

Kod szkolenia: **OCAP**

Tytuł szkolenia: **Programowanie w języku Java (OCA+OCP)**

Przygotowanie do uzyskania certyfikatów OCAJP oraz OCPJP

Dni: 5

Opis:

Adresaci szkolenia

Szkolenie adresowane jest do osób chcących przystąpić do egzaminów certyfikacyjnych Oracle Certified Associate, Java SE Programmer oraz Oracle Certified Professional, Java SE Programmer.

Cel szkolenia

Celem szkolenia jest nabycie wiedzy i umiejętności programowania w języku Java w stopniu pozwalającym na przystąpienie do egzaminów urawniających do otrzymania certyfikatów OCAJP i OCPJP. Szkolenie zostało przygotowane tak, by zarówno ze strony teoretycznej, jak i praktycznej przedstawić zagadnienia objęte certyfikatami, zwracając jednocześnie uwagę uczestników na specyficzne aspekty poszczególnych elementów języka Java.

Materiał szkolenia jest rozłożony na 5 dni wykładów i warsztatów. Specjalna konstrukcja programu szkolenia umożliwi uczestnictwo w ramach jednego z trzech wariantów:

- **w pełnym wymiarze (5 dni) - całościowe przygotowanie zarówno do OCAJP, jak i OCPJP,**
- w części podstawowej (pierwsze 3 dni) - dla osób chcących przygotować się jedynie do OCAJP ([zobacz](#)),

- **Adresaci szkolenia**

Szkolenie adresowane jest do osób chcących przystąpić do egzaminów certyfikacyjnych Oracle Certified Associate, Java SE Programmer oraz Oracle Certified Professional, Java SE Programmer.

Cel szkolenia

Celem szkolenia jest nabycie wiedzy i umiejętności programowania w języku Java w stopniu pozwalającym na przystąpienie do egzaminów urawniających do otrzymania certyfikatów OCAJP i OCPJP. Szkolenie zostało przygotowane tak, by zarówno ze

strony teoretycznej, jak i praktycznej przedstawić zagadnienia objęte certyfikatami, zwracając jednocześnie uwagę uczestników na specyficzne aspekty poszczególnych elementów języka Java.

Materiał szkolenia jest rozłożony na 5 dni wykładów i warsztatów. Specjalna konstrukcja programu szkolenia umożliwi uczestnictwo w ramach jednego z trzech wariantów:

- **w pełnym wymiarze (5 dni) - całościowe przygotowanie zarówno do OCAJP, jak i OCPJP,**
- w części podstawowej (pierwsze 3 dni) - dla osób chcących przygotować się jedynie do OCAJP ([zobacz](#)),
- w części zaawansowanej (ostatnie 3 dni) - dla osób posiadających już wiedzę wymaganą przez certyfikat OCAJP i zainteresowanych zakresem certyfikatu OCPJP. ([zobacz](#)).

Mocne strony szkolenia

Program pokrywa zagadnienia objęte certyfikatami OCAJP oraz OCPJP.

Szkolenie jest ciągle 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.

Wymagania

Od uczestników wymagana jest umiejętność programowania w dowolnym języku.

Parametry szkolenia

5*8 godzin (5*7 godzin netto) wykładów i warsztatów (z wyraźną przewagą warsztatów). W trakcie warsztatów projektowana i implementowana jest prosta aplikacja, obejmująca kluczowe zagadnienia poruszane na szkoleniu. Prezentowane są także przykładowe pytania, zbliżone formą i poziomem trudności do tych, mogących pojawić się na egzaminach.

Wielkość grupy: maks. 8-10 osób.

([zobacz](#)).

Mocne strony szkolenia

Program pokrywa zagadnienia objęte certyfikatami OCAJP oraz OCPJP.



Szkolenie jest ciągle 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.

Wymagania

Od uczestników wymagana jest umiejętność programowania w dowolnym języku.

Parametry szkolenia

5*8 godzin (5*7 godzin netto) wykładów i warsztatów (z wyraźną przewagą warsztatów). W trakcie warsztatów projektowana i implementowana jest prosta aplikacja, obejmująca kluczowe zagadnienia poruszane na szkoleniu. Prezentowane są także przykładowe pytania, zbliżone formą i poziomem trudności do tych, mogących pojawić się na egzaminach.

Wielkość grupy: maks. 8-10 osób.

