

Kod szkolenia: **J/JOB**

Tytuł szkolenia: **Programowanie w języku Java (przygotowanie do zawodu)**

Dni: 7

## Opis:

### Adresaci szkolenia

Szkolenie adresowane jest do osób rozpoczynających przygodę z programowaniem, pragnących poznać język Java.

### Cel szkolenia

Celem szkolenia jest nabycie umiejętności programowania w języku Java oraz zapoznanie się z platformą Java w zakresie umożliwiającym podjęcie pracy jako programista Java. Program szkolenia ułożony jest tak, by łagodnie wprowadzić uczestników w tajniki sztuki programowania z wykorzystaniem popularnych technologii i nowoczesnych narzędzi stosowanych przy prawdziwych projektach (bazy danych, IDE, testowanie, notacja UML).

Szkolenie przebiega w dwóch etapach. Pierwszy etap obejmuje podstawy języka Java oraz fundamenty programowania w ogóle, jak struktury danych, algorytmy i zasady programowania zorientowanego obiektowo.

Podczas drugiego etapu przedstawiane są zaawansowane zagadnienia języka Java, jak: programowanie aplikacji desktop przy użyciu biblioteki Swing lub JavaFX, programowanie sieci TCP/IP i aplikacji wielowątkowych. W tej części omawiane są również: Java Reflection API, JDBC oraz wprowadzenie do Java Enterprise Edition. Podczas warsztatów wykorzystywane są narzędzia: Eclipse, Maven, JUnit. Wymienione narzędzia mogą być zastąpione innymi, stosowanymi w Państwa firmie.

### Mocne strony szkolenia

Program prowadzi uczestników od fundamentów programowania aż po zaawansowane zagadnienia języka Java, a zarazem dostarcza podstaw niezbędnych do zrozumienia platformy Java Enterprise Edition (JEE).

Szkolenie kładzie nacisk na zagadnienia i umiejętności związane z tworzeniem kodu dobrej jakości, z użyciem zalecanych praktyk i wzorców.

Szkolenie jest ciągle uaktualniane do pojawiających się najnowszych wersji języka, przy zachowaniu informacji niezbędnych do utrzymywania programów tworzonych we

wcześniejszych wersjach.

## Wymagania

Od uczestników nie wymaga się wcześniejszej umiejętności programowania.

## Parametry szkolenia

7\*8 godzin (7\*7 godzin netto) wykładów i warsztatów (z wyraźną przewagą warsztatów) podzielone na 2 etapy (zajęcia odbywają się w dni robocze w trybie 5 + 2). W trakcie warsztatów, oprócz prostych ćwiczeń, projektowana i implementowana jest prosta aplikacja, obejmująca kluczowe zagadnienia poruszane na szkoleniu.

Wielkość grupy: maks. 8-12 osób.

## Program szkolenia:

---

### Część I (4 dni)

---

1. Wprowadzenie do programowania
  - a. Programowanie jako metoda rozwiązywania problemów
  - b. Podstawowe pojęcia (struktura danych, algorytm, program)
  - c. Dziedzina problemu i jej opisywanie
  - d. Rodzaje języków programowania
  - e. Cykl budowania programu (projektowanie, edycja kodu, kompilacja, uruchomienie, debugowanie)
  - f. Podstawowe informacje o języku Java
  - g. Kompilacja i uruchomienie pierwszego programu
2. Główne składowe języków programowania
  - a. Podstawowe typy danych (typy liczbowe, znakowe, logiczne, ciągi znaków)
  - b. Zmienne, deklaracja zmiennych typów prostych
  - c. Inicjalizacja zmiennych, instrukcje przypisania
  - d. Operatory arytmetyczne i logiczne
  - e. Konwersja i rzutowanie typów prostych
  - f. Typ tablicowy, zmienne tablicowe, operacje na tablicach
  - g. Bloki instrukcji, zasięg zmiennych
  - h. Metody, parametry metod, wartość zwracana
  - i. Metoda main i przekazywanie argumentów wywołania programu
  - j. Podstawowa obsługa standardowego wejścia/wyjścia
3. Instrukcje sterujące wykonaniem programu
  - a. Instrukcje warunkowe: if, else, switch
  - b. Pętle for, for-each, while, do-while

