
TRANSFORMACJE WSPÓŁRZĘDNYCH

INFORMATYKA GEODEZYJNA II
SEM. 5, ĆWICZENIA, ROK AKAD. 2023-2024

WYDZIAŁ GEODEZJI I KARTOGRAFII, POLITECHNIKA WARSZAWSKA
Warszawa, 12 maja 2024

Spis treści

1	Wstęp	2
1.1	Cel ćwiczenia	2
1.2	Wykorzystane narzędzia	2
2	Etapy rozwiązywania	2
3	Podsumowanie, wnioski	2
3.1	Link do repozytorium GitHub	2
3.2	Umiejętności nabyte w trakcie rozwiązywania projektu	3
3.3	Spostrzeżenia i trudności napotkane w trakcie ćwiczenia	3
4	Bibliografia	3

1 Wstęp

1.1 Cel ćwiczenia

Celem zadania jest utworzenie skryptu jako klasy, który będzie wykonywał transformacje:

- XYZ -> fi, lam, h
- fi, lam, h -> XYZ
- XYZ -> NEU
- fi, lam -> 2000
- fi, lam -> 1992

1.2 Wykorzystane narzędzia

Do napisania programu przez obu twórców zostały wykorzystane:

- Python 3.11
- Spyder
- System operacyjny: Microsoft Windows 10

2 Etapy rozwiązywania

Krokiem rozpoczynającym nasze działania było zdefiniowanie klasy 'Transformacje'. Następnie do klasy dodaliśmy algorytmy, pozwalające wykonać transformacje podane w punkcie 1.1. Algorytmy te zostały oparte na funkcjach z ćwiczeń Geodezji Wyższej semestru 3.

Następnie przy pomocy `def __init__()`, tworzymy obiekty klasy Transformacje. Zawierają one informacje o parametrach elipsoidy: a- wielka półoś elipsoidy oraz e2 - mimośród elipsoidy.

Następnym krokiem było zaimplementowanie klauzuli *if name == main*, która wykorzystywała "sys.argv". Biblioteka ta umożliwia użytkownikowi przekazywanie argumentów przez wiersz poleceń. Należy tam podać plik(z którego zostaną wyciągnięte dane), elipsoide(jedną z opisanych elipsoid), transformacje(jedną z tych opisanych w klasie).

Po zaimplementowaniu wszystkich transformacji i klauzul, kolejnym etapem było stworzenie plików z przeliczonymi danymi za pomocą komend `open`, `readlines` i `write`.

Końcowym krokiem było uwzględnienie kilku komend i odpowiedzi gdyby użytkownik programu nie zaznajomił się wcześniej z plikiem README.txt. Poniżej przedstawiono kilka przykładów:

- Jeśli poda się nieistniejącą transformację dostaniemy powiadomienie: "Transformation function not recognized."
- Jeśli poda się dwie sprzeczne ze sobą transformacje to dostaniemy powiadomienie; "Only one flag can be added."
- Jeśli poda się model nieuwzględniony w programie to pojawi się wiadomość: "Given model is not supported."

3 Podsumowanie, wnioski

3.1 Link do repozytorium GitHub

<https://github.com/Asiakras/projekt1>

3.2 Umiejętności nabyte w trakcie rozwiązywania projektu

- Zapoznanie się z obsługą GitHub, gdzie możemy w grupie pracować na jednym pliku i zapisywać zmiany widoczne dla wszystkich
- Zapoznanie się ze zmienną `sys.argv` oraz jej zastosowaniem
- Tworzenie dokumentów w latex

3.3 Spostrzeżenia i trudności napotkane w trakcie ćwiczenia

Pierwszym problemem było skorzystanie z `sys.argv`, gdyż jest to nowa zmienna i musieliśmy się chwilę dłużej zastanowić jak ona działa i jak skutecznie jej użyć w tym konkretnym ćwiczeniu. Dłuższą chwilę zastanawialiśmy się czemu kod, który wydawał się poprawny wyrzuca błąd: "An exception has occurred, use `%tb` to see the full traceback.

SystemExit: 1"

Dopiero później zorientowaliśmy się, że w Spyderze nie będziemy mogli skorzystać z tego polecenia(przez `sys.argv`), ale w wierszu poleceń wszystko jest sprawne.

Użycie i stworzenie komendy `header_lines` wywołało delikatne zamieszanie, ponieważ tworzenie funkcji transformacji i ich flag było podzielone między nami. Samą klauzulę wprowadzono praktycznie pod koniec pisania flag powodując różnicę między kodami twórców.

4 Bibliografia

Materiały zajęciowe z ćwiczeń Geodezja Wyższa sem.3 oraz materiały zajęciowe z ćwiczeń Informatyka sem. 4