

# ОБЈЕКТНО ОРИЈЕНТИСАНО ПРОГРАМИРАЊЕ

*Испитна питања - 2022/2023*

## 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. Водопадни и спирални модел.
  - a. Описати поступак решавања проблема помоћу рачунара употребом водопадног модела развоја софтвера.
  - b. Описати поступак решавања проблема помоћу рачунара употребом спиралног модела развоја софтвера.
  - c. Упоредити водопадни и спирални модел развоја софтвера. По чему су слични, по чему се разликују?
2. 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

3. 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.
4. 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?

## Неке програмске парадигме

5. Програмске парадигме.
- a. Шта је програмска парадигма? Навести и описати неке од основних програмских парадигми.
  - b. Описати императивну програмску парадигму и њене потпарадигме.
  - c. Описати декларативну програмску парадигму и њене потпарадигме.
  - d. Упоредити императивну и декларативну програмску парадигму.

## Карактеристике програмског језика Јава

6. Развој програмског језика Јава, основне карактеристике, захтеви, типови Јава апликација.
- a. Кратко описати настанак и развој програмског језика Јава.
  - b. Навести и описати основне карактеристике програмског језика Јава.
  - c. Описати основне захтеве који су постављени приликом развоја програмског језика и окружења Јава.
  - d. Навести и описати различите типове Јава апликација.
7. Превођење и извршавање Јава програма.
- a. Описати процес извршавања Јава програма.
  - b. Описати процес превођења и извршавања Јава програма.
  - c. Упоредити процес превођења изворног кода, који је написан у програмском језику Јава, у извршни кôд и процес превођења изворног кода написаног у програмском језику С у извршни кôд.
8. Јава виртуелна машина и ЈИТ.
- a. Шта је Јава виртуелна машина? Укратко описати њену архитектуру и начин употребе.
  - b. Шта је ЈИТ преводилац и које су предности његове употребе?

9. JDK, Јава API, модули.
  - a. Шта су Јава алати за развој (JDK)?
  - 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

10. Опис језика и конструкција Јаве, општи елементи језика (не треба учити Бекусову нотацију за појединачне конструкције).
  - a. Језици и опис конструкција језика Јава - граматика, синтакса и семантика.
  - b. Језици и опис конструкција језика Јава - Бекусова нотација.
  - c. Описати основне конструкције језика Јава - идентификаторе и литерале.
  - d. Описати основне конструкције језика Јава - операторе и изразе.
  - e. Описати основне конструкције језика Јава - кључне речи, коментаре и сепараторе.
11. Примитивни типови података у Јави.
  - a. Описати целобројне примитивне типове језика Јава.
  - b. Описати реалне примитивне типове језика Јава.
  - c. Описати знаковне примитивне типове језика Јава.
  - d. Описати логички примитивни тип језика Јава.
12. Објектни тип.
  - a. Описати објектни тип језика Јава.
  - b. Објаснити зашто је Јава строго типизиран језик.
  - c. Како се у меморији записују подаци објектног типа, а како подаци примитивног типа?
  - d. Шта је експлицитна конверзија типа?
13. Променљиве, наредбе, изрази.
  - a. Променљиве, декларација и иницијализација вредности, опсег важења.
  - b. Наредбе, обележена наредба.
  - c. Наредбе, празна наредба.
  - d. Изрази, оператори: арност, асоцијативност, приоритет.
  - e. Наредбе гранања, наредба if.
  - f. Наредбе гранања, наредба switch и наредба break.
  - g. Наредбе циклуса, наредба while, наредба do-while, наредба for бројачки циклус.
  - h. Наредбе циклуса, наредба break, наредба continue.

## Коришћење класа и објеката испоручених уз JDK

### 14. 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 C.
- f. Да ли у Јави програмер може вршити експлицитно ослобађање меморије? Описати захтев за покретањем скупљача отпадака, метод System.gc().

### 15. Класа Object.

- a. Креирање објекта.
- b. Суштинско поређење објеката. Такође, објаснити како оператор поређења на једнакост == пореди променљиве примитивног типа, а како инстанцне променљиве.

### 16. Класа String.

- a. Карактеристике ниски, имутабилност.
- b. Креирање ниски.
- c. Поређење ниски.
- d. Ниске, коришћење метода класе String и StringBuilder.

### 17. Класе омотачи примитивних типова.

- a. Шта су омотачи примитивних типова и за шта се користе? Које су основне карактеристике омотача типова.
- b. Класе-омотачи за примитивне типове, рад са објектима типа Integer, Long, Character, Float, Double.

### 18. Класа Math и Random.

- a. Класа за математичке функције, класа Math, поља и методи.
- b. Коришћење метода класе Math за рад са псеудо-случајним бројевима.

### 19. Класа Random, рад са псеудослучајним бројевима.

### 20. Класе за рад са датумима и временима, мерење протеклог времена.

## Низови у Јави

### 21. Низови уопштено.

- a. Декларација и иницијализација низа.

- b. Која су основна својства низовног типа податка у програмском језику Јава?
  - c. Низовна променљива и индексна променљива.
  - d. Бројачки и колекцијски for циклус.
22. Вишедимензионални низови и класа Arrays.
- a. Низови низова, дводимензионални низ.
  - b. Тродимензионални низ и низови већих димензија.
  - c. Класа Arrays.
23. Аргументи командне линије и аргументи метода променљиве дужине.
- a. Аргументи командне линије код улазне тачке програма, static метода main.
  - b. Аргументи метода променљиве дужине.
  - c. Аргументи метода променљиве дужине, препоруке за коришћење.

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

24. Класе, објекти, поља.
- a. Детаљно објаснити основне карактеристике програмског језика Јава са становишта објектно оријентисане парадигме.
  - b. Класе у Јави. Креирање објекта – примерка дате класе.
  - c. Класе и објекти - поља.
  - d. Поље објекта. Приступ пољу.
  - e. Класна поља.
  - f. Опсег важења за променљиве и поља.
  - g. “Сакривање” поља параметрима, референца this.
25. Методи.
- a. Који су најзначајнији методи класе Object? Зашто је ова класа важна у програмском језику Јава?
  - b. 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.
26. 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.
27. 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?
28. 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.
29. 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.
30. 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

31. 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?

32. Интерфејси.

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

33. 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 објекта, интерфејс Cloneable.

34. SOLID принципи и препоруке за наслеђивање. .

- a. Принцип једнозначне одговорности.
- b. Принцип отворености и затворености.
- c. Принцип замене Лисков.
- d. Принцип раздвајања интерфејса.
- e. Принцип инверзије зависности.
- f. Препоруке за наслеђивање.

## Угнеждене класе

35. 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?

36. 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

37. 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`.
38. 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?
39. Шта су тврдње у Јави, како се реализују и када се користе?

## Набројиви (енумерисани) тип

40. 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

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

## Колекције и речници

43. 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?
44. Листе.
- a. Интерфејс List.
  - b. Итератор листе, интерфејс ListIterator.
  - c. Повезана листа, класа LinkedList.
  - d. Интерфејс RandomAccess.
  - e. Низовна листа, класа ArrayList.
45. 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.
46. 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).
47. 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?
48. Колекције, остали концепти.
- a. Генерици и колекције.
  - b. Шта је џокер тип код колекција и када се користи? Објаснити употребу џокер типа

- горње границе и џокер типа доње границе.
- c. Ограничења над џокер типом код колекција.
- d. Генеричке методе са ограничењима џокер типа имплементиране у JDK-у.

## Улаз и излаз

- 49. 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?
- 50. 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.