Wydział Geodezji i Kartografii

POLITECHNIKA WARSZAWSKA

Projekt nr 1

Informatyka Geodezyjna
sem. IV, ćwiczenia, rok akad. 2022-2023

Adam Sala, Adrian Rykiel
grupa 3B, Numer Indeksu: 3119369, 319368
adam.sala@pw.edu.pl, adrian.rykiel@pw.edu.pl
Wydział Geodezji i Kartografii, Politechnika Warszawska
Warszawa, 30 kwietnia 2023

Spis treści

1 Cel ćwiczenia

Celem naszego projektu było napisanie skryptu o podanych właściwościach:

- być napisany jako klasa zawierająca metody implementujące poszczególne transformacje,
- posiadać strukturę, w której definicje są oddzielone od wywołań klauzulą podaną przez prowadzącego,
- implementować następujące trnsformacje :
  - XYZ (geocentryczne) -> BLH (elipsoidalne fi, lambda, h)
  - BLH -> XYZ
  - XYZ -> NEUp (topocentryczne northing, easting, up)
  - BL(GRS80, WGS84, ew. Krasowski) -> 2000
  - BL(GRS80, WGS84, ew. Krasowski) -> 1992
- umożliwiać podawanie argumentów przy wywołaniu (biblioteka argparse)
- potrafić transformować wiele współrzędnych zapisanych w pliku tekstowym przekazywanym do programu jako argument i tworzyć plik wynikowy
- obsłużyć przypadki gdy użytkownik wprowadzi niepoprawne wartości (np. nieobsługiwaną elipsoidę)
- być napisany w parach i wersjonowany z użyciem git-a oraz hostowany na githubie w publicznym repozytorium
- udukumentowany na githubie w pliku README.md

2 Wykorzystane narzędzia i materiały potrzebne do replikacji ćwiczenia

Do pracy nad programem użyliśmy następujących narzędzi :

- program był zapisywany i przechowywany w githubie,
- do pisania programu wykorzystaliśmy program Python ,
- naszym system operacyjnym był Windows 11,
- potrzebne informacje do utworzenia podanych transformacji uzyskalismy z stron podanych przez prowadzącego na platformie Teams oraz niezbędne okazały, się materiały z poprzedniego semestru.

3 przebieg ćwiczenia

Zaczęliśmy od implementacji wymaganych metod klasowych. Do tego celu wykorzystaliśmy nasze programy z poprzedniego semestru. Było to też głównym powodem zastosowania bibliotek math oraz numpy, które już tam były w użyciu. Ponadto biblioteka numpy jest powszechnie stosowana, stąd wiele potencjalnych użytkowników już ma ją w posiadaniu i potrzeba jej instalacji nie stanowi żadnego problemu, podobnie jak w przypadku biblioteki wbudowanej pythona math.

Następnym etapem było połączenie wszystkiego w jedną klasę oraz zawarcie wymaganych funkcjonalności związanych z bibioteką argparse oraz wczytywniem i zapisywaniem danych do pliku tekstowego. W związku z tym musieliśmy też wdrożyć modyfikacje do zawartych już metod, tak aby pasowały one do naszej koncepcji wykonania tego ćwiczenia.

Postanowiliśmy bowiem maksymalnie uprościć obsługę naszego programu z myślą o wygodzie użytkownika. Z tego powodu wszystkie dane potrzebne do wykonania obliczeń w naszym przypadku podaje się przy wywołaniu klasy, a nie poszczególnych metod (chyba że wprowadzamy dane z użyciem pliku tekstowego lub za pośrednictwem konsoli). Dzięki temu możliwe też było wykonanie niezbędnych obliczeń pośrednich samoczynnie przez program. Przykładowo, chcąc przeliczyć współrzędne φ, λ, h do układu

PL2000, wystarczy wynonać tylko metdę fl2PL2000(). Program automatycznie przeliczy wtedy wprowadzone wartości kątów na radiany, a następnie przeliczy je na współrzędne X,Y,Z, aby móc wykonać metodę, o którą prosiliśmy.

Z tego samego powodu wszsytkie wprowadzane dane program wrzucana do listy - nawet jednoelementowej, a następnie przy wykonaniu każdej metody obliczenia są wykonywane w pętli dla kolejnych danych zawartych we wspomianych listach. Takie postępowanie drastycznie uprościło procedurę wczytywania plików z danymi. Nie wymaga to bowiem od użytkownika stosowania jakichkolwiek pętli. Wystarczy stworzyć obiekt i wywołać metodę wczytajzpliku().

