

# ОБЈЕКТНО ОРИЈЕНТИСАНО ПРОГРАМИРАЊЕ

Испитна питања - 2024/2025

## ОПШТЕ НАПОМЕНЕ

- Студент добија 3 питања и потом приступа писању концепта.
  - Концепт није обавезан, али је препоручен због лакоће прављења илустрација и ефикаснијег тока испитивања.
- Оцењује се комплетност и коректност показаног знања, што кроз концепт, што кроз усмену дискусију.
  - На сваком питању се мора показати довољан степен знања. Колико је довољно, процењује наставник (испитивач).
- Одговор је потребно илустровати на примеру кад год то има смисла.

## Решавање проблема помоћу рачунара

1. Језички процесори.
  - a. Шта су језички процесори и која је њихова улога?
  - b. Шта су асемблери? Описати процес асемблирања.
  - c. Шта су компајлери? Описати процес компајлирања.
  - d. Шта су интерпретери? Описати процес интерпретирања.
  - e. Упоредити процес компајлирања програма и интерпретације програма.
  - f. Шта се подразумева под процесом препроцесирања?

## Објектно оријентисано програмирање

2. Основни аспекти ООП.
  - a. Која је основна идеја објектно оријентисане парадигме?
  - b. Које су предности, а које мане објектно оријентисаног програмирања?
  - c. Кратко описати историјат објектно оријентисаног програмирања.
3. Класе, инстанце, наслеђивање, композиција, везивање и учауривање.
  - a. Шта је класа, а шта је инстанца/конкретан објекат класе?
  - b. Објаснити механизам наслеђивања.
  - c. Шта подразумева концепт композиције код наслеђивања?
  - d. Упоредити динамичко и статичко везивање.
  - e. Шта се подразумева под учаурењем објекта?

## Карактеристике програмског језика Јава

4. Основне карактеристике програмског језика Јава, захтеви, типови Јава апликација.
  - a. Навести и описати основне карактеристике програмског језика Јава.
  - b. Описати основне захтеве који су постављени приликом развоја програмског језика и окружења Јава.
  - c. Навести и описати различите типове Јава апликација.

5. Превођење и извршавање Јава програма.
  - a. Описати процес превођења и извршавања Јава програма.
  - b. Упоредити процес превођења изворног кода, који је написан у програмском језику Јава, у извршни код и процес превођења изворног кода написаног у програмском језику С у извршни код.
6. Јава виртуелна машина и ЈИТ.
  - a. Шта је Јава виртуелна машина? Укратко описати њену архитектуру и начин употребе.
  - b. Шта је ЈИТ преводац и које су предности његове употребе?
7. ЈДК, Јава API, модули.
  - a. Шта су Јава алати за развој (ЈДК)?
  - b. Шта је Јава API и за шта се користи?
  - c. На који начин се у Јави обезбеђује јединствени потпис елемената?
  - d. Шта су модули у Јави и која је предност њихове употребе?

## Језици и опис конструкција језика Јава

8. Објектни тип.
  - a. Описати објектни тип језика Јава.
  - b. Објаснити зашто је Јава строго типизиран језик.
  - c. Како се у меморији записују подаци објектног, а како примитивног типа?
  - d. Шта је експлицитна конверзија типа?

## Коришћење класа и објеката испоручених уз ЈДК

9. Класа System.
  - a. Шта је садржано у класи System и за шта се користи та класа?
  - b. Упоредити System.out и System.err. По чему су слични, по чему се разликују?
  - c. Приказ текста на конзоли, објекат System.out.
  - d. Приступ елементима Јава окружења, метод System.lineSeparator().
  - e. Колико траје животни век објекта направљеног у програмском језику Јава? Упоредити оператор new из програмског језика Јава и функцију malloc из програмског језика С.
  - f. Да ли у Јави програмер може вршити експлицитно ослобађање меморије? Описати захтев за покретањем скупљача отпадака, метод [System.gc\(\)](#).
10. Класа Object.
  - a. Креирање објекта.
  - b. Суштинско поређење објеката. Објаснити како оператор поређења на једнакост == пореди променљиве примитивног типа, а како инстанцне променљиве.
11. Класа String.
  - a. Карактеристике ниски, имутабилност.
  - b. Креирање ниски.
  - c. Поређење ниски.
  - d. Ниске, коришћење метода класе String и StringBuilder.

