**Centrum Szkoleniowe Comarch**
ul. Prof. Michała Życzkowskiego 33
31-864 Kraków

COMARCH
Szkolenia

# Programowanie w języku Java –

## kompendium wiedzy

## Cele szkolenia

Intensywny 5-dniowy kurs programowania w języku JAVA.  Kurs oprócz podstaw języka obejmuje m.in. podstawy Java Reflection API, JDBC oraz elementy JEE.  Szkolenie pokrywa się z większością zagadnień objętych certyfikatem OCPJP.

## Umiejętności

Po ukończeniu kursu uczestnik/czka pozna:

- Podstawy języka JAVA
- Programowanie webowych z wykorzystaniem frameworka Spring
- Programowanie sieci TCp/IP
- Programowanie aplikacji wielowątkowych

## Profil uczestników

Szkolenie skierowane jest do programistów chcących poznać język JAVA, pozwala na wsparcie przygotowań do certyfikacji OCPJP.

## Przygotowanie uczestników

Z uwagi na ilość realizowanych zagadnień oraz intensywność ich realizacji niezbędna jest umiejętność programowania w dowolnym języku.

## Szczegółowy program szkolenia

**1.Wprowadzenie**

- Platforma i język Java: najważniejsze pomysły i cechy charakterystyczne
- Edycje Javy i rodzaje instalacji
- Narzędzia Java SE: kompilator, debuger, interpreter, javadoc
- Zasoby dostępne dla programisty: specyfikacja języka, dokumentacja API, tutoriale
- Kompilacja i uruchamianie programów
- Sposoby dystrybucji aplikacji Java SE

2. **Środowisko programisty Java**

- Intelij IDEA

- Narzędzie kontroli wersji (Git, ewentualnie Subversion)
- Budowanie aplikacji (Maven, opcjonalnie Gradle)

3. **Podstawy języka Java**

- Nazwy, obowiązujące konwencje
- Literały liczbowe i tekstowe
- Zmienne, deklaracja, inicjalizacja, przypisanie
- Instrukcje sterujące: warunkowe (if, switch), pętle (while, for), przerwania i etykiety
- Wyrażenia arytmetyczne i logiczne
- Typy proste, konwersja i rzutowanie
- Metody i parametry
- Metoda main i parametry wiersza poleceń
- Podstawowe sposoby komunikacji z otoczeniem i interakcji z użytkownikiem

**4. Programowanie obiektowe w Javie**

- Pojęcia obiektu i klasy
- Definiowanie klas w Javie, pola i metody instancyjne
- Konstruktory i bloki inicjalizacyjne
- Pola i metody statyczne
- Dziedziczenie i polimorfizm
- Nadpisywanie metod (override)
- Klasy abstrakcyjne, interfejsy, metody domyślne w interfejsach
- Przeciążanie metod (overload)
- Typy wyliczeniowe (enum)

**5. Podstawy projektowania obiektowego**

- Koncepcje związane z programowaniem obiektowym i ich znaczenie w Javie: Encapsulation, Coupling, Cohesion
- Diagram klas UML
- Zależności między klasami: agregacja, kompozycja, asocjacja, generalizacja (dziedziczenie)
- Podstawowe wzorce projektowe (Singleton, Fabryka, Metoda Szablonowa)

**6. Java SE API – biblioteka standardowa Javy – wprowadzenie**

- Najważniejsze pakiety
- Klasa Object, rola i sposoby nadpisywania wybranych metod
- Klasy opakowujące (wrappers)
- Klasa Math

**7. Wyjątki**

- Mechanika działania wyjątków, konstrukcja try-catch
- Klauzula finally i jej wykorzystywanie
- Podstawowe rodzaje wyjątków: Throwable, Exception, RuntimeException, Error; wyjątki wyłapywane i nie
- Klauzula try-with-resources, automatyczne zamykanie zasobów
- Przegląd wbudowanych wyjątków platformy Java
- Tworzenie własnych typów wyjątków

- Asercje

**8. Obsługa napisów w Javie**

- Klasa String i operacje na napisach
- Klasy StringBuilder i StringBuffer
- Podstawowa obsługa wyrażeń regularnych (Pattern, Matcher)
- Przetwarzanie tekstu fragment po fragmencie (String.split, StringTokenizer, Scanner)
- Formatowanie napisów, dat i liczb (klasa Formatter i podklasy)
- Ustawienia regionalne (Locale)
- Umiędzynarodowienie i lokalizacja aplikacji (ResourceBundle)

