

Kod szkolenia: **OCP**

Tytuł szkolenia: **Przygotowanie do uzyskania certyfikatu OCP - Oracle Certified Professional**

Preparation for OCP exam - Oracle Certified Professional

Dni: **3**

Opis:

## Adresaci szkolenia

Szkolenie adresowane jest do osób chcących przystąpić do egzaminu Oracle Certified Professional oraz chcących usystematyzować wiedzę w zakresie programowania w języku Java.

## Cel szkolenia

Celem szkolenia jest nabycie wiedzy i umiejętności programowania w języku Java w stopniu pozwalającym na przystąpienie do egzaminu uprawniającego do otrzymania certyfikatu OCP. Szkolenie zostało przygotowane tak, by szczegółowo przedstawić (zarówno od strony teoretycznej, jak i praktycznej) zagadnienia obejmowane przez egzamin, zwracając jednocześnie uwagę uczestników na specyficzne aspekty poszczególnych elementów języka Java.

## Mocne strony szkolenia

Szkolenie jest stale uaktualniane do pojawiających się kolejnych wersji języka w oparciu o wymagania najnowszych wersji certyfikatów.

W trakcie kursu uczestnicy przyglądają się specyficznym aspektom języka zawartym w egzaminach, poznają przykładowe pytania i realizują próbne testy cząstkowe i całościowe.

Szkolenie jest prowadzone przez trenerów, którzy uzyskali certyfikat OCP i wiedzą jak się do niego przygotować.

Oprócz przygotowania do egzaminu szkolenie pomaga usystematyzować wiedzę dla programistów Java oraz może być sposobem na migrację do Javy z innego języka programowania.

## Wymagania

Od uczestników wymagana jest znajomość zagadnień obejmowanych przez egzamin OCA. Przystąpienie do egzaminu OCP wymaga uzyskania certyfikatu OCA.

## Parametry szkolenia

Wykłady oraz ćwiczenia praktyczne obejmujące zagadnienia egzaminacyjne. Ich uzupełnieniem są pytania, podsumowujące każdą sekcję, zbliżone formą oraz trudnością do egzaminu. Dodatkowo udostępniany jest kod pokazujący w praktyce poruszane na szkoleniu aspekty.

### Program szkolenia:

1. Zaawansowane pojęcia związane z klasami
  - Użycie klas: static inner class, local class, nested class, anonymous inner class
  - Użycie typu enum
2. Interfejsy
  - Tworzenie i wykorzystanie interfejsów oraz metod default
  - Tworzenie i użycie metod prywatnych w interfejsach
3. Interfejsy funkcyjne i wyrażenia Lambda
  - Definiowanie oraz tworzenie interfejsów funkcyjnych
  - Tworzenie i użycie wyrażen lambda
4. Interfejsy funkcyjne w JDK
  - Użycie interfejsów funkcyjnych z pakietu java.util.function
  - Użycie interfejsów Predicate, Consumer, Function i Supplier
  - Użycie prymitywnych oraz binarnych odmian interfejsów funkcyjnych
5. Migracja do aplikacji modułowej
  - Migracja aplikacji napisanych przed Javą 9 do postaci modułowej
  - Użycie jdeps do definiowania zależności i wykrywanie zależności cyklicznych
6. Wielowątkowość
  - Tworzenie wątków przy użyciu Runnable, Callable oraz użycie ExecutorService
  - Użycie kolekcji z pakietu java.util.concurrent
  - Pisanie kodu bezpiecznego wątkowo
  - Identyfikacja problemów z wielowątkowością, takich jak livelock i deadlock
7. I/O oraz NIO2
  - Czytanie danych z konsoli oraz plików przy użyciu I/O Streams
  - Czytanie i zapisywanie obiektów z wykorzystaniem serializacji/ deserializacji
  - Użycie interfejsu Path
  - Użycie klasy Files do operacji na plikach
  - Użycie strumieni do pracy z plikami
8. JDBC
  - Połączenie do bazy danych przy użyciu JDBC

- Użycie PreparedStatement do wykonywania operacji CRUD
  - Wykorzystanie PreparedStatement i CallableStatement do realizacji operacji na bazie danych
9. Adnotacje
    - Cel oraz użycie adnotacji w Javie
    - Zastosowanie adnotacji w klasach i metodach
    - Identyfikacja najpopularniejszych adnotacji w JDK
    - Definiowanie własnych adnotacji
  10. Obsługa wyjątków oraz asercje
    - Użycie konstrukcji try-with-resources
    - Tworzenie własnych wyjątków
    - Wykorzystanie asercji
  11. Typy generyczne i kolekcje
    - Użycie klas wrapperów, auto-boxing i auto-unboxing
    - Tworzenie i użycie klas generycznych oraz metod z wykorzystaniem typów generycznych
    - Wykorzystanie głównych typów kolekcji
    - Użycie interfejsu Comparator i Comparable
  12. Stream API
    - Użycie interfejsu Stream
    - Użycie wyrażeń Lambda i referencji do metod
  13. Wykonywanie operacji na strumieniach z wykorzystaniem wyrażeń Lambda
    - Wykorzystanie metod map, flatMap, peek
    - Wyszukiwanie z użyciem findFirst, findAny, anyMatch, allMatch, noneMatch
    - Użycie klasy Optional
    - Wykonywanie obliczeń z użyciem metod count, max, min, average i sum
    - Sortowanie z użyciem wyrażeń Lambda
    - Użycie kolektorów
  14. Serwisy w aplikacji modułowej
    - Opisanie komponentów serwisów
    - Zaprojektowanie serwisu z użyciem ServiceLoader oraz sprowadzenie zależności
  15. Parallel Streams
    - Pisanie kodu z użyciem parallel streams
    - Zaimplementowanie dekompozycji oraz redukcji
  16. Bezpieczny kod
    - Poufne informacje w aplikacji napisanej w Javie
    - Implementowanie wytycznych dotyczących spójności danych
    - Zapobieganie atakom poprzez ograniczenie dostępności i rozszerzalności, walidację danych wejściowych oraz mutację
    - Bezpieczne tworzenie wrażliwych obiektów
    - Bezpieczna serializacja i deserializacja
  17. Lokalizacja
    - Użycie klasy Locale
    - Użycie ResourceBundle
    - Formatowanie komunikatów, dat oraz liczb