- c. Instrukcje break, continue, wykorzystanie etykiet
- 4. Praca z obiektami
  - a. Pojęcie klasy i obiektu
  - b. Deklaracja zmiennych typu obiektowego
  - c. Zmienne składowe obiektu i klasy (pola)
  - d. Metody obiektu i klasy
  - e. Zakres widoczności pól, metod i klas
  - f. Podział programu na pakiety
  - g. Konwencje obowiązujące w języku Java
- 5. Podstawy algorytmów i struktury danych
  - a. Pojęcie struktury danych i algorytmu
  - b. Sposoby zapisu algorytmów (schematy blokowe, pseudokod)
  - c. Podstawy szacowania złożoności algorytmów
  - d. Listy jednokierunkowe i dwukierunkowe, operacje na listach
  - e. Podstawowe algorytmy na listach (przeszukiwanie, odwracanie, element minimalny)
  - f. Przegląd algorytmów sortowania list
  - g. Algorytmy rekurencyjne
  - h. Metoda "dziel i zwyciężaj"
  - i. Zbiór, stos, kolejka
  - j. Drzewa, podstawowe operacje i algorytmy na drzewach
  - k. Grafy i algorytmy grafowe
    - l. Inne rodzaje algorytmów
- 6. Programowanie obiektowe
  - a. Deklaracja klas
  - b. Tworzenie konstruktorów i inicjalizacja obiektów
  - c. Składowe statyczne, import statyczny
  - d. Encapsulation, Coupling, Cohesion
  - e. Dziedziczenie
    - f. Dziedziczenie a zakres widoczności
    - g. Dziedziczenie a przypisania, operator instanceof
  - h. Używanie konstruktorów i metod nadklasy
    - i. Overload a override
    - j. Klasy abstrakcyjne, interfejsy, metody domyślne w interfejsach
  - k. Klasa Object
    - l. Typy wyliczeniowe
- 7. Obsługa wyjątków 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)
  - d. Klauzula try-with-resources, automatyczne zamykanie zasobów
  - e. Deklaracja metod zgłaszających wyjątki oraz zgłaszanie wyjątków (throws, throws)
  - f. Wyjątki a dziedziczenie
  - g. Przegląd wybranych wyjątków
  - h. Projektowanie własnych typów wyjątków

- i. Asercje
- 8. Wprowadzenie do projektowania
  - a. Podstawy UML
  - b. Rozważania projektowe: agregacja, kompozycja, asocjacja, generalizacja
  - c. Podstawowe wzorce projektowe (Singleton, DAO, Fabryka)
- 9. Klasa Math, Operacje na napisach i formatowanie
  - a. Metody Klasy Math
  - b. Klasa String, StringBuffer, StringBuilder
  - c. Operacje na napisach
  - d. Atomizacja napisów i podstawy wyrażeń regularnych (metoda split, klasa StringTokenizer, Scanner, Pattern)
  - e. Formatowanie napisów, dat i liczb (klasa Formatter)
  - f. Formatowanie z uwzględnieniem ustawień regionalnych
- 10. Kolekcje i klasy Generyczne
  - a. Rodzaje kolekcji w Java
  - b. Kolekcje uporządkowane, nieuporządkowane, posortowane, nieposortowane
  - c. Metoda equals i hashCode
  - d. Interfejsy Collection, Set, Map, List
  - e. Przegląd dostępnych kolekcji
  - f. Kolekcje synchronizowane
  - g. Iteracja po kolekcjach
  - h. Interfejsy Comparable i Comparator
  - i. Metody użytkowe z klasy Collections
  - j. Interfejs Clonable
  - k. Kolekcje generyczne
  - l. Zasady tworzenia klas generycznych
- 11. Wprowadzenie do platformy Java
  - a. Rodzaje instalacji
  - b. Narzędzia dostępne w instalacji JSE (kompilator, debugger, interpreter, generator dokumentacji i inne)
  - c. Dokumentacja
  - d. Sposoby dostarczania aplikacji Java SE (aplikacja klasyczna, aplet, java web start, plugin)
  - e. Zasoby internetowe