Program szkolenia:

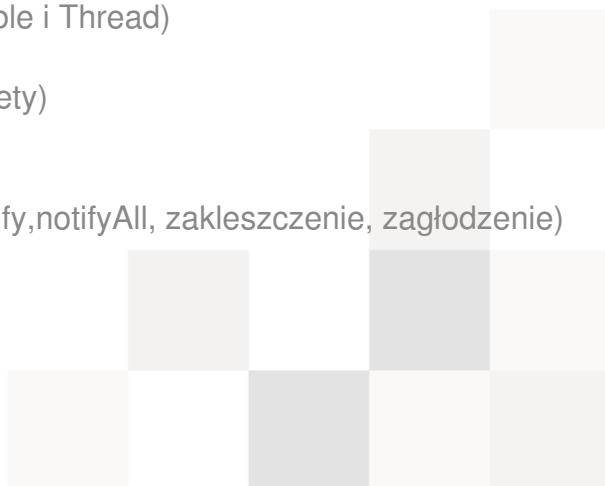
1. Wprowadzenie
 - a. Podstawowe informacje o certyfikatach OCA i OCP
 - b. Cele certyfikatów OCA i OCP
 - c. Zasoby internetowe
2. Podstawy języka Java
 - a. Struktura klas, interfejsów i ich komponentów
 - b. Wykonywalne klasy Java, metoda main()
 - c. Pakiety, importowanie klas, import statyczny
 - d. Zakres widoczności, modyfikatory
3. Typy danych
 - a. Deklaracja i inicjalizacja zmiennych, identyfikatory
 - b. Zmienne typów prostych, tablicowych, wyliczeniowych
 - c. Klasy opakowujące
 - d. Zmienne typów obiektowych, referencje do obiektów
 - e. Operatory przypisania, arytmetyczne, logiczne, priorytet operatorów
4. 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
5. Instrukcje sterujące wykonaniem programu



- a. Instrukcje warunkowe: if, else, switch
- b. Pętle for, for-each, while, do-while
- c. Instrukcje break, continue, wykorzystanie etykiet
6. Operacje na napisach i tablicach, równość obiektów
 - a. Klasa String, StringBuffer, StringBuilder
 - b. Operacje na napisach, równość napisów
 - c. Tablice obiektów, deklaracja, alokacja, inicjalizacja
 - d. Tablice wielowymiarowe
 - e. Interfejs List, klasa ArrayList
 - f. Równość obiektów, metody equals() i hashCode()
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. Obsługa wyjątków
 - a. Koncepcja sytuacji wyjątkowych w Java
 - b. Typy sytuacji wyjątkowych (klasy Exception, RuntimeException, Error)
 - c. Obsługa wyjątków (klauzule try, catch, multi-catch, finally)
 - d. Przegląd wybranych wyjątków
9. Zasady projektowania zorientowanego obiektowo
 - a. Stosowanie klas i metod abstrakcyjnych
 - b. Klasy niemodyfikowalne (immutable)
 - c. Klasy zagnieżdżone, klasy wewnętrzne
 - d. Podstawy UML
 - e. Rodzaje związków między obiektami: agregacja, kompozycja, asocjacja, generalizacja
 - f. Zasady używania dziedziczenia i kompozycji
 - g. Podstawowe wzorce projektowe (Singleton, DAO, Fabryka)
10. Zaawansowana obsługa wyjątków
 - a. Deklaracja metod zgłaszających wyjątki oraz zgłaszanie wyjątków (throws, throw)
 - b. Klauzula try-with-resources, automatyczne zamykanie zasobów
 - c. Wyjątki a dziedziczenie
 - d. Projektowanie własnych typów wyjątków
 - e. Testowanie niezmienników za pomocą asercji
11. Kolekcje i klasy Generyczne
 - a. Rodzaje kolekcji w Java
 - b. Kolekcje uporządkowane, nieuporządkowane, posortowane, nieposortowane
 - c. Interfejsy Collection, Set, Map, List, Deque
 - d. Przegląd dostępnych kolekcji
 - e. Kolekcje synchronizowane
 - f. Iteracja po kolekcjach
 - g. Interfejsy Comparable i Comparator
 - h. Metody użytkowe z klasy Collections



- i. Interfejs Clonable
 - j. Kolekcje Generyczne
 - k. Zasady tworzenia klas Generycznych
12. Przykładowy test OCAJP
13. Obsługa Wejścia/Wyjścia
- a. Koncepcja strumieni w Java
 - b. Rodzaje strumieni
 - c. Strumienie Binarne (InputStream, OuputStream 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
 - m. Atomizacja strumieni (StreamTokenizer)
14. 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
 - f. Obserwowanie zawartości katalogu za pomocą WatchService
15. Podstawy JDBC
- a. O JDBC
 - b. Rejestracja sterownika
 - c. Wykonywanie zapytań
 - d. Transakcje
 - e. Projektowanie warstwy dostępu do danych
16. Wydajność i zarządzanie zasobami
- a. Struktura pamięci
 - b. Metoda finalize i Garbage Collector
 - c. Monitorowanie zasobów wirtualnej maszyny
17. Programowanie aplikacji wielowątkowych
- a. Pojęcie wątku
 - b. Konstrukcja wątków w Javie (Runnable i Thread)
 - c. Cykl życia wątku
 - d. Scheduler (zasada działania i priorytety)
 - e. Metody yield, join, sleep
 - f. Grupy wątków
 - g. Zagadnienia synchronizacji (wait,notify,notifyAll, zakleszczenie, zagłodzenie)
 - h. Klasy pakietu java.util.concurrent
 - i. Wykonawcy i pule



- j. Fork/Join
- 18. 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
- 19. Przykładowy test OCPJP

