

Објектно оријентисано програмирање

Питања за припрему испита – 2025/2026

Напомена: на испиту се проверава разумевање наведених тема, а не репродукција дефиниција. Очекује се да одговор буде поткрепљен примером кад год то има смисла, да тврдње буду образложене и примењене на конкретне и граничне случајеве. Питања могу бити формулисана тако да траже поређење, анализу или закључивање на основу наведених тема.

Решавање проблема помоћу рачунара

1. Шта су језички процесори и која је њихова улога?
2. Шта су асемблери? Описати процес асемблирања.
3. Шта су компајлери? Описати процес компајлирања.
4. Шта су интерпретери? Описати процес интерпретирања.
5. Упоредити процес компајлирања програма и интерпретације програма.
6. Шта се подразумева под процесом препроцесирања?

Објектно оријентисано програмирање

7. Која је основна идеја објектно оријентисане парадигме?
8. Које су предности, а које мане објектно оријентисаног програмирања?
9. Кратко описати историјат објектно оријентисаног програмирања.
10. Шта је класа, а шта је инстанца / конкретан објекат класе?
11. Објаснити механизам наслеђивања.
12. Шта подразумева концепт композиције код наслеђивања?
13. Упоредити динамичко и статичко везивање.
14. Шта се подразумева под учаурењем објекта?

Карактеристике програмског језика Јава

15. Навести и описати основне карактеристике програмског језика Јава.
16. Описати основне захтеве који су постављени приликом развоја програмског језика и окружења Јава.
17. Навести и описати различите типове Јава апликација.
18. Описати процес превођења и извршавања Јава програма.
19. Упоредити процес превођења изворног кода написаног у програмском језику Јава у извршни кôд и процес превођења изворног кода написаног у програмском језику С у извршни кôд.
20. Шта је Јава виртуелна машина? Укратко описати њену архитектуру и начин употребе.
21. Шта је JIT преводац и које су предности његове употребе?
22. Шта су Јава алати за развој (JDK)?
23. Шта је Јава API и за шта се користи?
24. На који начин се у Јави обезбеђује јединствени потпис елемената?
25. Шта су модули у Јави и која је предност њихове употребе?

Објектни тип

26. Описати објектни тип језика Јава.
27. Објаснити зашто је Јава строго типизиран језик.
28. Како се у меморији записују подаци објектног типа, а како подаци примитивног типа?
29. Шта је експлицитна конверзија типа?

Коришћење класа и објеката испоручених уз JDK

30. Шта је садржано у класи System и за шта се користи та класа?
31. Упоредити System.out и System.err. По чему су слични, по чему се разликују?
32. Приказ текста на конзоли, објекат System.out.

33. Приступ елементима Јава окружења, метод `System.lineSeparator()`.
34. Колико траје животни век објекта направљеног у програмском језику Јава? Упоредити оператор `new` из програмског језика Јава и функцију `malloc` из програмског језика C.
35. Да ли у Јави програмер може вршити експлицитно ослобађање меморије? Описати захтев за покретањем скупљача отпадака, метод `System.gc()`.
36. Креирање објекта (класа `Object`).
37. Суштинско поређење објеката. Објаснити како оператор поређења на једнакост `==` пореди променљиве примитивног типа, а како инстанчне променљиве.
38. Карактеристике ниски, имутабилност.
39. Креирање ниски.
40. Поређење ниски.
41. Ниске, коришћење метода класа `String` и `StringBuilder`.
42. Шта су омотачи примитивних типова и за шта се користе? Које су основне карактеристике омотача типова?
43. Класе-омотачи за примитивне типове, рад са објектима типа `Integer`, `Long`, `Character`, `Float`, `Double`.
44. Класа за математичке функције, класа `Math`, поља и методи.
45. Коришћење метода класе `Math` за рад са псеудослучајним бројевима.
46. Класа `Random`, рад са псеудослучајним бројевима.
47. Класе за рад са датумима и временима, мерење протеклог времена.

