

Seminarium dyplomowe – Język Python. Podstawy i zastosowania

I. Informacje ogólne

- 1) Tematyka seminarium obejmuje poznanie podstaw programowania oraz wybranych zastosowań języka Python.
- 2) Bardzo proszę o zapoznanie się z materiałem podanym w slajdach na stronie <https://kswider.kia.prz.edu.pl/sem/> oraz tematami referatów do indywidualnego opracowania.
- 3) Slajdy stanowią tylko prezentowane przez prowadzącego na zajęciach wprowadzenie do języka Python. W ramach pracy własnej każda osoba wybiera 1 DOWOLNY temat i przygotowuje ok. 10-15 stronicowe opracowanie pisemne w formie samouczka (tutorial). Oczekuje się, że samouczek w przystępny i przejrzysty sposób, najlepiej na przykładach, rozwinie wybrany temat.
- 4) Wybór tematu należy zgłosić mailowo w możliwie nieodległym terminie. Zakłada się, że opracowania będą gotowe do 20.06. Do tego czasu w trybie stacjonarnym lub zdalnym będzie możliwość konsultowania ich zawartości z prowadzącym.
- 5) W razie pytań bardzo proszę o kontakt mailowy lub telefoniczny.

II. Tematyka opracowań

1. Instrukcje przepływu sterowania

Podejmowanie decyzji za pomocą instrukcji `if`, `if ... else` i `if ... elif ... else`. Iterowanie za pomocą pętli `while` i `for`. Generowanie ciągów liczbowych za pomocą funkcji `range` i iterowanie po tych ciągach za pomocą instrukcji `for`. Tworzenie złożonych wyrażeń boolowskich przy użyciu operatorów `and`, `or` i `not`. Modyfikowanie działania pętli `while` i `for` za pomocą instrukcji `continue` i `break`.

2. Funkcje

Definiowanie własnych funkcji. Importowanie i wykorzystywanie modułów standardowej biblioteki Pythona. Przepływ danych między funkcjami. Zwracanie wieloczęściowego wyniku funkcji za pomocą krotki. Różnicowanie zasięgu widoczności definicji identyfikatora. Przypisywanie wartości domyślnych parametrom definiowanej funkcji. Wywoływanie funkcji z użyciem argumentów kluczowych. Definiowanie funkcji wywoływanych ze zmienną liczbą argumentów. Wywoływanie metod obiektów. Tworzenie i wykorzystywanie funkcji rekurencyjnych.

3. Kolekcje

Listy, krotki, słowniki i zbiory. Definicja, podstawowe operacje i zastosowania.

4. Przetwarzanie danych tekstowych

Łańcuchy znaków i ich metody. Porównywanie łańcuchów za pomocą operatorów. Wyszukiwanie i zastępowanie podłańcuchów. Definiowanie wyrażeń regularnych oraz ich wykorzystywanie do dopasowywania wzorców, wyszukiwania i zastępowania podłańcuchów oraz walidacji danych. Znaczenie obsługi łańcuchów dla przetwarzania języka naturalnego.

5. Wykresy i wizualizacje - biblioteki Matplotlib i Seaborn

Podstawy tworzenia grafik 2D i 3D za pomocą Matplotlib. Opracowanie i prezentacja przykładów użycia biblioteki Seaborn.

6. Obliczenia naukowe - biblioteki NumPy i SymPy

Obiekt `ndarray` jako struktura danych reprezentująca N-wymiarową tablicę w bibliotece NumPy. Funkcje umożliwiające tworzenie oraz manipulowanie tablicami. Funkcje i operatory służące do wykonywania obliczeń na obiektach typu `ndarray`. Wprowadzenie do wspomaganych komputerowo obliczeń symbolicznych z użyciem Pythona i biblioteki SymPy.

7. Przetwarzanie i analiza danych - biblioteki Pandas, statsmodels i scikit-learn

Prezentacja możliwości użycia typów `Series` i `DataFrame`. Podstawy statystyki w Pythonie z użyciem modułu `stats` z biblioteki SciPy. Generowanie liczb losowych. Zmienne losowe, rozkłady statystyczne i testowanie hipotez. Użycie biblioteki statsmodels. Klasy i funkcje do definiowania modeli statystycznych, dopasowywania modeli do danych, obliczania statystyk opisowych oraz przeprowadzania testów statystycznych. Przykłady użycia algorytmów uczenia maszynowego z wykorzystaniem biblioteki scikit-learn.

8. Przetwarzanie języka naturalnego

Zastosowanie biblioteki TextBlob do przetwarzania języka naturalnego (NLP – *natural language processing*). Tokenizacja. Wyodrębnianie części mowy (POS — *Part Of Speech*). Identyfikowanie fraz rzeczownikowych. Analiza sentymentu (ocena wydźwięku emocjonalnego wypowiedzi „pozytywny-obojętny-negatywny”). Tłumaczenie z języka na język ewentualnie w połączeniu z automatycznym wykrywaniem języka. Pluralizacja i singularyzacja słów, czyli zamiana liczby pojedynczej na mnogą i odwrotnie. Stematyzacja i lematyzacja.

9. Programowanie zorientowane obiektowo

Korzyści wynikające z programowania obiektowego. Definiowanie klas i tworzenie obiektów. Kontrolowanie dostępu do atrybutów obiektów i klas. Zalety obiektowego podejścia do programowania. Przeciążanie operatorów. Dziedziczenie klas i relacje „klasa bazowa-podklasa”.

10. Efektywne przetwarzanie dużych ilości danych

Omówienie platformy Apache Spark. Opracowanie i uruchomienie przykładowej aplikacji z wykorzystaniem biblioteki PySpark.