

SZKOLENIE ŚREDNIO ZAAWANSOWANE

Przygotowanie do uzyskania certyfikatu OCP Java Developer

OCPJD

Czas trwania: 5 dni (40h)

Najpopularniejszy i uznany na rynku certyfikat dla programistów Java

Cele szkolenia

- Nabycie wiedzy i umiejętności programowania w języku Java w stopniu pozwalającym na przystąpienie do egzaminu uprawniającego do otrzymania certyfikatu OCPJD
- Szczegółowe przedstawienie (zarówno od strony teoretycznej, jak i praktycznej) zagadnień obejmowanych przez egzamin, zwracając jednocześnie uwagę uczestników na specyficzne aspekty poszczególnych elementów języka Java

Zalety

- 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 OCPJD 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
- Praktyka przed teorią - wszystkie szkolenia technologiczne prowadzone są w formie warsztatowej. Konieczna teoria jest wyjaśniana na przykładzie praktycznych zadań
- Konkretność umiejętności - w ramach każdego szkolenia rozwijamy praktyczne umiejętności związane z daną technologią i tematyką
- Nauka z praktykami - wszyscy trenerzy na co dzień pracują w projektach, gwarantuje to dostęp do eksperckiej wiedzy i praktycznego know-how

Dla kogo?

- Osoby chcące przystąpić do egzaminu Oracle Certified Professional
- Osoby chcące usystematyzować wiedzę w zakresie programowania w języku Java

Wymagania



-
- Od uczestników wymagana jest umiejętność programowania w stopniu podstawowym w dowolnym obiektowym języku

Program

1. Wprowadzenie
 - a. Podstawowe informacje o certyfikacie OCP Java Developer
 - b. Podejście do egzaminu - formalności
 - c. Zasoby internetowe
2. Podstawy języka Java
 - a. Struktura klas, interfejsów i ich komponentów
 - b. Wykonywalne klasy Java, metoda main()
 - c. Pakiety, importowanie klas, import statyczny
 - d. Zakres widoczności, modyfikatory
3. Typy danych
 - a. Deklaracja i inicjalizacja zmiennych, identyfikatory
 - b. Zmienne typów prostych, tablicowych, wyliczeniowych, klasy opakujące
 - c. Zmienne typów obiektowych, referencje do obiektów
 - d. Operatory przypisania, arytmetyczne, logiczne, priorytet operatorów
4. Instrukcje sterujące wykonaniem programu
 - a. Instrukcje warunkowe: if, else, switch
 - b. Pętle for, for-each, while, do-while
 - c. Instrukcje break, continue, wykorzystanie etykiet
5. Operacje na tablicach
 - a. Tablice obiektów, deklaracja, alokacja, inicjalizacja
 - b. Tablice wielowymiarowe
6. Pola, metody i enkapsulacja
 - a. Zasięg zmiennych
 - b. Zmienne składowe obiektu i klasy (pola)
 - c. Deklaracja metod, parametry, wartość zwracana
 - d. Przekazywanie argumentów do metod
 - e. Przeładowanie metod (overload)
 - f. Konstruktory, bloki inicjalizacyjne
 - g. Cykl życia obiektu
 - h. Enkapsulacja
7. Dziedziczenie
 - a. Dziedziczenie z użyciem klas i interfejsów
 - b. Dziedziczenie a zakres widoczności
 - c. Dziedziczenie a przypisania, operator instanceof
 - d. Używanie konstruktorów i metod nadklasy
 - e. Polimorfizm, rzutowanie, overload a override
8. Wyjątki i asercje
 - a. Koncepcja sytuacji wyjątkowych w Java
 - b. Typy sytuacji wyjątkowych (klasy Exception, RuntimeException, Error)



