**1) რა არის JVM?**

* JVM, ანუ ჯავა ვირტუალური მანქანა.ის არის ე.წ ძრავა რომელიც მართავს Java-ს კოდს.ის ჩვენს კოდს გარდაქმნის Bytecode-ად,რომლის საშუალებითაც ის უკავშირდება მასპინძელ სისტემას, რათა ასე ვთქვათ ოპერავიული სისტემისთვის გასაგებ ენაზე იყოს გადათარგმნილი კოდი.

**2)რა არის JDK?**JDK არის java პროგრამის სრულად შემუშავებული ნაკრები, რათა მომხმარებელმა შეძლოს java პროგრამების შედგენა და შესრულება.

ძირითადად, უზრუნველყოფს გარემოს შექმნას Java- ზე დაფუძნებული ყველა აპლიკაციისთვის.

**3) რა არის პრიმიტივი?**

მონაცემების ყველაზე ძირითადი ტიპებია, რომლებიც ჯავას ენაზეა შესაძლებელი.

*მონაცემთა პრიმიტიული ტიპები წარმოადგენს წინასწარ განსაზღვრული ტიპის ტიპებს, რომელთა მხარდაჭერა ხდება პროგრამირების ენაზე. მათ იყენებენ პროგრამისტების მიერ ახალი ცვლადის შექმნის დროს.*

*ex: - char, int, float, boolean*

**4)რა განსხვავებაა პრიმიტივსა და ობიექტს შორის?**

**პრიმიტივები** წარმოადგენენ მონაცემთა დამოუკიდებელ ტიპებს, ანუ მათთვის **არ არსებობს** იერარქია / მშობელი კლასი.ამასთანავე, პრიმიტივებს აქვთ ფიქსირებული გაომოყოფილი მეხსიერება,- ნებისმიერი int ყოველთვის 32 ბიტი იქნება, შესაბამისად მათი გაოყენება უფრო მართებულია სისწრაფისთვის.

 ყველა ობიექტი არის კლასის "ობიექტის" ობიექტი იკავებს ძალიან დიდ ადგილს რადგან მას თან მოჰყვება დიდი მოცულობის დატა მეთოდებთან ერთად.

**5)რატომ ეძახიან JAVA-ს მულტიპლატფორმულ პროგრამირების ენას?**

ჯავა არის მულტიპლათფორმული პროგრამირების ენა იმ გაგებით, რომ  ჯავა პროგრამა მუშაობს ყველა პლატფორმაზე, რისთვისაც არსებობს JVM. (ეს მოიცავს ყველა მნიშვნელოვან ოპერაციულ სისტემას, მათ შორის Windows- ს, Mac OS- ს და Linux- ს.)

**6) რა არის ნაკადი?**

ნაკადი არის საშუალება, რომელიც უზრუნველყოფს ერთი პროცესის განმავლობაში მრავალჯერადი საქმიანობის თანაარსებობას(შესრულებას).

**7) რამდენი ხერხით შეიძლება ნაკადის შექმნა?**

Threads can be created by using tწო mechanisms :

1. Extending the Thread class

2. Implementing the Runnable Interface

**8) აღწერეთ ნაკადის სასიცოცხლო ციკლი.**

**New:**იქმნება ახალი ნაკადი

**Runnable:** ნაკადი მზადაა კოდის გასაშვებად.

**Running :** პროცესორი უშვებს კოდს ნაკადზე.

ამის შემდეგ კოდი ან მთავრდება(**Dead)** ან გადადის შედეგ ციკლზე**(Waiting):**

**Waiting :** ნაკადი ელოდება(გადადის blocked state waiting-ში), როდის დამთავრდება, სხვა გაშვებული პროცესი და შემდეგ მთავრდება.(Dead)

**9) რას ეწოდება ინკაფსულაცია?**

ენკაფსულაცია საშუალებას გვაძლევს შექმნილი ობიქტები, პრიმიტივები, კლასი..... გავხადოთ წვდომადი ყველასთვის(public),არ იყოს წვდომადი ყველასთვის არამედ მხოლოდ იმ კლასისთვის სადაც ვართ(private), გავხადოთ წვდომადი შვილობილი კლასებისთვის და თვითონ კლასისთვის (protected), ან default-ის შემთხვევაში გავხადოთ წვდომადი კლასში და package-ში.(ის ვერ ხედავს სხვა package-ებს გარდა მიმდინარე package-სი)

**10)რას ეწოდება მემკვიდრეობითობა?**

მემკვიდრეობა არის მექანიზმი, რომლის დროსაც ახალი კლასი გამომდინარეობს არსებული კლასიდან. ჯავაში კლასებმა შეიძლება მემკვიდრეობით მიიღონ ან შეიძინონ სხვა კლასების თვისებები და მეთოდები.

