

Kalendarz

Projekt zaliczeniowy

Bazując na wiadomościach zdobytych na wykładzie oraz ćwiczeniach napisz program będący prostym kalendarzem. Kalendarz powinien mieć następujące możliwości:

- dodawanie nowych wydarzeń (nazwa, dzień, godzina),
- wyświetlanie wszystkich wydarzeń w formie tekstowej,
- wyświetlanie wydarzeń w formacie iCalendar (<https://pl.wikipedia.org/wiki/ICalendar>), pliki zapisane w tym formacie można zaimportować do innej aplikacji kalendarza (np. Google Calendar).

Przykład działania programu (na zielono zaznaczono tekst wprowadzony przez użytkownika):

```
$ python3 main.py
1. New event
2. List calendar
3. Export calendar to iCalendar
4. Exit
Select menu item (1-4): 1
Title: Programowanie obiektowe w języku PYTHON - Wykład
Date (DD.MM.YYYY): 28.03.2020
Time (HH:MM): 08:00
1. New event
2. List calendar
3. Export calendar to iCalendar
4. Exit
Select menu item (1-4): 1
Title: Programowanie obiektowe w języku PYTHON - Cwiczenia
Date (DD.MM.YYYY): 28.03.2020
Time (HH:MM): 12:45
1. New event
2. List calendar
3. Export calendar to iCalendar
4. Exit
Select menu item (1-4): 2
Title: Programowanie obiektowe w języku PYTHON - Wykład
Date: 28.03.2020, 08:00
Title: Programowanie obiektowe w języku PYTHON - Cwiczenia
```

Notatki:

```
Date: 28.03.2020, 12:45
1. New event
2. List calendar
3. Export calendar to iCalendar
4. Exit
Select menu item (1-4): 3
BEGIN:VCALENDAR
VERSION:2.0
BEGIN:VTIMEZONE
TZID:Europe/Warsaw
X-LIC-LOCATION:Europe/Warsaw
END:VTIMEZONE
BEGIN:VEVENT
DTSTART:20200328T080000
DTEND:20200328T080000
SUMMARY:Programowanie obiektowe w języku PYTHON - Wykład
END:VEVENT
BEGIN:VEVENT
DTSTART:20200328T124500
DTEND:20200328T124500
SUMMARY:Programowanie obiektowe w języku PYTHON - Ćwiczenia
END:VEVENT
END:VCALENDAR
1. New event
2. List calendar
3. Export calendar to iCalendar
4. Exit
Select menu item (1-4): 1
Title: Wykład $2
Invalid input
1. New event
2. List calendar
3. Export calendar to iCalendar
4. Exit
Select menu item (1-4): 4
```

Architektura rozwiązania

Menu programu powinno zostać wykonane za pomocą wzorca Polecenie (podobnie jak miało to miejsce na ćwiczeniach). Klasa MenuCommand powinna być klasą bazową dla poleceń menu i zawierać:

- metodę o nazwie `description` zwracającą nazwę polecenia, nazwa ta będzie umieszczona w menu,

Notatki:

- metodę o nazwie `execute` wykonującą określoną czynność.

W menu powinny znajdować się cztery opcje zrealizowane za pomocą podklas klasy `MenuCommand`:

- opcja nr 1: dodanie nowego wydarzenia do kalendarza
- opcja nr 2: wypisanie wydarzeń w formie tekstowej
- opcja nr 3: wypisanie wydarzeń w formacie iCalendar
- opcja nr 4: zakończenie działania programu

Wskazówka: Narzędzie sprawdzające będzie weryfikowało kolejność opcji w menu. Jeśli chcesz rozszerzyć menu o własne opcje, użyj numerów od 5 w górę.

Wypisywanie wydarzeń powinno być zrealizowane według wzorca Strategia. Powinna istnieć ogólna funkcja wypisująca zawartość kalendarza, ale format wypisywanych danych powinien być określony za pomocą określonego obiektu strategii.

Program powinien być zrealizowany przy pomocy dołączonego szablonu projektu, w którym znajduje się:

- plik `menu.py` z klasami `Menu` oraz `MenuCommand`, `ExitCommand`
- plik `main.py` z funkcją `main`, która wyświetala menu na ekranie,
- plik `calendar.py` z funkcją `list_calendar` wraz z klasą `ListingStrategy`, funkcja ta jest odpowiedzialna za wywoływanie odpowiednich metod klasy `ListingStrategy`, metody te powinny wypisywać na ekranie dane w odpowiednim formacie,
- dodatkowe wskazówki w poszczególnych plikach.

Dodawanie nowego wydarzenia do kalendarza

Po wybraniu w menu opcji nr 1 program powinien zapytać, w następującej kolejności o:

- nazwę wydarzenia,

Notatki:

- datę,
- godzinę.