W międzyczasie zadbaliśmy również o to, aby użytkownikowi zwracany był stosowny komunikat błędu wszędzie tam, gdzie użytkownik może w stosunkowo łatwy spoób wygenerować problem dla naszego programu. W tym celu stworzyliśmy też własną klase błędu NieprawidlowaWartosc, aby nie musieć stosować wbudowanych rodzajów błędów do naszych celów i nie tworzyć tym samym ryzyka, iż użytkownik zostanie wprowadzony w błąd zasugerowawszy się typem zwróconego mu błędu.

W trakcie tworzenia programu niejedokronie niezmiernie pomocne okazywały się źródła zewnętrzne, do których sięgaliśmy w przypadku jakichkowlniek napotkanych trudności z którymi nie mogliśmy sobie poradzić samodzielnie. Polegało to na implementacji znalezionego kodu do naszego programu lub na samodzielnym wprowadzenie poprawek na podstawie znalezionych w takim źródle informacji. Do takich źródeł najczęściej należały `https://stackoverflow.com`, materiały dostarczone przez platformę MS Teams, dokumentacja biblioteki argparse oraz różnego rodzaju poradniki dotyczące programowania obiektowego i biblioteki argparse.

Równie pomocne było korzystanie ze zdalnego repozytorium utworzonego przez nas na pltformie github oraz stosowanie systemu kontroli wersji git. Znacząco uporządkowało i ułatwiło to nam pracę zespołową nad programem przez cały okres wykonywania ćwiczenia.

Po za tym niemalże od samego początku porównywaliśmy wyniki uzyskane z pisanego programu do tych uzyskanych w różnego rodzaju ćwiczeniach z ubiegłego semestru. Ponieważ programy lub wyniki ich obliczeń były werefikowane przez prowadzących, mieliśmy pewność co do ich poprawności. Stąd też mogliśmy na bieżąco werefikować czy zamiany, jakie wprowadziliśmy do kodu nie wpłynęły na poprawność wykonywanych obliczeń.

4 Podsumowanie

- link do naszego repozytorium, z którego można pobrać program:
  `https://github.com/PlaceForNick/projekt_1`
- umiejętności nabyte podczas wykonywania ćwiczenia:
  - pisanie kodu obiektowego w Pythonie
  - implementowanie algorytmów pochodzących ze źródeł zewnętrznych
  - tworzenie dokumentów w latex
  - współpraca w wieloosobowym zespole z wykorzystaniem systemu kontroli wersji git
  - tworzenie narzędzi w interfejsie tekstowym potrafiących przyjmować argumenty przy wywołaniu
  - pisanie użytecznej dokumentacji
- spostrzeżenia i trudności napotkane w trakcie wykonywania ćwiczenia
  - Podczas pracy nad programem nieraz natykaliśmy się na szereg większych lub miniejszych problemów, z którymi koniec końców udało nam się uporać. Jednakże nieudało nam się wytłumaczyć jednego zagadnienia. W pewnym momencie, w trakcie prac nad możliwością podawania argumentów w konsolę wszystkie znaki specjalne (m.in. znaki polskie) zostały przez program zamienione na inne. Dodatkowo u jednego z nas skutkowało zaprzestaniem działania programu oraz zwracaniem błędów na konsolę. (Co jest również ciekawe, dlaczego incydent uniemożliwiał działanie programu tylko na jednym komputerze, a nie u nas obu). W celu pozbycia się problemu musieliśmy pozbyć się problematycznych znaków i starać się ich nie używać w przyszłości. Nie udało nam się bowiem znaleźć żadnego lepszego rozwiązania tego problemu.

Literatura

ASG EUPOS. (2021). *Strona internetowa ASG-EUPOS*. http://www.asgeupos.pl/index.php?wpg_type=tech_transf&sub=xyz_blh.

prof. dr hab. inż. Roman Kadaj. (2001). Opracowanie naukowe. https://ewmapa.pl/dane/wytyczne_g-1.10.pdf.

prof. dr hab. inż. Roman Kadaj. (2002). Opracowanie naukowe. http://www.geonet.net.pl/images/2002_12_uklady_wspolrz.pdf.

ASG EUPOS (2021); prof. dr hab. inż. Roman Kadaj (2001, 2002)