

Techniki przetwarzania i wizualizacji danych w MATLABie. Techniki programowania w MATLABie

Tematyka szkolenia:

Szkolenie dotyczące importowania i przygotowania danych dla aplikacji analitycznych. Kurs jest przeznaczony dla osób, które chcą zautomatyzować przetwarzanie, analizę i wizualizację danych z różnych źródeł. Szkolenie pokazuje jak wykorzystać funkcjonalność oprogramowania MATLAB w celu napisania wydajnego, niezawodnego i dobrze zorganizowanego kodu. Omawiane koncepcje stanowią podstawę do pisania pełnych aplikacji, opracowywania algorytmów i rozszerzania wbudowanych funkcji MATLAB'a. Omówiono szczegóły optymalizacji wydajności, a także narzędzia do pisania, debugowania i profilowania kodu.

- metody importu danych,
- przetwarzanie danych,
- dostosowywanie sposobu wizualizacji,
- praca z nieregularnymi danymi
- wykorzystanie narzędzi środowiska MATLAB do rozwijania aplikacji,
- efektywne zarządzanie danymi,
- techniki strukturyzacji danych,
- techniki strukturyzacji kodu,
- tworzenie aplikacji odpornych na błędy,
- weryfikacja zachowania aplikacji.

Plan szkolenia:

Dzień 1 z 3

Import danych

Cel: Wczytywanie plików tekstowych zawierających różne typy danych, separatory i nagłówki.

- Importowanie zróżnicowanych typów danych z dowolnie sformatowanych plików tekstowych
- Importowanie wyłącznie potrzebnych kolumn z pliku tekstowego
- Importowanie i scalanie danych z kilku plików

Przetwarzanie danych

Cel: Przetwarzanie surowych danych poprzez wyodrębnianie, modyfikowanie, agregowanie i zliczanie porcji danych.

- Przetwarzanie danych z brakującymi elementami
- Tworzenie i modyfikowanie tablic kategoriycznych
- Agregowanie komórek i zliczanie grup danych

Modyfikowanie wizualizacji

Cel: Komentowanie i modyfikowanie standardowych wykresów w celu stworzenia czytelnych grafik zawierających wiele informacji.

- Określanie właściwości obiektów graficznych i związanych z nimi danych
- Umiejscawianie i przemieszczanie obiektów graficznych
- Tworzenie własnych wykresów poprzez modyfikację parametrów obiektów graficznych

Działania na nieregularnych danych

Cel: Importowanie i wizualizacja rozproszonych danych z plików tekstowych o nieregularnym formatowaniu.

- Analiza plików tekstowych w celu określenia formatowania
- Import danych z oddzielnych sekcji pliku tekstowego
- Wyodrębnianie danych z kontenerów danych
- Interpolowanie nieregularnie rozmieszczonych danych trójwymiarowych
- Wizualizacja danych 3D w dwóch i trzech wymiarach

Dzień 2 z 3

Strukturyzacja danych

Cel: Zbadanie możliwości przechowywania danych w MATLAB'ie

- Typy danych
- Konkatenacja i konwersja
- Tablice komórkowe i struktury
- Wydobywanie danych

Efektywne zarządzanie danymi

Cel: Używanie typowych technik zwiększania wydajności podczas przechowywania, uzyskiwania dostępu i przetwarzania danych.

- Wstępna alokacja
- Wektoryzacja
- Zarządzanie pamięcią
- Obiekty i klasy

Wykorzystanie narzędzi deweloperskich

Cel: Używanie zintegrowanych narzędzi do zarządzania kodem, diagnozowania problemów, debugowania aplikacji i mierzenia wydajności.

- Zarządzanie plikami za pomocą projektów
- Znajdowanie potencjalnych problemów w kodzie
- Debugowanie kodu
- Ocena wydajności kodu
- Korzystanie z kontroli źródła

Dzień 3 z 3

Tworzenie niezawodnych aplikacji

Cel: Tworzenie niezawodnych aplikacji, które obsługują nieoczekiwane dane wejściowe i generują jednoznaczne błędy. Używanie wbudowanych funkcji MATLAB'a i konstrukcji programistycznych oraz stosowanie standardowych technik obsługi błędów.

- Tworzenie elastycznych interfejsów funkcji
- Sprawdzanie warunków błędów
- Praca z konstrukcją try-catch i obiektami MException

Strukturyzacja kodu

Cel: Modularyzacja kodu w czytelne i łatwe do utrzymania funkcje. Poznawanie różnych typów funkcji dostępnych w języku MATLAB i kompromisy związane z każdym typem. Sprawdzanie jak typ funkcji wpływa na wydajność i niezawodność kodu.

- Funkcje prywatne
- Funkcje lokalne
- Funkcje zagnieżdżone
- Uchwyty funkcji
- Zasady pierwszeństwa
- Porównanie typów funkcji

Weryfikacja zachowania aplikacji

Cel: Tworzenie testów w celu sprawdzenia, czy kod spełnia wymagania.

- Pisanie skryptów testowych
- Uruchamianie skryptów testowych
- Korzystanie z funkcji testowych

Tworzenie Toolbox'ów

Cel: Spakowanie kodu i dokumentacji do niestandardowego toolbox'a przygotowanego do dystrybucji wśród innych użytkowników

- Tworzenie dokumentacji
- Pakowanie i dystrybucja zestawu narzędzi

Test sprawdzający dla uczestników szkolenia

Cel: Weryfikacja wiedzy i umiejętności zdobytej podczas kursu.