego obsługi do oddzielnych modułów. Jako, że mechanizm bazy danych często jest wykorzystywany w innych projektach wskazane jest również pisanie kodu w sposób umożliwiający wykorzystanie go ponownie w innych aplikacjach.

Schemat Ogólny

(SEKCJA DO ZROBIENIA)

Przebieg tworzenia aplikacji

Zaprojektowanie struktury baz danych

Aplikacja wykorzystuje dwie tabele (które znajdują się w plikach tekstowych) oraz przechowuje informacje w jeszcze dwóch innych plikach tekstowych.

Wspomniane tabele:

- events – zawiera informacje o wszelkich wydarzeniach wprowadzonych przez administratora
- accounts – zawiera informacje o wszystkich loginach i hasłach

Wspomniane pliki tekstowe które nie przechowują danych w stylu typowym dla tabel:

- administrators – zawiera listę użytkowników posiadających uprawnienia administratorskie
- settings – zawiera informacje na temat wielkości czcionki oraz na temat wielkości okien

events

Plik tekstowy ten zawiera informacje o wszystkich wydarzeniach utworzonych przez administratora. Ma on klucz główny id (który jest unikatowy dla każdego wpisu), który ułatwia rozróżnianie poszczególnych wpisów programowi. Plik ten może zawierać komentarze po znaku '#' a puste linijki są pomijane. W celu dokładnej informacji jak dodawać manualnie wydarzenia bez użycia programu należy zobaczyć komentarze w pliku „events”.

Struktura „tabeli” pliku tekstowego:

Nazwa Pola	Typ Pola	Opis
id	int – L. Całkowita	Klucz główny – ułatwia rozróżnienie poszczególnych wpisów programowi
name	String – Ciąg znaków	Przechowuje nazwę wydarzenia
description	String – Ciąg znaków	Przechowuje opis wydarzenia
teamASquad	String – Ciąg znaków	Przechowuje imiona i nazwiska zawodników oddzielone przecinkiem
teamBSquad	String – Ciąg znaków	Przechowuje imiona i nazwiska zawodników oddzielone przecinkiem
stadium	String – Ciąg znaków	Przechowuje informację nt. obiektu
day	int – L. Całkowita	Przechowuje informację nt. dnia
month	int – L. Całkowita	Przechowuje informację nt. miesiąca
year	int – L. Całkowita	Przechowuje informację nt. roku
startTimeHour	int – L. Całkowita	Przechowuje informację nt. godziny rozpoczęcia
startTimeMinute	int – L. Całkowita	Przechowuje informację nt. minuty rozpoczęcia
EndTimeHour	int – L. Całkowita	Przechowuje informację nt. godziny zakończenia
EndTimeMinute	Int – L. Całkowita	Przechowuje informację nt. minuty zakończenia

accounts

Plik tekstowy przechowujący informacje na temat loginów oraz haseł użytkowników. Tabela jest minimalistyczna i nie przypomina zbytnio tabeli, które występują w bazach danych relacyjnych ze względu na to, że elementy takie jak klucz główny nie są tutaj potrzebne. Struktura „tabeli” wygląda tak:

Nazwa Pola	Typ Pola	Opis
login	String – Ciąg znaków	Przechowuje login
hasło	String – Ciąg znaków	Przechowuje hasło

administrators

Plik tekstowy przechowujący informacje nt. loginów które mają dostęp do uprawnień administracyjnych. Loginy są wypisane w kolejnych liniach.

settings

Plik tekstowy przechowujący informacje nt. Ustawień aplikacji. Tak jak w przypadku events możliwe jest dodawanie komentarzy oraz puste linie są pomijane.

Projektowanie mechanizmu obsługi bazy

Edytowanie bazy danych jest czymś co stosuje się w wielu projektach oraz tak naprawdę w wielu klasach dlatego uznałem za najlepsze rozwiązanie utworzenie nowej klasy w której znajdować się będą statyczne metody dzięki którym będę mógł dokonać zmiany z wielu poziomów programu. Klasa ta została nazwana **EventQueryProcessor** . Oto metody które je obsługują:

```
public static void setEventAmount()
```

Ustawia ilość zapisanych eventów w statycznej zmiennej

```
public static void addEvent(CalendarEvent event)
```

Dodaje nowe wydarzenie na końcu pliku „events”

```
public static void editEvent(CalendarEvent event)
```

Edytuje wydarzenie oraz zapisuje je w tym samym miejscu w pliku „events” po wykonaniu edycji

```
public static void deleteEvent(CalendarEvent event)
```

Usuwa wybrane wydarzenie z pliku „events”.

```
public static ArrayList<CalendarEvent> getEventByDate(int day, int month, int year)
```

Zwraca listę wydarzeń na podstawie kryterium daty czyli wszystkie wydarzenia które wydarzają się danego dnia.

Schemat blokowy klas:

(Ze względu na rozmiar zostały załączone w folderze z dokumentacją pod nazwą „diagram_1.png” i „diagram_2.png”)

Kod źródłowy:

(Ze względu na rozmiar został załączony w projekcie folderu „src”)