სხვა კლასიდან მიღებული კლასს ეწოდება ქვეკლასი(subclass), ხოლო კლასს, საიდანაც ქვეკლასია წარმოქმნილი, სუპერკლასს(supercluss) უწოდებენ. ქვეკლასს შეიძლება ჰქონდეს მხოლოდ ერთი სუპერკლასი(იგივე მშობელი კლასი), ხოლო სუპერკლასს შეიძლება ჰქონდეს ერთი ან მეტი ქვეკლასი.

**11)რას ეწოდება პოლიმორფიზი?**

**პოლიმორფიზმი** არის ობიექტზე ორიენტირებული პროგრამირების კონცეფცია, რომელიც გულისხმობს ცვლადის, მეთოდის ან ობიექტის მრავალჯერადი ფორმირების უნარს.

ამის ერთ-ერთი მაგალითია overloading: როდესაც არსებობს მრავალი მეთოდი იმავე სახელწოდებით, მაგრამ სხვადასხვა პარამეტრით.ამ დროს მეთოდებს შეიძლება შევუცვალოთ **არგუმენტების რაოდენობის** ან / და **არგუმენტების ტიპი.**.

**12) რას ეწოდება სტეკი?**

შეიძლება ჩავთვალოთ როგორც ოპერაციული მეხსიერების ნაწილი, რომელშიც ლოკალური პრიმიტიული მნიშვნელობები და ლოკალური ობიექტების მიმთითებლები ინახება

**13) რა არის ჰიპი?**

Java Heap სივრცე გამოიყენება java runtime მიერ ობიექტების განსათავსებლად. როდესაც ჩვენ შევქმნით რაიმე ობიექტს, ის ყოველთვის იქმნება Heap სივრცეში.იგი ინახავს ობიექტებს მათი მთლიანი შიგთავსით. ანუ რაღაც ობიექტს რომ ჰქონდეს შიგნით პრიმიტივის ცვლადი, ეგეც ამ ობიექტთან ერთად იქნება შენახული და არა სტეკში.

**14) რა არის Garbage Collection?**

ეს არის პროგრამა რომელიც ავტომატურად მართავს მეხსიერებას.მხოლოდ იმ ობიექტებს შლის, რომლებსაც უკვე ვეღარ მივწვდებით (მიმთითებლები აღარ ჰყავთ)

.

**15) რას ეწოდება კონსტრუქტორი?**

კონსტრუქტორი ჯავაში არის **სპეციალური მეთოდი,** რომელიც გამოიყენება ობიექტების ინიციალიზაციისთვის. კონსტრუქტორი ეწოდება, როდესაც იქმნება კლასის ობიექტი. მისი გამოყენება შესაძლებელია საწყისი მნიშვნელობების ინიციალიზირებისთვის ობიექტის ატრიბუტების საშუალბით.

კლასს და კონსტრუქტორს ერთი და იგივე სახელი აქვთ.

**16) რა განსხვავება კლასის მეთოდსა და კლასის წევრს შორის?**

Class member, ანუ კლასის წევრი ზოგადად მოისაზრება როგორც კლასის რაღაც გარკვეული ნაწილი, იქნება ეს კლასის ცვლადი, მეთოდი, თუ სხვა რამ ხოლო class method არის კონკრეტულად კლასის მეთოდი

**17) რას ეწოდება Package?**

**პაკეტი** არის იმ სახელთა სივრცე, რომელიც აკავშირებს კლასებსა და ინტერფეისებს ერთმანეთთან.

პაკეტების გამოყენებით, უფრო ადვილია წვდომის კონტროლის უზრუნველყოფა და ასევე ადვილია დაკავშირებული კლასების განთავსება.ასევე, გამოიყენება იმისთვის სახელების დარქმევის პრობლემა რომ აირიდო თავიდან.

**18) რა განსხვავებაა final კლასსა და abstract კლასს შორის?**

Final ტიპის კლასს აკრძალული აქვს მემკვიდრეობითობა, ანუ მისი მემკვიდრის გაკეთება არ შეგვიძლია.

