

Kod szkolenia: **PYTHON/ADV**

Tytuł szkolenia: **Zaawansowane programowanie w języku Python**

Dni: 5

## Opis:

Uczestnicy szkolenia zapoznają się z zaawansowanymi mechanizmami programowania obiektowego w języku Python oraz innymi paradygmatami tworzenia oprogramowania. Na zajęciach zapoznają się z koncepcjami tworzenia aplikacji wielowątkowych, kompilacji w locie (JIT), transformowaniem danych XML.

Słuchacze poznają aspekty zalet i wad konstrukcji językowych, API interpretera, przetwarzania struktur grafowych. Szczególny nacisk kładziemy na praktyczne aspekty programowania oraz osadzenie technik wytwarzania oprogramowania w języku Python we współczesnych realiach.

Szkolenie odpowiada potrzebom średnio zaawansowanych programistów i nie jest polecane osobom początkującym.

Zajęcia prowadzone są przez doświadczonych praktyków, którzy na co dzień używają technologie związane z językiem Python w rzeczywistych projektach. W trakcie zajęć używamy technologii i środowisk otwartych.

## Zakres szkolenia

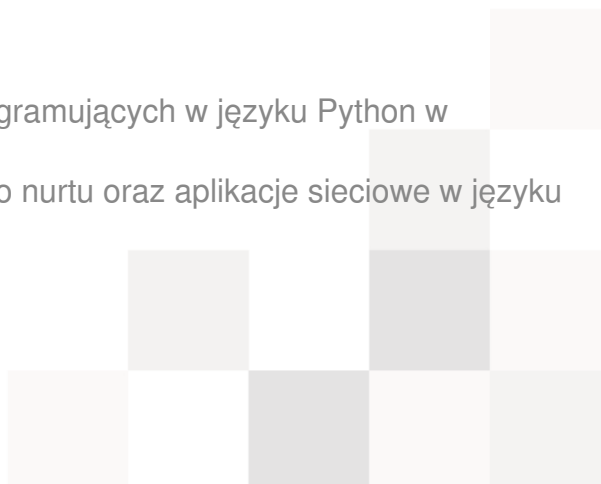
Szkolenie obejmuje:

- programowanie obiektowe w języku Python,
- programowanie wielowątkowe w języku Python,
- transformacje danych w formacie XML,
- wstęp do tworzenia własnych modułów języka Python implementowanych w języku C/C++.

## Adresaci szkolenia

Szkolenie adresowane jest do:

- średnio zaawansowanych programistów programujących w języku Python w środowisku GNU/Linux lub MS Windows,
- programistów tworzących aplikacje głównego nurtu oraz aplikacje sieciowe w języku Python.



## Cel szkolenia

Głównymi celami w procesie szkoleniowym są:

- kształcenie umiejętności i rozwijanie wiedzy dotyczącej zaawansowanego programowania w języku Python,
- osadzenia zaawansowanych narzędzi i procesów tworzenia oprogramowania w realiach współczesnych wymagań Inżynierii Oprogramowania,
- dokonania konsolidacji wiedzy i uzupełnienia braków w kompetencjach w zakresie,
- zapoznanie się z alternatywnymi wersjami interpretera,
- pogłębienie znajomości mechanizmów języka Python.

## Umiejętności zdobywane podczas zajęć

Uczestnicy szkolenia po jego zakończeniu zdobędą następujące umiejętności:

- analizy obiektowej i idiomów językowych Python,
- identyfikacji wąskich gardeł przetwarzania,
- tworzenia aplikacji z użyciem asynchronicznej komunikacji sieciowej,
- tworzenia zaawansowanych struktur danych,
- wybierania i używania paradygmatów programowania.

## Wymagania

Od uczestników szkolenia wymaga się:

- biegłego posługiwania się wybranym środowiskiem IDE (Eclipse, NetBeans, vim),
- znajomości koncepcji programowania obiektowego, systemowego i aplikacyjnego,
- znajomości narzędzi do tworzenia aplikacji z rodziny gcc i binutils,
- umiejętności posługiwania się narzędziem debuggera języka Python,
- znajomości koncepcji związanych z programowaniem w języku Python na poziomie średnio i zaawansowanym.