Низови у Јави

48. Декларација и иницијализација низа.
49. Која су основна својства низовног типа податка у програмском језику Јава?
50. Низовна променљива и индексна променљива.
51. Бројачки и колекцијски `for` циклус.
52. Низови низова, дводимензионални низ.
53. Тродимензионални низ и низови већих димензија.
54. Класа `Arrays`.
55. Аргументи командне линије код улазне тачке програма, `static` метод `main`.
56. Аргументи метода променљиве дужине. Препоруке за коришћење.

Класе, пакети, поља, методи и објекти у Јави

57. Детаљно објаснити основне карактеристике програмског језика Јава са становишта објектно оријентисане парадигме.
58. Класе у Јави. Креирање објекта - примерка дате класе.
59. Класе и објекти - поља.
60. Поље објекта. Приступ пољу.
61. Класна поља.
62. Опсег важења за променљиве и поља.
63. „Сакривање“ поља параметрима, референца `this`.
64. Који су најзначајнији методи класе `Object`? Зашто је ова класа важна у програмском језику Јава?
65. Дефиниција метода. Параметри метода, потпис метода, тело метода, повратна вредност метода.
66. Позив метода. Аргументи метода, константни параметри, супституција параметара при позиву метода.
67. Преоптерећење метода. Позивање другог преоптерећеног метода, референца `this`.
68. Класни методи.
69. Организација класа по пакетима. Дефинисање пакета, увоз класа из пакета.
70. Који су разлози за паковање класа у пакете? Навести неке од најчешће коришћених пакета програмског језика Јава.
71. Увоз класних метода.

72. Класе - наслеђивање.
73. Класе - наслеђивање. Тип објектне променљиве у времену извршавања.
74. Објаснити употребу оператора instanceof.
75. Како се врши конверзија између објекта основне и наслеђених класа?
76. Превазилажење поља и метода у подкласама. Објаснити превазилажење стандардних метода класе Object (toString(), equals(), hashCode()).
77. Приступ пољима надкласе у методима.
78. Позивање методе надкласе.
79. Објаснити својство полиморфизма програмског језика Јава.
80. Модификатори за контролу приступа пољима.
81. Модификатори за контролу видљивости метода.
82. Модификатор константности за поља.
83. Модификатор константности за аргументе метода.
84. Модификатор за ограничавање наслеђивања и превазилажења.
85. Иницијализациони блок.
86. Класни иницијализациони блок.
87. Шта су конструктори?
88. Шта су подразумевани конструктори?
89. Супституција параметара при позиву конструктора.
90. „Везивање“ поља и аргумената, копирајући конструктор.
91. Преоптерећење конструктора, референца this.
92. Позив конструктора надкласе, референца super.
93. По чему се конструктор разликује од осталих метода класе? У чему је разлика између конструктора, подразумеваног конструктора и копирајућег конструктора?

Напредни рад са класама и објектима

94. Дефинисање апстрактне класе.
95. Наслеђивање између апстрактних и конкретних класа.
96. Објаснити наслеђивање апстрактних класа.
97. Које су сличности, а које разлике између апстрактних класа и интерфејса?
98. Дефинисање интерфејса.
99. Проширивање интерфејса.
100. Имплементирање интерфејса од стране класа.
101. Параметри типа интерфејса.
102. Уређење у колекцији, интерфејс Comparable.
103. Уређење у колекцији, интерфејс Comparator.
104. Шта је омогућено имплементацијом интерфејса Comparable, а шта имплементацијом интерфејса Comparator?
105. Упоредити предности и недостатке употребе конструктора копије за прављење копије објекта са предностима и недостацима употребе механизма клонирања имплементацијом интерфејса Cloneable.
106. Интерфејси у ЈДК-у. Клонирање објекта, интерфејс Cloneable.