12. Класе омотачи примитивних типова.
- Шта су омотачи примитивних типова и за шта се користе? Које су основне карактеристике омотача типова.
  - Класе-омотачи за примитивне типове, рад са објектима типа Integer, Long, Character, Float, Double.
13. Класа Math и Random.
- Класа за математичке функције, класа Math, поља и методи.
  - Коришћење метода класе Math за рад са псеудо-случајним бројевима.
  - Класа Random, рад са псеудослучајним бројевима.
14. Класе за рад са датумима и временима, мерење протеклог времена.

## Низови у Јави

15. Низови уопштено.
- Декларација и иницијализација низа.
  - Која су основна својства низовног типа податка у програмском језику Јава?
  - Низовна променљива и индексна променљива.
  - Бројачки и колекцијски for циклус.
16. Вишедимензионални низови и класа Arrays.
- Низови низова, дводимензионални низ.
  - Тродимензионални низ и низови већих димензија.
  - Класа Arrays.
17. Аргументи командне линије и аргументи метода променљиве дужине.
- Аргументи командне линије код улазне тачке програма, static метода main.
  - Аргументи метода променљиве дужине. Препоруке за коришћење.

## Класе, пакети, поља, методи и објекти у Јави

18. Класе, објекти, поља.
- Детаљно објаснити основне карактеристике програмског језика Јава са становишта објектно оријентисане парадигме.
  - Класе у Јави. Креирање објекта – примерка дате класе.
  - Класе и објекти - поља.
  - Поље објекта. Приступ пољу.
  - Класна поља.
  - Опсег важења за променљиве и поља.
  - “Сакривање” поља параметрима, референца this.
19. Методи.
- Који су најзначајнији методи класе Object? Зашто је ова класа важна у програмском језику Јава?
  - Дефиниција метода. Параметри метода, потпис метода, тело метода, повратна вредност метода.

- c. Позив метода. Аргументи метода, константни параметри, супституција параметара при позиву метода.
  - d. Препоптерећење метода. Позивање другог препоптерећеног метода, референца `this`.
  - e. Класни методи.
20. Пакети.
- a. Организација класа по пакетима. Дефинисање пакета, увоз класа из пакета.
  - b. Који су разлози за паковање класа у пакете? Навести неке од најчешће коришћених пакета програмског језика Јава.
  - c. Увоз класних метода.
21. Наслеђивање и конверзија објеката.
- a. Класе – наслеђивање.
  - b. Класе – наслеђивање. Тип објектне променљиве у времену извршавања.
  - c. Објаснити употребу оператора `instanceof`.
  - d. Како се врши конверзија између објеката основне и наслеђених класа?
22. Превазилажење, приступ пољима и позивање метода надкласе.
- a. Превазилажење поља и метода у подкласама. Објаснити превазилажење стандардних метода класе `Object` (`toString()`, `equals()`, `hashCode()`).
  - b. Приступ пољима надкласе у методима.
  - c. Позивање методе надкласе.
  - d. Објаснити својство полиморфизма програмског језика Јава.
23. Модификатори.
- a. Модификатори за контролу приступа пољима.
  - b. Модификатори за контролу видљивости метода.
  - c. Модификатор константности за поља.
  - d. Модификатор константности за аргументе метода.
  - e. Модификатор за ограничавање наслеђивања и превазилажења.
24. Подешавање иницијалног стања објекта: иницијализациони блокови и конструктори.
- a. Иницијализациони блок.
  - b. Класни иницијализациони блок.
  - c. Шта су конструктори?
  - d. Шта су подразумевани конструктори?
  - e. Супституција параметара при позиву конструктора.
  - f. “Везивање” поља и аргумената, копирајући конструктор.
  - g. Препоптерећење конструктора, референца `this`.
  - h. Позив конструктора надкласе, референца `super`.
  - i. 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

25. Апстрактне класе.

- a. Дефинисање апстрактне класе.
- b. Наслеђивање између апстрактних и конкретних класа.
- c. Објаснити наслеђивање апстрактних класа.
- d. Које су сличности, а које разлике између апстрактних класа и интерфејса?

26. Интерфејси.

- a. Дефинисање интерфејса.
- b. Проширивање интерфејса.
- c. Имплементирање интерфејса од стране класа.
- d. Параметри типа интерфејса.

27. Преглед неких JDK интерфејса.

- a. Уређење у колекцији, интерфејс Comparable.
- b. Уређење у колекцији, интерфејс Comparator.
- c. Шта је омогућено имплементацијом интерфејса Comparable, а шта имплементацијом интерфејса Comparator?
- d. Упоредити предности и недостатке употребе конструктора копије за прављење копије објекта са предностима и недостацима употребе механизма клонирања имплементацијом интерфејса Cloneable.
- e. Интерфејси у JDK-у. Клонирање објеката, интерфејс Cloneable.