## Parametry szkolenia

Szkolenie trwa 5 dni. Szkolenie obejmuje 60% wykładu i 40% ćwiczeń.

## Program szkolenia:

1. Metaprogramowanie w języku Python
  - I. Diament dziedziczenia
  - II. MRO (ang. Method Resolution Order)
  - III. Klasy domieszkowe
  - IV. Rozszerzone wzorce GoF – zakres zastosowań i implementacja
  - V. Mechanizm metaklas
2. Dekoratory



- I. Działanie i zakres zastosowania dekoratora
  - II. Biblioteka dekoratorów
  - III. Implementacja własnych dekoratorów
3. Kontekst wykonania
  - I. Działanie i implementacja kontekstu wykonania (with)
  - II. Zakres zastosowań kontekstu wykonania
4. Deskryptory
  - I. Pojęcie i stosowanie deskryptorów
  - II. Zakres stosowania deskryptorów w języku Python
  - III. Implementowanie własnych deskryptorów
  - IV. Deskryptory a reguły S.O.L.I.D. i GRASP
5. Mechanizmy i konwencje
  - I. Konwencja tworzenia poprawnego kodu (pep8 i pylint)
  - II. Wstrzykiwanie zależności
  - III. Faktorie
  - IV. Callback
  - V. Duck Typing
  - VI. Delegacja i „owijanie” zamiast dziedziczenia
6. Wydajność struktur danych
  - I. Wbudowane struktury i typy danych – aspekt wydajności
  - II. Moduł collections
  - III. Wydajność typowania dynamicznego
7. Idiomy języka
  - I. Leniwe przetwarzanie
  - II. Domknięcia, generatory i wyrażenia lambda
  - III. Mechanizmy introspekcji
8. Alternatywne wersje interpretera/kompilatora
  - I. Kompilacja do kodu natywnego
  - II. Kompilacja w locie (JIT – ang. Just In Time)
  - III. Mikro-wątki
9. Python – moduły w języku C i C++
  - I. Narzędzia integracji modułów w języku C i C++
  - II. API interpretera
  - III. Pułapki interpretera
10. Programowanie wielowątkowe
  - I. Procesy i wątki
  - II. Mechanizmy blokujące
  - III. Komunikacja oparta o pamięć współdzieloną – aspekty wielowątkowości
  - IV. Współprogramy (gevent i eventlet)
11. Paradygmaty programowania w języku Python
  - I. Maszyna stanu
  - II. Programowanie funkcyjne
  - III. Programowanie deklaratywne
  - IV. Programowanie zdarzeniowe
12. Programowanie sieciowe
  - I. Asynchroniczna komunikacja sieciowa



- II. Obsługa protokołów rozgłaszania
- 13. Zaawansowane struktury danych
  - I. Przetwarzanie struktur grafowych
  - II. Kolejki oraz kolejki rozproszone
- 14. Przetwarzanie i transformacja danych XML
  - I. Parsowanie danych XML
  - II. Transformaty XSLT
  - III. Obsługa i przetwarzanie danych ElementTree i SAX
- 15. Testowanie w języku Python
  - I. Biblioteki i metody testowania
  - II. Testowanie jednostkowe z użyciem unittest, oraz pytest
  - III. Szybkie tworzenie testów z użyciem doctest
  - IV. Osadzenie testów w środowisku integracji ciągłej
- 16. Python w środowisku produkcyjnym
  - I. Standardy inspekcji i metryki kodu
  - II. Reguły tworzenia i dostarczania (ang. deploy) aplikacji
  - III. Python w zastosowaniach SOA
- 17. Języki DSL (ang. Domain Specific Language ) (moduł opcjonalny)
  - I. Drzewo AST (ang. Abstract Syntax Tree )
  - II. Definicja własnych języków dziedzinowych
- 18. Maszyna wirtualna języka Python (moduł opcjonalny)