Угнеждене класе

107. Шта су угнеждене класе и које су предности употребе угнеждених класа?
108. Објаснити концепт статичке угнеждене класе. Којим елементима спољашње класе може да приступа статичка угнеждена класа?
109. Шта су нестатичке угнеждене класе?
110. Које су разлике између статичке угнеждене класе и унутрашње класе?
111. Шта су локалне унутрашње класе?
112. Када је корисно дефинисати унутрашњу класу, а када локалну унутрашњу класу?
113. Шта су анонимне класе и када се користе?

Изузеци и тврдње

- 114. Шта су изузеци у Јави и које су предности употребе изузетака?
- 115. Објаснити хијерархију изузетака у програмском језику Јава.
- 116. Објаснити ситуацију када долази до изузетка типа Error, изузетка типа RuntimeException, а када до изузетка типа Exception.
- 117. Избацивање изузетака.
- 118. Хватање (обрада) изузетака.
- 119. Пропагирање изузетака.
- 120. Објаснити ситуацију када се користи вишеструки catch блок. О чему посебно треба водити рачуна при употреби вишеструког catch блока?
- 121. Када се користи finally блок?
- 122. Шта су тврдње у Јави, како се реализују и када се користе?

Набројиви (енумерисани) тип

- 123. Карактеристике и дефинисање енумерисаног типа.
- 124. Претварање у ниску и добијање из ниске.
- 125. Додатни подаци придружени енумерисаном типу.
- 126. Упоредити руковање набројивим типовима пре и после увођења верзије Јава 5.

Генерички тип

- 127. Појам генеричког типа. Дефинисање генеричког типа. Генерички позив типа.
- 128. Појам генеричког типа. Генерички метод.
- 129. Појам генеричког типа. Ограничења за типове.
- 130. Генерици и виртуелна машина.
- 131. Генерици и наслеђивање.

Колекције и речници

- 132. Описати интерфејс Collection.
- 133. Колекције и итератори.
- 134. Описати интерфејс Iterator.
- 135. Интерфејс Iterable и колекцијски for циклус.
- 136. Објаснити везу између интерфејса Iterable и интерфејса Iterator.
- 137. Уређење у колекцији.
- 138. Методи класе Collections.
- 139. Шта су апстрактне колекцијске класе и која је њихова предност у односу на одговарајући интерфејс?
- 140. Интерфејс List.
- 141. Итератор листе, интерфејс ListIterator.
- 142. Повезана листа, класа LinkedList.
- 143. Интерфејс RandomAccess.
- 144. Низовна листа, класа ArrayList.
- 145. Интерфејс Set.
- 146. Сортирани скуп, интерфејс SortedSet.
- 147. Хеш-скуп, класа HashSet.
- 148. Како је имплементирана и како се користи класа HashSet? Зашто је битно да методи hashCode() и equals() буду конзистентно (ре)дефинисани?
- 149. Дрво-скуп, класа TreeSet.
- 150. Како је имплементирана и како се користи класа TreeSet? Када је погодније користити HashSet, а када TreeSet имплементацију скупа?
- 151. Интерфејс Queue.
- 152. Ред са два краја, интерфејс Deque.
- 153. Низовни ред са два краја, класа ArrayDeque.

- 154.** Реализација реда преко повезане листе и преко кружног низа (само описно).
- 155.** Интерфејс Map.
- 156.** Хеш-речник, класа HashMap.
- 157.** Дрво-речник, класа TreeMap.
- 158.** Које су разлике, а које сличности између колекције и речника? Како се речник може представити употребом колекција?

Улаз и излаз

- 159.** Које су класе изведене из апстрактне класе InputStream?
- 160.** Које су класе изведене из апстрактне класе OutputStream? Које од тих класа имају своје парњаке у хијерархији класа изведених из InputStream?
- 161.** Објаснити употребу метода класе Reader.
- 162.** Објаснити употребу метода класе Writer.
- 163.** Шта је уланчавање токова, како функционише и зашто се користи?
- 164.** Описати рад са датотекама употребом класе File.
- 165.** Објаснити како се класа Scanner може користити као алтернатива читачима и улазним токовима.