**9. Tablice, kolekcje i klasy generyczne**

- Tablice: deklaracja, utworzenie, tworzenie tablicy o podanej zawartości
- Tablice wielowymiarowe
- Klasa narzędziowa Arrays
- Kolekcje platformy Java (JCF): rodzaje, główne interfejsy, cechy wspólne i różnice
- Porównanie wydajności operacji na różnych kolekcjach
- Porównywanie obiektów: metody equals i hashCode, interfejsy Comparable i Comparator
- Klasa narzędziowa Collections
- Kolekcje specjalnego zastosowania oraz opakowania kolekcji: tylko do odczytu, synchronizowane itp.
- Kopiowanie obiektów, interfejs Cloneable
- Typy generyczne, działanie na przykładzie kolekcji, tworzenie własnych klas i metod generycznych

**10. Wejście / wyjście**

- Strumienie: wejściowe i wyjściowe, binarne i tekstowe
- Wybrane wyspecjalizowane klasy strumieni (w tym do obsługi plików i konsoli), opakowywanie strumieni
- Obsługa kodowania znaków w Javie
- Klasa Files i możliwości „NIO.2” (od Javy 7)
- Serializacja: strumienie obiektowe, możliwość zmiany sposobu serializacji własnych klas

**11. Podstawowa obsługa baz danych**

- JDBC: wspólne klasy i interfejsy, sterowniki, otwieranie połączenia
- Zapytania, zapytania sparametryzowane, obsługa wyników
- Transakcje

**12. Programowanie wielowątkowe**

- Wątek: co to jest, sposoby tworzenia (Runnable i Thread)
- Przeploty, priorytety, metody yield, join, sleep
- Podstawowa synchronizacja: metody i bloki synchronized, metody wait i notify/notifyAll
- Ryzyka programowania współbieżnego: wzajemne wykluczanie, zakleszczenie, zagłodzenie
- Gotowe mechanizmy synchronizacji z pakietu java.util.concurrent
- Pule wątków, mechanizm Fork/Join

**13. Podstawy programowania sieciowego**

- Podstawy protokołów IP / TCP / UDP
- UDP w Javie (DatagramSocket, Datagram Packet)

- TCP w Javie (Socket, ServerSocket)
- Przesyłanie danych tekstowych, binarnych i obiektów
- Możliwości współbieżnej obsługi wielu połączeń, select

**14. Programowanie funkcyjne w Javie**

- Klasy wewnętrzne i anonimowe
- Lambda-wyrażenia i referencje do metod (od Javy 8)
- Strumienie, zastosowanie dla kolekcji

**15. Wprowadzenie do technologii Spring**

- Spring Boot
- Spring MVC
- Spring Data

**16. Podstawy przetwarzania XML w Javie (DOM, SAX, StAX, JAXB, Transformer)**

**17. Podstawy automatycznego testowania aplikacji**

- Rodzaje testów, testy jednostkowe
- Biblioteka JUnit
- Koncepcje TDD i BDD
- Dodatkowe biblioteki: AssertJ oraz Mockito

**18. Przegląd zagadnień zaawansowanych**

- Debugowanie, logowanie
- Obszary pamięci JVM, ustawianie rozmiaru podczas uruchomienia
- Koniec życia obiektu (Garbage Collector, metoda finalize)
- Monitorowanie zasobów wirtualnej maszyny
- Ładowanie klas i Classloader
- Podstawy refleksji
- Integracja i zdalne wywoływanie procedur (RMI, web services)
- Metody natywne (JNI)

# Metoda realizacji szkolenia

Szkolenie realizowane jest w formie naprzemiennie następującej po sobie części teoretycznej w postaci mini wykładów oraz części praktycznej w postaci ćwiczeń komputerowych. Szkolenie łączy w sobie fachową wiedzę merytoryczną z praktycznymi przykładami jej wykorzystania w środowisku pracy. Ćwiczenia skonstruowane są w sposób, który wspiera utrwalenie nabytej wiedzy, oraz przyszłe twórcze wykorzystanie jej w dalszym rozwoju umiejętności.

# Liczba dni, liczba godzin szkoleniowych

5 dni, 40 godzin szkoleniowych

# Ścieżka rozwoju po szkoleniu

Udział w szkoleniu wspiera przygotowanie do certyfikacji OCPJP.