

Podstawy Informatyki | Geoinformatyka

Celem ćwiczenia jest zapoznanie się z GitHubem, obsługa plików oraz wizualizacją danych. Ćwiczenie będzie polegało na napisaniu kalkulatora BMI oraz wizualizacji wyników. Należy uwzględnić informację zwrotną o tym czy BMI jest w normie oraz wpisać do pliku tekstowego poza samym wynikiem informacje umieszczone w kolumnach z Rysunku 1. Wizualizacje mają uwzględniać kolory analogiczne do tych poniżej:

Klasyfikacja masy ciała osób dorosłych na podstawie BMI^[7]

Kategoria	BMI (kg/m ²)	Waga ciała	Ryzyko chorób towarzyszących otyłości ^[4]
wyglodzenie	< 16,0	niedowaga	minimalne, ale zwiększony poziom wystąpienia innych problemów zdrowotnych
wychudzenie	16,0–16,99		
niedowaga	17,0–18,49		
pożądana masa ciała	18,5–24,99	optimum	minimalne
nadwaga	25,0–29,99	nadwaga	średnie
otyłość I stopnia	30,0–34,99	otyłość	wysokie
otyłość II stopnia (duża)	35,0–39,99		bardzo wysokie
otyłość III stopnia (chorobliwa)	≥ 40,0		ekstremalny poziom ryzyka

Rysunek 1 Tabela BMI, źródło: https://pl.wikipedia.org/wiki/Wska%C5%BAnik_masy_cia%C5%82a

UWAGA jest dowolność wyboru języka programowania – C lub Python

1. W grupach należy stworzyć repozytorium na githubie (UWAGA punkty 1-4 wykonuje jedna osoba, udostępniając ekran reszcie grupy). Można zrobić to z poziomu aplikacji desktopowej, strony lub poleceniem w terminalu (git init). Żeby stworzyć repozytorium lokalnie należy wcześniej wybrać lub stworzyć odpowiedni katalog.
2. Po utworzeniu repozytorium należy dodać do niego plik README.txt w którym będzie znajdował się opis programu zawierający informacje o tym czym jest BMI. Opis ten należy dodać do repozytorium (z poziomu CMD należy wpisać git add, git commit).
3. Należy sprawdzić status komendą git status.
4. Następnie należy wprowadzić zmiany w pliku README.txt wpisując imiona i nazwiska osób w grupie i ponownie wykonać sekwencję git add, git commit, git status.
5. Tak stworzone repozytorium należy udostępnić innym wraz z wszystkimi dostępnymi.

6. Każda z osób dodaje w pliku README.txt dodatkowy paragraf. Uwaga! Należy robić to we wcześniej utworzonej przez siebie gałęzi o nazwie „moja_galaz_imie_nazwisko” (przeczytaj <https://docs.github.com/en/pullrequests/collaborating-with-pull-requests/proposing-changes-to-your-work-withpull-requests/about-branches>) . Każdy wybiera sobie tematykę: BMI - jakie są jego zakresy, kto stworzył ten wskaźnik i po co oraz jakie są ryzyka chorobowe związane z nieprawidłowymi wartościami.
7. Następnie należy powtórzyć sekwencję git add, git commit. Następnie należy wystawić komuś z zespołu *pull request* (<https://docs.github.com/en/pullrequests/collaborating-with-pull-requests/proposing-changes-to-your-work-withpull-requests/creating-a-pull-request>) po to by sprawdził zmiany. Wszystkie wersje pliku należy następnie połączyć i zapisać w głównej gałęzi.
8. Należy stworzyć kilka gałęzi – main (końcowa – być może już jest), develop (do rozwijania funkcjonalności), podgałęzie develop: bmi-text, bmi-input, bmicharts.
9. W każdej gałęzi będą rozwijane inne funkcjonalności programu do liczenia BMI.
 - a) BMI – text – program liczący BMI po zaimportowaniu tabeli z pomiarami z pliku tekstowego/xlsx. Należy stworzyć plik z minimum 20 wpisami.
 - b) BMI – input - program liczący BMI po wpisaniu wartości przez użytkownika w konsoli. Program ma mieć możliwość wyboru jednostek metryczne lub imperialne. Użytkownik sam ustala ile obserwacji chce wpisać. Wyniki zapisywane są do pliku tekstowego.
 - c) BMI – charts – program pobiera wyliczone wyniki zapisane w plikach tekstowych (oraz/lub dane wejściowe) i wykonuje ich wizualizacje. Gotowy kod można znaleźć na stronie <https://seaborn.pydata.org/tutorial/introduction.html>.
10. Poszczególne podprogramy mogą być pisane przez innych członków zespołu lub podzespoły. Zawsze należy wystawić pull request z opcją review przez innego członka zespołu. Należy za każdym razem dodawać komentarze (wliczając w to „kod ok, nie mam uwag” itp.)