Nazwa wydarzenia to jedno-liniowy napis złożony ze znaków:

- od A do Z
- od a do z
- od 0 do 9
- - (minus)
- , (przecinek)
- . (kropka)
- spacji

Wczytywana data od użytkownika powinna być w formacie: DD.MM.RRRR

Wczytywany czas powinien być w formacie: GG:MM

Jeżeli w którymś z kroków użytkownik podał niepoprawne dane to program powinien natychmiast wyświetlić napis “Invalid input” i powrócić do menu głównego nie wprowadzając żadnych zmian w kalendarzu.

Wypisywanie danych w formie tekstowej

Dla każdego z przechowywanych wydarzeń program powinien wypisać dwie linie postaci:

```
Title: <tutaj tytuł wydarzenia>
Date: DD.MM.RRRR, GG:MM
```

Przykład:

```
Title: Programowanie obiektowe w języku PYTHON - Cwiczenia
Date: 28.03.2020, 12:45
```

Notatki:

.....

.....

Wypisywanie danych w formacie iCalendar

Wypisywane dane powinny rozpoczynać się sekwencją informującą o formacie danych i strefie czasowej:

```
BEGIN:VCALENDAR
VERSION:2.0
BEGIN:VTIMEZONE
TZID:Europe/Warsaw
X-LIC-LOCATION:Europe/Warsaw
END:VTIMEZONE
```

Następnie dla każdego z przechowywanych w kalendarzu wydarzeń program powinien wypisać informacje o wydarzeniu w formacie (zwróć uwagę, że po dwukropku nie ma spacji):

```
BEGIN:VEVENT
DTSTART:<data i godzina wydarzenia>
DTEND:<data i godzina wydarzenia>
SUMMARY:<tytuł wydarzenia>
END:VEVENT
```

Znaczenie poszczególnych pól:

- DTSTART - data i czas rozpoczęcia wydarzenia,
- DTEND - data i czas zakończenia wydarzenia (w naszym przypadku umieszczamy czas rozpoczęcia),
- SUMMARY - tytuł wydarzenia.

Format czasu DTSTART i DTEND jest następującej postaci: RRRRMMDDTGGMM00, gdzie:

- RRRR – rok,
- MM – miesiąc,

Notatki:

- DD – dzień,
- T - litera T oddzielająca datę od godziny,
- GG – godzina,
- MM – minuta,
- 00 - dwa zera (według standardu jest to miejsce na sekundy, ale na nasze potrzeby przyjmujemy, że są to zawsze dwa zera)

Przykład:

```
BEGIN:VEVENT
DTSTART:20200328T124500
DTEND:20200328T124500
SUMMARY:Programowanie obiektowe w języku PYTHON - Cwiczenia
END:VEVENT
```

Po wypisaniu wszystkich wydarzeń program powinien wyświetlić:

```
END:VCALENDAR
```

Przykład całości:

```
BEGIN:VCALENDAR
VERSION:2.0
BEGIN:VTIMEZONE
TZID:Europe/Warsaw
X-LIC-LOCATION:Europe/Warsaw
END:VTIMEZONE
BEGIN:VEVENT
DTSTART:20200328T124500
DTEND:20200328T124500
SUMMARY:Programowanie obiektowe w języku PYTHON - Cwiczenia
END:VEVENT
END:VCALENDAR
```

Notatki:

Testowanie

Rozważ testowanie swojego rozwiązania testami jednostkowymi z użyciem słowa kluczowego `assert`, podczas pisania projektu. Pomoże to zapobiec pojawiającym się nowym błędom lub błędom regresji¹, gdy dopiszesz kolejną funkcjonalność.

Punktacja

Obowiązująca punktacja (w nawiasach podano możliwe do zdobycia liczby punktów):

- warunek wstępny do uzyskania punktów (sprawdzone przez prowadzącego ćwiczenia): program został zaimplementowany z użyciem wzorców projektowych polecenie i strategia,
- 0pkt albo 5pkt - program działa poprawnie dla podanego w treści zadania przykładu,
- od 0pkt do 13pkt - program działa poprawnie dla poprawnie wprowadzonych danych,
- od 0pkt do 12pkt - program działa poprawnie, nawet gdy użytkownik wprowadza błędne dane.

Uwaga:

- *Nie korzystaj z bibliotek do obsługi formatu iCalendar, obsługę tego formatu napisz samodzielnie.*
- *Dozwolone jest korzystanie z bibliotek dostarczanych z domyślną instalacją Pythona.*

Rozwiązania niespełniające powyższych wymagań otrzymają 0pkt!

¹ [https://pl.wikipedia.org/wiki/Regresja_\(informatyka\)](https://pl.wikipedia.org/wiki/Regresja_(informatyka))

Notatki:

.....

.....