

# Java Software Development

*Junior Backend Developer*

devmind™

# Programa de pregătire

## Introducere în programare **1**

1. Etapele dezvoltării software. Cod. Aplicații 1
2. Programare în Java 1
3. Tipuri de date. Operatori 1
4. Structuri de control. Fluxul de execuție 1

## Programare Orientată Obiect (POO) **2**

1. Concepte introductive 2
2. Proiectarea arhitecturii de clase 2
3. Abstractizare și interfețe. Practici de proiectare 2
4. Genericitate 3

## Colecții și Structuri de Date **3**

1. Tipuri predefinite și structuri de date de bază 3
2. Colecții și structuri de date 3

## Concepte Avansate (Introducere) **3**

1. Excepții 3
2. I/O. Fluxul de intrare - ieșire 4
3. Concurrency. Programare paralelă utilizând thread-uri 4
4. Introducere în SQL. JDBC. Lucrul cu baze de date în Java 4

## Pregătirea pentru Interviu **4**

1. Workshop de carieră – HR 4

# Introducere în programare

## 1. Etapele dezvoltării software. Cod. Aplicații

- Limbaje de programare. Principii, similarități și deosebiri
- Fluxul de la cerință la soluție
- Coding Style. Comentarii. Documentație
- Terminal. Linia de comandă
- Sisteme de versionare distribuită. Git

## 2. Programare în Java

- Ecosistemul Java
- Eclipse IDE
- JVM
- JDK, JRE
- javac, java, etc.
- Metoda specială main()
- Java documentation. Java API

## 3. Tipuri de date. Operatori

- Tipuri de date primitive. Tipul special void
- Identificatori. Variabile
- Operatori: unari, binari, ternari. Înlănțuire. Prioritate. Asociativitate
- Expresii matematice
- Tablouri n-dimensionale. Particularizare vectori și matrici.

## 4. Structuri de control. Fluxul de execuție

- Structuri de control condiționale
- Structuri de control iterative
- Declarații speciale: break, continue
- Bloc de instrucțiuni. Context local
- Imbricare în Java



# Programare Orientată Obiect (POO)

## 1. Concepte introductive

- Clase. Obiecte
- Câmpuri. Proprietăți
- Metode. Design, definire, apelare, semnătură, return
- Varargs
- Specificatori de acces: private, public, protected
- Încapsulare. „getter”, „setter” pentru manipularea membrilor
- Constructori. Ordinea inițializării
- Referințe. Cuvântul-cheie „this”. Obiect vs. referință
- Parametrii. „pass-by-value”, modificarea parametrilor
- Cuvântul-cheie „static”. Particularități. Static vs. Instance
- „final”. Proprietăți clase, metode, variabile. Immutable

## 2. Proiectarea arhitecturii de clase

- Compunere. Agregare.
- Moștenire (Inheritance). Cuvântul-cheie „super”
- Conversii de tip. Cast implicit vs. explicit. Upcasting. Downcasting
- Cuvântul-cheie „instanceof”
- Principii cheie în design – modularitate, reutilizare, izolare
- Polimorfism (Polymorphism). Compile time vs runtime
- Supraîncărcarea metodelor și constructorilor (method overloading)
- Parametrii polimorfici
- Suprascrierea metodelor (method overriding). Utilizare „super”
- Suprascrierea metodelor „toString()”, „equals()”

## 3. Abstractizare și interfețe. Practici de proiectare

- Clase abstracte
- Metode abstracte
- Interfețe. Implementarea și extinderea interfețelor
- Clase. Interne. Interne statice. Anonime. Locale
- Static binding vs. dynamic binding. Virtual methods
- Funcții Lambda. Sintaxă
- Cohesion & Coupling



## 4. Genericitate

- Clase generice. Type Erasure
- Metode generice
- Interfețe generice
- Bounds

# Colecții și Structuri de Date

## 1. Tipuri predefinite și structuri de date de bază

- Java Collections Framework
- Clasa String. Definirea și manipularea șirurilor de caractere
- StringBuilder. StringBuffer. Comparație
- Tablouri multidimensionale. Array. ArrayList
- Liste. Tipuri (simplu înlănțuite, dublu înlănțuite, circulare)
- Interfața List. LinkedList
- Parcurgerea colecțiilor. Iteratori. for-each
- Compararea elementelor. Comparable vs. Comparator
- Sortarea colecțiilor. Căutări
- Date & Time. Creare și manipulare

## 2. Colecții și structuri de date

- Stivă. Utilizare. Performanță. Clasa Stack
- Coadă. Utilizare. Performanță. Interfața: Queue. PriorityQueue
- Interfața Set. Colecții: TreeSet, HashSet, LinkedHashSet
- Interfața Map: Colecții: TreeMap, HashMap, LinkedHashMap
- HashTable. Funcția de hash. Tratarea coliziunilor. Performanță
- Aprofundare: scenarii comparative de aplicabilitate; diferențe de complexitate a operațiilor (add, remove, update, find)

