# DOKUMENTACJA PROJEKTU W RAMACH PRZEDMIOTU PROGRAMOWANIE W JĘZYKU PYTHON.

**Autor:**

Filip Maciborski

**Tytuł projektu:**

Vocabulary tester

Tematy uwzględnione w projekcie:

1. Git.
2. Czytelny kod w Pythonie – PEP8.
3. Zaawansowane struktury danych – moduł Pandas.
4. Obsługa dat i czasu – moduł DateTime.
5. Programowanie funkcyjne.
6. Programowanie obiektowe.

## Git i GitHub

Projekt od pewnego momentu przechowywany był na GitHubie i także za jego pośrednictwem został przekazany wykładowcy. W trakcie pracy nad projektem, choć dopiero od pewnego momentu stosowany był Git.

Git został zainstalowany oraz odpowiednio skonfigurowany. Został także połączony poprzez klucz SSH z GitHubem.

Obraz zawierający tekst, elektronika, zrzut ekranu, oprogramowanie

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.Historia commitów


## Czytelny kod w Pythonie – PEP8.

Nad projektem pracowałem z wykorzystaniem Visual Studio Code. Jako dodatek zainstalowałem *Flake8* podkreślający wszystkie możliwe błędy jakie występują w kodzie niezgodnym z *PEP8* np.:

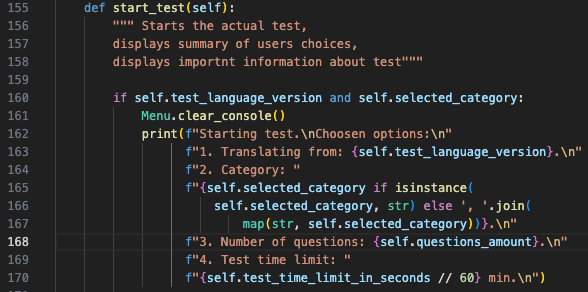
Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Zastosowałem się do wszystkich wskazań dodatku i dostosowałem kod do *PEP8*.

**Przykłady** (na podstawie klasy *NewTest*)**:**

* + Dzielenie długich stringów na linie i stosowanie odpowiednich wcięć:



* + ‘Zawijanie’ długich wywołań funkcji:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

* + Komentarze opisujące zadania/działanie metod jako multiline docstring:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

* + Prawidłowe uporządkowanie argumentów funkcji w formie listy:

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

* + Uporządkowane importy:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

* + Prawidłowe nazewnictwo klas i metod:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

## Zaawansowane struktury danych – moduł Pandas.

W projekcie został wykorzystany moduł *Pandas*. W największym zakresie wykorzystana została *DataFrame* w oparciu, o który program funkcjonuje:

* + Praca z plikiem Excel i arkuszami pliku w których została utworzona baza danych programu (klasa *AppController*):

Pobranie danych z pliku:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Zapis danych do pliku:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

* + Wykorzystanie *DataFrame* do zarządzania przepływem danych, podczas tworzenia pytań i zbierania odpowiedzi użytkownika, dodawania nowych danych do bazy danych.

Łączenie danych:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Tworzenie nowej kategorii słów:  
Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Operacje na kolumnach w celu pozyskania wyników testu (klasa *NewTest*):

- **map()** – zmiana wartości bazując w serii na podstawie podanej innej wartości

- **apply()** – stosowanie funkcji do wierszy lub kolumn

- **astype()** - zmiana typu danych w series

- **mean()** – średnia procentowa kolumny

- **sum()** – obliczenie sumy kolumny

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

## Obsługa dat i czasu – moduł DateTime.

Moduł *DateTime* został w programie wykorzystany do pobrania daty i godziny przeprowadzenia testu, a także zmierzenia czasu jego trwania. W oparciu o limit czasu ustawiany przez użytkownika w minutach, obliczany jest limit czasu na sekundy. Stworzony dekorator mierzy czas trwania testu. Na podstawie porównania wartości określane jest to czy użytkownik zmieścił się w limicie czasowym.

**Przykłady** (klasa NewTest)**:**

* + Pobranie daty i czasu podczas tworzenia instancji klasy *NewTest* – wybranie   
    z menu głównego opcji -> *‘Start New Test’*

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

* + Ustalenie limitu czasu dla testu – bez wykorzystania *DateTime,* ale ważne   
    w późniejszym użyciu (klasa *TimeManager*)

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

* + Pomiar czasu trwania testu (wrapper w *TimeManager*):

**Obraz zawierający tekst, zrzut ekranu, Czcionka

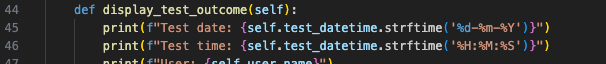
Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.**

* + Pomiar czasu uruchomiony dla funkcji, która wyświetla pytania i pobiera odpowiedzi użytkownika, czyli w momencie, gdy użytkownik przechodzi do rozwiązywania testu (klasa *NewTest*, metoda *def submit\_answer()* wywołana   
    w metodzie def *start\_test()*):

**Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.**

* + Wyświetlanie poszczególnych elementów *DateTime*, osobno daty   
    i czasu w odpowiednim formacie (klasa *ResultManager* metoda   
    *def display\_test\_outcome()*):

****

## Programowanie funkcyjne.

## Programowanie obiektowe.