

Sprawozdanie

Magdalena Frąckiewicz i Emilia Felczak

10 maja 2024

Spis treści

1	Cel ćwiczenia	2
2	Wykorzystane narzędzia i materiały	2
3	Przebieg ćwiczenia	2
4	Podsumowanie	3
4.0.1	Rezultat	3
4.0.2	Nabyte umiejętności	3
4.0.3	Spostrzeżenia i trudności	3
5	Bibliografia	3

1 Cel ćwiczenia

Celem ćwiczenia jest stworzenie skryptu implementującego transformacje:

- zamiana współrzędnych XYZ na flh
- zamiana współrzędnych flh na XYZ
- zamiana współrzędnych XYZ na neu
- zamiana współrzędnych flh na XY w układzie 2000
- zamiana współrzędnych flh na XY w układzie 1992

2 Wykorzystane narzędzia i materiały

Narzędzia i materiały wykorzystane do replikacji ćwiczenia:

- Python 3.11 (64-bit)
- Python 3.12 (64-bit)
- Spyder
- GitHub
- Wiersz polecenia
- System operacyjny Microsoft Windows 11
- Biblioteka numpy, math
- prezentacje z zajęć (w tym kody z poprzedniego semestru)

3 Przebieg ćwiczenia

Pierwszym krokiem było stworzenie nowego repozytorium w GitHubie, aby móc wykonywać commity. Dzięki temu wspólnie mogliśmy pracować na jednym pliku z różnych komputerów. Następnie stworzyliśmy klasę o nazwie Transformacje, która umożliwiła korzystanie z kilku elipsoid i odpowiednio przypisanych do nich parametrów za pomocą "self.". Importowałyśmy potrzebne biblioteki takie jak: numpy, math, sys. Biblioteka numpy potrzebna jest do obliczeń macierzowych aby uzyskać precyzyjniejsze wyniki i schludniejszy kod. Kolejnym krokiem było napisanie funkcji zamieniających współrzędne. Na podstawie pliku wyjściowego (wsp_inp.txt) mogliśmy sprawdzać poprawność wyników. Było to możliwe dzięki wprowadzeniu do kodu odczytu danych z pliku .txt oraz utworzenia nowego pliku z wynikami. Po zweryfikowaniu działania kodu zajęłyśmy się stylistyką pliku wyjściowego oraz za pomocą sys.argv dałyśmy możliwość korzystania z kodu w wierszu poleceń z zainstalowaną biblioteką numpy. Użytkownik może wybrać flagę opisującą daną funkcję, elipsoidę odniesienia, liczbę wierszy nagłówka w pliku wejściowym oraz plik wejściowy .txt. W rezultacie tworzy się plik wynikowy w tym samym folderze co kod (kod1.py).

4 Podsumowanie

4.0.1 Rezultat

Link do repozytorium GitHub: https://github.com/Emilka2133/PROJEKT_1.git

4.0.2 Nabyte umiejętności

Umiejętności nabyte w trakcie wykonywania ćwiczenia:

- pisanie kodu obiektowego w Pythonie
- implementowanie algorytmów pochodzących ze źródeł zewnętrznych (tj. takich, których nie wymyśliliśmy sami)
- tworzenie dokumentów w latex
- współpraca w dwuosobowym zespole z wykorzystaniem systemu kontroli wersji git (tworzenie repozytorium w GitHub, wykonywanie commitów)
- tworzenie narzędzi w interfejsie tekstowym (cli) potrafiących przyjmować argumenty przy wywołaniu (program jest w stanie pobierać dane z pliku tekstowego)
- pisanie dokumentacji (parametry, wyniki, przebieg i opis funkcji)
- pisanie Readme w GitHub.md

4.0.3 Spostrzeżenia i trudności

Trudności:

Często napotkanym problemem było odpowiednie wcięcie poszczególnych części kodu. Miałśmy trudność z `sys.argv`, aby wszystko poprawnie działało w konsoli. Rozwiązaniem problemu było odpowiednie przyporządkowanie miejsca danej flagi w wierszu poleceń. Przez dłuższy czas wyskakiwał błąd związany z liczbą wierszy w nagłówku. Natomiast udało nam się mu zapobiec poprzez zmianę typu na `integer`. Wszystkie napotkane na drodze komplikacje ostatecznie udało nam się naprawić.

Spostrzeżenia:

Projekt wymagał od nas dużo pracy oraz cierpliwości. Natomiast po skończeniu całego projektu czujemy, że ta praca dała nam nowe możliwości, nauczyła pracować zespołowo oraz poszerzyła nasze umiejętności w zakresie informatyki i geodezji.

5 Bibliografia

Książka wykorzystana w procesie tworzenia projektu do poprawności wzorów:
"Geodezja współczesna Kazimierz Czarnecki