

## (Posibile) intrebari la interviuri

1. Care este diferenta dintre o clasa abstracta si o interfata?
2. O clasa abstracta se poate instantia? Dar o interfata?
3. Un obiect poate avea mai multe clase? Dar o clasa poate avea mai multe obiecte? Descrie care este diferenta dintre o clasa si un obiect.

R: O clasa este un blueprint/ template/ definitie pentru declararea si crearea de obiecte. Un obiect are attribute si functionalitati si el reprezinta instantierea unei clase care continue membrii si metode. Clasa este un tip de date OOP. Nu se aloca memorie in definirea clasei, ci doar in instantierea ei intr-un obiect.

4. Cum functioneaza coliziunile intr-un HashMap in Java? Cum sunt tratate in spate? Key si value ce tip de date pot primi?

R: <https://stackoverflow.com/questions/19691920/collision-resolution-in-java-hashmap>  
<https://javarevisited.blogspot.com/2016/01/how-does-java-hashmap-or-linkedhashmap-handles.html#axzz81xw7kpIP>

<https://www.youtube.com/watch?v=mvEE4JsPFWY>

5. Ai folosit generice? Cand si cum? Cand folosim Generics in Java, cu ce keyword ne putem referi ca un obiect T este un subtip al obiectului? Dar un supertip? (Extends si super keywords din Generics)
6. Este var un keyword in Java?

R: Nu este un keyword, este un nume de tip de date rezervat in Java.

<https://www.geeksforgeeks.org/var-keyword-in-java/>

<https://stackoverflow.com/questions/63073153/var-keyword-in-java>

7. Care este diferenta dintre Thread.run si Thread.start? Se creeaza un thread nou sau nu?
8. Ce este un deadlock? Cum putem obtine un deadlock in Java? (Producer & Consumer problem)
9. Ce este thread starvation si livelock? Cum putem obtine un thread starvation in Java si un livelock?
10. Cum putem seta mai multe 'pools' unui thread in Java ? (--> Executor Service)
11. Cum putem face un thread in Java?
12. Diferenta dintre Thread.fork() si Thread.join() in Java?
13. Cate tipuri de exceptii sunt in Java?
14. Care este diferenta dintre programare concurenta si programare reactiva?

R: <https://www.baeldung.com/spring-webflux-concurrency#:~:text=3.,concurrency%20with%20better%20resource%20utilization.>

15. De ce se cheama polimorfism static/dinamic? R: overriding si overloading

16. Ce structura de date foloseste cautare in  $O(\log n)$ ? R: Binary search tree, AVL tree
17. Cum putem sorta o lista in Java?
18. Doua obiecte egale au acelasi hashcode?
19. Diferenta dintre StringBuilder si StringBuffer
20. Diferenta dintre HashMap si TreeMap si Hashtable, ce este un hashmap? + complexitati

R: <https://www.geeksforgeeks.org/differences-between-hashmap-and-hashtable-in-java/>

<https://stackoverflow.com/questions/40471/what-are-the-differences-between-a-hashmap-and-a-hashtable-in-java>

21. Diferenta dintre HashSet si TreeSet si ce este un hashset? + complexitati
22. Diferenta dintre ArrayList si LinkedList, descriere + complexitati
23. Ce se intampla daca ai intr-un hashmap toate elementele cu un hashcode null, ce se intampla la inserare si care e complexitatea?
24. Cum functioneaza in spate hashmap-ul?
25. Ce este load factor si capacity?
26. Cum putem face o clasa imutabila?

R: <https://www.digitalocean.com/community/tutorials/how-to-create-immutable-class-in-java>

27. Cum se face o clasa singleton?
28. Ce design pattern putem avea pentru a crea o singura instanta daca apelam clasa cu mai multe thread-uri? Cum mai exact?
29. Cum se face un obiect thread-safe in Java?
30. Ce este un keyword in Java?
31. Ce este final keyword?
32. Ce este static keyword?
33. Descrie cele 4 concepte POO. R: abstraction, encapsulation, inheritance and polymorphism
34. Descrie SOLID Principles , cu exemple reale , in aplicatii Java

R: <https://www.baeldung.com/solid-principles>

35. Care este diferenta dintre & si && in Java?
36. Care e diferenta dintre static synchronized si synchronized?
37. Cum fac o lista imutabila in Java? R: Collections.unmodifiableList(list);
38. Descrie Design Pattern-ul Builder
39. Descrie Design Pattern-ul Facade
40. Care e diferenta dintre Abstract Factory Design Pattern si Factory Design Pattern? Descrie-le.  
ref: <https://www.baeldung.com/cs/factory-method-vs-factory-vs-abstract-factory>
41. Care sunt modifcatorii de acces in Java? Descrie
42. Ce face o clasa final in Java?

- 43. Intr-o clasa abstracta, putem avea metode neabstracte? Dar invers?
- 44. Descrie agregarea si compozitia
- 45. Cate clase publice putem avea intr-un fisier .java?
- 46. Care este lifecycle ul unui thread in Java? ref: <https://www.baeldung.com/java-thread-lifecycle>
- 47. In general de stream-uri, operatii intermediare si finale, exemple.
- 48. Ce este un pachet in Java? Ce este un modul in Java?
- 49. Despre Consumer, Supplier, Function si Predicate.
- 50. Cum putem face o lambda function in Java?
- 51. Avem nevoie neaparat de adnotarea de FunctionInterface pentru a face o interfata sa fie functionala?

R: Da. <https://stackoverflow.com/questions/50892117/why-to-use-functionalinterface-annotation-in-java-8>

- 52. Daca extindem o interfata, toate metodele trebuie sa fie implementate? Dar daca avem o clasa abstracta, trebuie toate metodele sa fie implementate?
- 53. Comparable vs Comparator
- 54. Intr-un enum putem avea metode abstracte? Daca da, cum si ce fac?
- 55. Pattern matching, variabila este 'final'.
- 56. Ce modificali de acces este implicit pentru metode in interfete?
- 57. In java cate clase putem extinde in mostenire? Dar cate putem implementa?

De obicei, la posturile de Java, nu se intreaba doar din Java, deci sa va asteptati si la intrebari de networking (retele de calculatoare), de sisteme de operare (ce este un proces zombie, thread vs proces, comenzile ls, cd, grep, systemctl, etc) si de sql (select-uri, join-uri, view-uri, cum se fac tabelele sau cum ati proiecta tabelele pe baza unor exemple de aplicatii si se mai intreaba de HTTP Requests (diferenta dintre POST vs PUT -> idempotent, ce face GET, PUT, POST, PATCH, DELETE)