

## Dokumentacja kalendarza.

### Działanie:

Kalendarz jest aplikacją wykorzystującą GUI Shoes. Po uruchomieniu wyświetla on na samej górze dzisiejszą datę oraz 4 przyciski - odpowiadają one odpowiednio za dodanie notatki („Add note”), usunięcie jej („Delete note”), wyświetlenie wszystkich nadchodzących wydarzeń („All upcoming notes”) oraz ustawień wyświetlania („Settings”).

Dodanie notatki odbywa się przez ustalenie jej treści w polu tekstowym, określenie jej daty, godziny, minuty, interwału co jaki czas się powtarza i określeniu typu notatki (Notatka bez daty, z datą lub cykliczna) - gdy jakieś pole nie jest wykorzystywane w notatce, to może zostać ustawione na dowolną wartość.

Usunięcie notatki odbywa się przez zaznaczenie typu notatki, którą chcemy usunąć, treści notatki z listy oraz wybranie dokładnej daty oraz godziny jej wykonania.

Poniżej są wyświetlane notatki, dla których data wykonania nie została określona („To-do notes”) i notatki z określoną datą lub cykliczne (powtarzające się co jakiś interwał czasu).

W katalogu z programem tworzą się trzy pliki o nazwach „calendar\_example.ca”, „config.ca” oraz „notes\_example.ca” odpowiadające za przechowywanie, odpowiednio, notatek z datami, ustawień konfiguracyjnych programu oraz notatek bez daty.

Oprócz tego kalendarz raz na 5 sekund wczytuje pliki kalendarza (prócz pliku konfiguracyjnego - żeby zobaczyć zmiany trzeba uruchomić program jeszcze raz) i, jeśli nastąpiła jakaś zmiana, zamienia napisy oraz sprawdza, czy właśnie nie mija jakieś wydarzenie i jeśli mija, to wyświetla odpowiednie powiadomienie.

### Kod:

Program jest podzielony na trzy moduły - moduł *calendar.rb* odpowiada za obsługę GUI (kontroler), moduł *calendarnotes.rb* jest odpowiedzialny model kalendarza, a moduł *calendarshow.rb* jest odpowiedzialny za wyświetlanie obiektów modelu (widok).

W pliku *calendar.rb* mamy następujące klasy, które odpowiadają za kontroler:

**Config** - obiekt ustawień konfiguracyjnych kalendarza. Obiekt klasy Config ma pola **@note\_view**, **@default\_view\_mode**, **@color** i **@font\_color** w domyśle przechowujące nazwę funkcji lub kolor, która jest wykorzystywana odpowiednio do wyświetlania „To-do notes”, notatek z datą, koloru tła kalendarza oraz koloru czcionki.

**Program** - obiekt klasy program otwiera program i obsługuje gui.

W pliku *calendarnotes.rb* mamy następujące klasy, które odpowiadają za model kalendarza:

**BlankNote** - notatka, która nie ma określonej daty. Ma pole **@content**, które przechowuje treść notatki.

**DateNote** - podklasa klasy **BlankNote**, która oprócz treści notatki ma jeszcze datę i godzinę jej wykonania, przechowywane pod polami **@date** oraz **@hour**. (Uwaga: @hour jest przechowywany jako liczba, której część ułamkowa jest dokładnie wartością minut - przykładowo gdy notatka jest ustawiona na 14:32 to wartość @hour jest równa 14.32 - rozpatrywanie tego „liniowo” (tj. @hour = 14.5(3)) sprawia problem z dokładnością ułamków i możliwymi rozbieżnościami na poziomie +/- 1 minuty)

**CyclicNote** - podklasa klasy **DateNote**, która oprócz wyżej wymienionych ma jeszcze interwał co jaki powtarza się notatka przechowywany pod polem **@repeat**. Metoda switch

„przełącza” notatkę na odpowiednią datę (czyli przesuwając o interwał @repeat), jeśli termin jej wykonania już minął.

**Noteline** - obiekt tej klasy lista notatek **BlankNote** (nie posiadających daty i godziny). Jedynym polem jest pole **@events** będące listą notatek. Metody **add\_event** oraz **delete\_event** odpowiednio dodają i usuwają notatki z **@events**. (Uwaga: argumentem **delete\_event** jest napis generowany przez notatkę, a nie pole **@content**).

**Timeline** - jest to struktura przechowująca wszystkie notatki o określonej dacie i godzinie. Pole **timeline** jest Hashem, który za klucze przyjmuje daty. Wartościami natomiast są kolejne Hashe, które za klucze przyjmują godziny, a za wartości przyjmują listę notatek na daną porę. Metoda **add\_event** odpowiada za dodanie notatki (obiektu klasy **DateNote** lub **CyclicNote**), a metoda **delete\_event** odpowiada za usunięcie notatki względem jej komunikatu (jej metody **to\_s**).

**CollectOld** - jest to klasa definiująca metodę „odśmieciania” obiektów klasy **Timeline** - metoda **collect\_old** usuwa notatki, których termin wykonania minął. Przechodzi ona po całym hashu **timeline** obiektu **Timeline** i usuwa go, jeśli jego data już minęła.

**SwitchCyclic** - definiuje metody „przenoszenia” notatek cyklicznych na kolejne daty. Metoda **switch\_cyclic\_from\_date** bierze datę oraz obiekt **Timeline**, przechodzi wszystkie obiekty o określonej dacie i jeśli są notatkami cyklicznymi to przesuwają je o interwał czasu @repeat określony w takim obiekcie. Metoda **switch\_old\_cyclic** wykonuje powyższą metodę na wszystkich datach w **Timeline**, pod warunkiem, że ich data wykonania już minęła.

W pliku *calendarshow.rb* mamy następujące klasy, które odpowiadają za widok:

**ShowNoteline** - jest to klasa definiująca metody wyświetlania obiektów klasy **Noteline**. Metoda **to\_s** przyjmuje w domyśle obiekt klasy **Noteline** i wypisuje wszystkie notatki łącząc je myślnikiem. Metoda **list\_all** tworzy listę wszystkich notatek w postaci napisów. Metoda **firstn\_to\_s** tworzy napis z min(n, ilość notatek) pierwszych notatek połączonych myślnikiem. Metoda **first10\_to\_s** jest szczególnym przypadkiem metody **firstn\_to\_s** dla n = 10.

**ShowTimeline** - jest to klasa definiująca wyświetlanie obiektów klasy **Timeline**. Metody **hour\_to\_s** oraz **dateTime\_to\_s** odpowiadają za wyświetlanie odpowiednio godzin oraz obiektów klasy **dateTime** (pomijają „rozwiniecie godzinowe” obiektów klasy **dateTime**). Metoda **list\_all** tworzy listę treści wszystkich notatek przechowywanych w obiekcie klasy **Timeline**. Metoda **day\_to\_s** bierze dzień (jako obiekt typu Hash) i zwraca odpowiednio sformatowany napis z godzinami i wydarzeniami o danych godzinach. Metoda **timeline\_to\_s** wypisuje wszystkie notatki chronologicznie, z datami i godzinami. Metoda **upcoming\_ndays\_to\_s** konkatenuje napis wykonany przez metodę **day\_to\_s** wywołaną na nadchodzących n dniach. Metody **upcoming\_month\_to\_s** i **upcoming\_week\_to\_s** to szczególne przypadki powyższej metody, tak samo jak metoda **today\_to\_s** jest szczególnym przypadkiem metody **day\_to\_s**.