# Concepte Avansate (Introducere)

## 1. Excepții

- Rolul și utilizarea excepțiilor
- Scrierea codului sigur. Declararea excepțiilor. „throws”



- Definirea și aruncarea excepțiilor. „throw”
- Manipularea excepțiilor - „try-catch-finally”
- Tipuri - runtime vs. checked vs errors
- Aserții (Assertions)

## 2. I/O. Fluxul de intrare - ieșire

- Fluxul de date în Java. Streams
- Citirea și scrierea datelor la consolă
- Citirea și scrierea datelor în fișiere (File I/O)
- Serializarea obiectelor. Interfața „Serializable”

## 3. Concurrency. Programare paralelă utilizând thread-uri

- Concepte de bază pentru calculul paralel
- Fire de execuție. Thread-uri
- Sincronizare. Locks. Deadlocks. Coruperea memoriei
- Design-ul unui algoritm paralel simplu
- Testarea programelor paralele

## 4. Introducere în SQL. JDBC. Lucrul cu baze de date în Java

- Definirea și utilizarea unei baze de date
- Limbajul SQL. Query-uri generale
- Funcții specifice. Manipularea datelor
- JDBC. Legarea aplicațiilor Java la o bază de date

# Pregătirea pentru Interviu

## 1. Workshop de carieră – HR

- CV-ul și backgrpund-ul anterior
- Profilul de LinkedIn
- Procesul de evaluare a unei companii
- Căutare unui job si etapele de aplicare
- Tratarea feedback-ului negativ
- Oferta de angajare



## Abilități și cunoștințe dobândite

1. Proiectează aplicații complexe, folosind principii fundamentale în dezvoltarea de software, precum: proiectarea arhitecturii de clase, reutilizarea codului și a funcționalității, izolare, etc.
2. Înțelege în profunzime concepte avansate de abstractizare și interfațare. Este capabil să proiecteze corect comunicarea între componente, expunerea funcționalității prin interfețe și înțelege balanța cohesion – coupling.
3. Se familiarizează cu utilizarea și funcționarea internă a setului principal de structuri de date folosite în programare. Înțelege complexitatea computațională a operațiilor caracteristice și diferențele de performanță.
4. Aplică strategii complexe de imbricare și grupare funcțională a mai multor tipuri de structuri de date pentru obținerea de soluții eficiente.
5. Se familiarizează cu conceptul de genericitate și este capabil să proiecteze clase, interfețe și metode generice, independente funcțional. Mai mult, înțelege mecanismul compilatorului Java în conversiile de tip, verificarea tipurilor și modul de legare.
6. Înțelege noțiunile teoretice introductive și avantajele de performanță oferite de calculul paralel. Redactează programe simple multi-thread și aplică corect mecanisme de sincronizare.
7. Asimilează principiile fundamentale și tipurile excepțiilor și le aplică pentru scrierea codului sigur, preventiv și robust.
8. Lucrează cu fluxul de date (intrare / ieșire) în Java pentru citirea și scrierea datelor.
9. Stăpânește noțiunile teoretice din programa modulului, fiind capabil să le folosească în scenarii variate și să le aplice pentru rezolvarea problemelor cu grad mediu și avansat de complexitate.
10. Prin învățare susținută de exemple, exerciții, proiecte și colaborare pe echipe, acumulează experiență „hands-on” de lucru efectiv. Mai mult, continuă să dezvolte o gândire critică, algoritmică, riguroasă, esențială în mediul IT și la interviuri.

## Observații

Cursul are ca obiectiv însușirea cunoștințelor și deprinderilor practice pentru a deveni un **Junior Backend Developer** complet, cu abilitățile necesare angajării pe o poziție Entry Level în cadrul unei companii de top.

Pe parcurs, vei dezvolta proiecte interesante cu tematici inspirate din realitate. Scopul lor este să integreze cunoștințele acumulate până la acel moment și să simuleze mediul de lucru real din Software Development. În cadrul lor, vei porni de la cerințe abstracte și vei realiza soluții complete, funcționale și optimizate, în conformitate cu principiile din industrie.

## Detalii desfășurare

<b>Durată</b>	8 luni, 32 săptămâni  705h total de pregătire, din care 200h de curs	Împărțit în 2 module Proces de admitere inițial Examen după fiecare modul Certificare acreditată emisă de Ministerul Educației și Ministerul Muncii la final
<b>Desfășurare</b>	25 ore / lună  Săptămânal: 2 zile, 18:30 - 21:00  Lunar: workshop 5h (1 zi de weekend)	