
TRANSFORMACJE WSPÓŁRZĘDNYCH

INFORMATYKA GEODEZYJNA II
SEM. III, ĆWICZENIA, ROK AKAD. 2023-2024

MAJA KUREK, MONIKA KULIŃSKA
GRUPA 3, NUMER INDEKSU: 325781 , 325780
WYDZIAŁ GEODEZJI I KARTOGRAFII, POLITECHNIKA WARSZAWSKA
Warszawa, 12 maja 2024

Spis treści

1	Wstęp	2
1.1	Cel projektu	2
1.2	Wymagane oprogramowania	2
2	Etapy rozwiązywania	2
3	Podsumowanie, wnioski	3
3.1	Link do repozytorium GitHub	3
3.2	Umiejętności nabyte w trakcie rozwiązywania projektu	3
3.3	Wnioski	3

1 Wstęp

1.1 Cel projektu

Celem ćwiczenia było utworzenie skryptu który będzie wykonywał dane transformacje:

- XYZ -> FLH
- FLH -> XYZ
- XYZ -> NEU
- FL -> PL2000
- FL -> PL1992

Program poza wczytywaniem danych z pliku tekstowego miał także tworzyć plik tekstowy zawierający wyniki transformacji. Skrypt powinien zawierać również klauzule, które naprowadzą użytkownika na poprawne korzystanie w przypadku niewłaściwego użytkowania.

1.2 Wymagane oprogramowania

Do napisania programu wymagane było:

- Python 3.11.5 oraz Python 3.12.3
- Spyder
- Biblioteka numpy, math oraz argparse
- System operacyjny: Microsoft Windows 11 oraz Microsoft Windows 10

2 Etapy rozwiązywania

Rozpoczęliśmy od zdefiniowania klasy 'Transformations', która ściśle współpracuje z biblioteką argparse. Następnie wykorzystując algorytmy napisane przez nas w III semestrze na przedmiocie Geodezja Wyższa oraz korzystając z pomocy naukowych, które dostaliśmy na początku projektu, definiujemy kolejne wymagane transformacje zgodnie ze wzorem **def nazwa_funkcji(self, argumenty)...** Następnie została napisana funkcja wczytująca dane, która wczytuje plik, pomija nagłówek (napisaliśmy flagę, tak aby użytkownik przy wywoływaniu programu mógł wybrać ile linii nagłówka chce pominąć), wykonuje transformacje i ostatecznie zapisuje przekształcone dane do nowego pliku. Do zapisu wyników wykorzystaliśmy funkcję biblioteki numpy "np.savetxt".

Dodano klauzulę **if name == "main"** z biblioteką argparse do obsługi argumentów z wiersza poleceń. Zdefiniowano argumenty: plik, elipsoidę i rodzaj transformacji. Utworzono słowniki dla elipsoidy i transformacji, zapewniając obsługę błędów. Zastosowano klauzulę 'try-except' dla reakcji na nieprawidłowe dane.

3 Podsumowanie, wnioski

3.1 Link do repozytorium GitHub

<https://github.com/MoniA3/IGprojekt1>

3.2 Umiejętności nabyte w trakcie rozwiązywania projektu

- Zapoznanie się z obsługą GitHub i współpraca w dwuosobowym zespole z wykorzystaniem systemu kontroli wersji git
- Zapoznanie się z biblioteką argparse oraz jej zastosowaniem
- Zapoznanie się z pracą w wierszu poleceń
- Tworzenie i pisanie dokumentów w Latex
- Pisanie użytecznej dokumentacji (README.md)
- Pisanie kodu obiektowego w Python

3.3 Wnioski

Wyzwaniem dla nas okazało się również napisanie funkcji do rozpakowywania pliku. Ciężko pisać funkcję na rozpakowanie pliku, którego teoretycznie się nie widzi i nie widać jaki jest postęp w takiej funkcji - chodzi mi tutaj o to, że piszemy uniwersalną funkcję dla użytkownika, a nie taką, która ma być dopasowana pod jeden konkretny plik.

Po zakończeniu pracy przy tym projekcie nasuwa się kilka wniosków:

Praca w GitHub przy tego rodzaju projektach jest bardzo praktyczna oraz intuicyjna. GitHub umożliwia łatwe śledzenie zmian jakie następują w kodzie. GitHub świetnie się sprawdza do współpracy dzięki takim funkcjom jak: proponowanie zmian czy zgłaszanie błędów oraz historia zmian która pozwala drugiej osobie szybko się zorientować jakie zmiany nastąpiły w programie i od czego powinna zacząć dalszą pracę.

Jedną z niewielu trudności sprawiła nam odpowiednio napisanie programu tak aby tworzył się plik z wynikami. Byłyśmy w sytuacji w której program odpowiednio pobierał dane oraz parametry transformacji, kończył działanie ale nie tworzył pliku. Ostatecznie po wielu próbach i reasearchu problem został rozwiązany a program działa bez zarzutów.