## Угнеждене класе

28. Угнеждене класе.

- a. Шта су угнеждене класе и које су предности употребе угнеждених класа?
- b. Објаснити концепт статичке угнеждене класе. Којим елементима спољашње класе може да приступа статичка угнеждена класа?
- c. Шта су нестатичке угнеждене класе?
- d. Које су разлике између статичке угнеждене класе и унутрашње класе?

29. Локалне унутрашње и анонимне класе.

- a. Шта су локалне унутрашње класе?
- b. Када је корисно дефинисати унутрашњу класу, а када локалну унутрашњу класу?
- c. Шта су анонимне класе и када се користе?

## Изузеци и тврдње

30. Изузеци, основни концепти, хијерархија изузетака.

- a. Шта су изузеци у Јави и које су предности употребе изузетака?
- b. Објаснити хијерархију изузетака у програмском језику Јава.
- c. Објаснити ситуацију када долази до изузетка типа Error, изузетка типа RuntimeException, а када до изузетка типа Exception.

31. Руковање изузецима.
- a. Избацивање изузетака.
  - b. Хватање (обрада) изузетака.
  - c. Пропагирање изузетака.
  - d. Објаснити ситуацију када се користи вишеструки catch блок. О чему посебно треба водити рачуна при употреби вишеструког catch блока?
  - e. Када се користи finally блок?
32. Шта су тврдње у Јави, како се реализују и када се користе?

## Набројиви (енумерисани) тип

33. Енумерисани тип.
- a. Карактеристике. Дефинисање.
  - b. Претварање у ниску и добијање из ниске.
  - c. Додатни подаци придружени енумерисаном типу.
  - d. Упоредити руковање набројивим типовима пре и после увођења верзије Јава 5.

## Генерички тип

34. Генерички тип, основни концепти.
- a. Појам генеричког типа. Дефинисање генеричког типа. Генерички позив типа.
  - b. Појам генеричког типа. Генерички метод.
35. Генерички тип, напреднији концепти.
- a. Појам генеричког типа. Ограничења за типове.
  - b. Генерици и виртуелна машина.
  - c. Генерици и наслеђивање.

## Колекције и речници

36. Колекције, основни концепти.
- a. Описати интерфејс Collection.
  - b. Колекције и итератори.
  - c. Описати интерфејс Iterator.
  - d. Интерфејс Iterable и колекцијски for циклус.
  - e. Објаснити везу између интерфејса Iterable и интерфејса Iterator.
  - f. Уређење у колекцији.
  - g. Методи класе Collections.
  - h. Шта су апстрактне колекцијске класе и која је њихова предност у односу на одговарајући интерфејс?

37. Листе.

- a. Интерфејс List.
- b. Итератор листе, интерфејс ListIterator.
- c. Повезана листа, класа LinkedList.
- d. Интерфејс RandomAccess.
- e. Низовна листа, класа ArrayList.

38. Скупови.

- a. Интерфејс Set.
- b. Сортирани скуп, интерфејс SortedSet.
- c. Хеш-скуп, класа HashSet.
- d. Како је имплементирана и како се користи класа HashSet? Зашто је битно да методи hashCode() и equals() буду конзистентно (ре)дефинисани?
- e. Дрво-скуп, класа TreeSet.
- f. Како је имплементирана и како се користи класа TreeSet? Када је погодније користи HashSet имплементацију, а када TreeSet имплементацију скупа.

39. Редови.

- a. Интерфејс Queue.
- b. Ред са два краја, интерфејс Deque.
- c. Низовни ред са два краја, класа ArrayDeque.
- d. Реализација реда пре повезане листе и преко кружног низа (само описно).

40. Речници, основни концепти.

- a. Интерфејс Map.
- b. Хеш-речник, класа HashMap.
- c. Дрво-речник, класа TreeMap.
- d. Које су разлике, а које су сличности између колекције и речника? Како се речник може представити употребом колекција?

## Улаз и излаз

41. Токови података.

- a. Које су класе изведене из апстрактне класе InputStream?
- b. Које су класе изведене из апстрактне класе OutputStream? Које од тих класа имају своје парњаке у хијерархији класа изведених из InputStream?
- c. Објаснити употребу метода класе Reader.
- d. Објаснити употребу метода класе Writer.
- e. Шта је уланчавање токова, како функционише и зашто се користи?

42. Рад са датотекама (класа File) и са парсерима (класа Scanner).

- a. Описати рад са датотекама употребом класе File.
- b. Објаснити како се класа Scanner може користи као алтернатива читачима и улазним токовима.