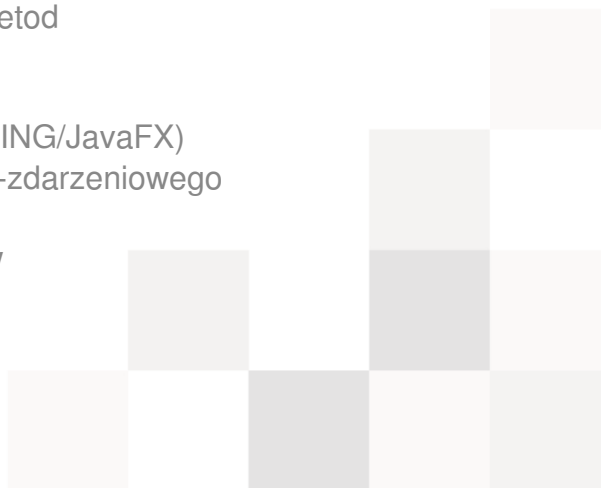
---

## Część II (3 dni)

---

- 12. 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. File I/O z NIO.2
  - i. Interfejs Serializable
  - j. Serializacja obiektów do strumieni (ObjectOutputStream, ObjectInputStream)
  - k. Modyfikacja domyślnej serializacji
    - l. Standardowe wejście/wyjście
  - m. Przykłady strumieni specjalizowanych
  - n. Atomizacja strumieni (StreamTokenizer)
- 13. Podstawy JDBC
  - a. O JDBC
  - b. Rejestracja sterownika
  - c. Wykonywanie zapytań
  - d. Transakcje
  - e. Projektowanie warstwy dostępu do danych
- 14. Zarządzanie zasobami JVM
  - a. Struktura pamięci
  - b. Metoda finalize i Garbage Collector
  - c. Monitorowanie zasobów wirtualnej maszyny
- 15. 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
- 16. Programowanie sieci TCP/IP
  - a. Podstawy protokołów IP / TCP / UDP
  - b. Programowanie z użyciem UDP (DatagramSocket, Datagram Packet)
  - c. Programowanie z użyciem TCP (Socket, ServerSocket)
  - d. Rodzaje i obsługa wyjątków
  - e. Zagadnienia przesyłania obiektów
  - f. Zagadnienia współbieżnej obsługi wielu połączeń
- 17. Elementy programowania funkcyjnego i metaprogramowania
  - a. Klasy wewnętrzne, statyczne, anonimowe
  - b. Lambda-wyrażenia i referencje do metod
  - c. Strumieniowe API dla kolekcji
  - d. Adnotacje i podstawy refleksji
- 18. Programowanie interfejsów graficznych (SWING/JavaFX)
  - a. Idea programowania komponentowo-zdarzeniowego
  - b. Pojęcie komponentu i kontenera
  - c. Przegląd komponentów i kontenerów
  - d. Zarządcy rozkładu



- e. Rodzaje zdarzeń i ich obsługa (Events, Listeners, Adapters)
  - f. Zagadnienia współbieżności w kontekście UI
  - g. Applety i Java2D, Java Web Start
19. Internacjonalizacja i lokalizacja aplikacji
20. Podstawy automatycznego testowania aplikacji
- a. Rodzaje testów
  - b. Biblioteki junit i testNG
  - c. Koncepcje TDD i BDD
  - d. Testowanie z użyciem obiektów Mock
21. Przegląd zaawansowanych zagadnień platformy Java
- a. Podstawy przetwarzania XML w Java
  - b. Classloader
  - c. Management & Logging
  - d. RMI
  - e. JNI
22. Wprowadzenie do Java Enterprise Edition