- c. Obsługa wyjątków (klauzule try, catch, multi-catch, finally, try-with-resources)
- d. Przegląd wybranych wyjątków
- e. Tworzenie zasobów auto-closeable
- f. Wykorzystanie asercji
- 9. Praca z podstawowymi klasami z JDK
 - a. Klasa String, StringBuffer, StringBuilder
 - b. Operacje na napisach, równość napisów
 - c. Interfejs List, klasa ArrayList
- 10. Przykładowy test OCA
- 11. Zaawansowane pojęcia związane z klasami
 - a. Użycie klas: static inner class, local class, nested class, anonymous inner class
 - b. Użycie typu enum
 - c. Tworzenie i używanie wyrażeń lambda
 - d. Podstawowe wzorce projektowe
- 12. Kolekcje oraz typy generyczne
 - a. Tworzenie i wykorzystywanie klas ArrayList, TreeSet, TreeMap, ArrayDeque
 - b. Wykorzystanie interfejsów java.util.Comparator i java.lang.Comparable
 - c. Koncepcja strumieni w kontekście kolekcji
 - d. Tworzenie i wykorzystywanie typów generycznych
- 13. Interfejsy funkcyjne oraz Stream API
 - a. Wyrażenie lambda
 - b. Przegląd interfejsów funkcyjnych w pakiecie java.util.function
 - c. Wersje interfejsów funkcjonalnych dla typów prostych
 - d. Wyszukiwanie danych przy użyciu metod findFirst(), findAny(), anyMatch(), allMatch(), noneMatch()
 - e. Sortowanie danych
 - f. Użycie metod z klasy Collectors
 - g. Wykorzystanie klasy Optional
- 14. Zagadnienia związane z Date/Time API
 - a. Użycie klas LocalDate, LocalTime, LocalDateTime, Instant, Period i Duration
 - b. Zarządzanie kodem obsługującym różne strefy czasowe
 - c. Formatowanie dat
- 15. Obsługa Wejścia/Wyjścia
 - a. Koncepcja strumieni w Java
 - b. Rodzaje strumieni
 - c. Strumienie Binarne (InputStream, OutputStream i pochodne)
 - d. Strumienie Znakowe (Reader, Writer i pochodne)
 - e. Zagadnienia kodowania w strumieniach znakowych
 - f. Łączenie strumieni
 - g. Strumienie skojarzone z plikami oraz interakcja z systemem plików klasy (File, RandomAccessFile, FileOutputStream, FileInputStream)
 - h. Interfejs Serializable
 - i. Serializacja obiektów do strumieni (ObjectOutputStream, ObjectInputStream)
 - j. Modyfikacja domyślnej serializacji



- k. Standardowe wejście/wyjście
- l. Przykłady strumieni specjalizowanych
- 16. File I/O z NIO.2
 - a. Operowanie na ścieżkach do plików i katalogów z użyciem klasy Path
 - b. Kopiowanie, przenoszenie, usuwanie plików z użyciem klasy Files
 - c. Rekurencyjna nawigacja po drzewie katalogów
 - d. Odczyt i zmiana atrybutów plików i katalogów
 - e. Znajdowanie plików przy użyciu klasy PathMatcher
- 17. Podstawy JDBC
 - a. Rejestracja sterownika
 - b. Wykonywanie zapytań
 - c. Transakcje
 - d. Projektowanie warstwy dostępu do danych
- 18. Wydajność i zarządzanie zasobami
 - a. Struktura pamięci
 - b. Metoda finalize i Garbage Collector
 - c. Monitorowanie zasobów wirtualnej maszyny
- 19. Programowanie aplikacji wielowątkowych
 - a. Konstrukcja wątków w Javie (Runnable i Thread)
 - b. Cykl życia wątku
 - c. Scheduler (zasada działania i priorytety)
 - d. Metody yield, join, sleep
 - e. Grupy wątków
 - f. Zagadnienia synchronizacji (wait, notify, notifyAll, zakleszczenie, zagłodzenie)
 - g. Klasy pakietu java.util.concurrent
 - h. Wykonawcy i pule
 - i. Fork/Join
- 20. Internacjonalizacja i lokalizacja aplikacji
 - a. Odczyt i zmiana ustawień językowych
 - b. Tworzenie pakietów zasobów dla poszczególnych języków
 - c. Ładowanie i używanie zasobów językowych w aplikacji
 - d. Formatowanie danych z użyciem NumberFormat, DateFormat
- 21. Moduły Java
 - a. Java Platform Module System
 - b. Architektura modułów
 - c. Deskryptor modułu
 - d. Usługi
- 22. Przykładowy test OCP