აბსტრაქტული კლასი არის class-ი და არა subclass-ი. ანუ, მათი გამოყენება (კლასის ინსტანციების/ობიექტების) შექმნა შეუძლებელია, მაგრამ მათ საფუძველზე შეგიძლია შექმნა შვილობილი კლასები

**19) რას ეწოდება გამონაკლისი შემთხვევა?**

შემთხვევა, რომელიც ხდება პროგრამის ან პროგრამის შესრულების შემდეგ..ანორმალური შემთხვევა, რომლის გამოსწორება შეგვიძლია.

**20) რამდენი ხერხით შეიძლება გამონაკლისი შემთხვევის დამუშავება?**

Checked exceptions: compile time exceptions

Unchecked exceptions: Runtime Exceptions

პრაქტიკული იმპლემენტაციები - try/catch/finally-ი და throws-ი

**21) რას ეწოდება ინტერფეისი?**

ინტერფეისი არის აბსტრაქტული ტიპი, რომელიც გამოიყენება იმ მოქმედების დასადგენად, რომელებიც კლასებმა უნდა შეასრულონ. ისინი პროტოკოლების მსგავსია. ინტერფეისები გამოცხადებულია ინტერფეისის საკვანძო სიტყვის გამოყენებით.(implements)

**22) რა განსხვავებაა ინტერფეისსა და აბსტრაქტულ კლასს შორის?**

**აბსტრაქტულ კლასს** შეიძლება ჰქონდეს მეთოდები რომლებიც ახორციელებენ რაიმე მოქმედებას.აბსტრაქტული კლასი არის კლასი, ანუ შვილობილი კლასი აუცილებლად მისი შვილი უნდა იყო

**ინტერფეისში** გამოცხადებული ცვლადები არის საბოლოო და მისი გადასაზღვრა არ არის შესაძლებელი, ანუ არ შეგვიძლია მისი შეცვლა ხოლო აბსტრაქტულ კლასში გამოცხადებული ცვლადების შეცვლა კი შეგვიძლია.

**23) რა განსხვავება public და protected მოდიფიკატორებით განსაზღვრულ კლასის წევრებს შორის?**

**Public** წევრები წვდომადია ყველა კლასისთვის.public სხვა package-ებთან წვდომადია

**protected** წვდომადია იმავე კლასისთვის და შვილობილებისთვის... შეიძლება რამდენიმე შვილობილი ჰყავდეს კლასს

**24) რა განსხვავება public მოდიფიკატორსა და მოდიფიკატორის გარეშე განსაზღვრულ კლასის წევრს შორის?**

**Public** წევრები წვდომადია ყველა კლასისთვის.(მათ შორის სხვა package-ში) **Default** წვდომადია მხოლოდ არსებულ package-ში.

**25) რა განსხვავება სტეკსა და ჰიპს შორის?**

**პიპში** ვარდება შექმენილი ობიექტები.იგი ინახავს ობიექტებს მათი მთლიანი შიგთავსით. ანუ რაღაც ობიექტს რომ ჰქონდეს შიგნით პრიმიტივის ცვლადი, ეგეც ამ ობიექტთან ერთად იქნება შენახული და არა სტეკში.

**სტეკში** ვარდება ლოკალური პრიმიტივები და არა ობიექტების პრიმიტივები.

**26) რა განსხვებაა FileWriter და BufferedWriter კლასების გამოყენებას შორის?**

**FileWriter-** ამ დროს ფაილში ინფორმაცია იწერება თითო სიმბოლოს მიხედვით.

**BufferedWriter** - ამ დროს ინფორმაცია ფაილში იწერება სტრიქონებად.

BufferedWriter აგროვებს ბევრ პატარ-პატარა მონაცემს ერთდროულად, შეამკობს, და შემდეგ უგზავნის FileWriter-ს

**27) რას ეწოდება Thread sheduler?**

**JVM-**ის ნაწილი რომელიც წყვეტს რომელა ნაკადი უნდა გაეშვას.

**28) რა საფრთხე არსებობს ნაკადებთან მუშაობის დროს?**

**პრობლემა იქმნება როცა საერთო რესურსებთან გაქვს წვდომა.**

**Deadlock** -როცა პროგრამა ეკიდება ანუ გაჭედა.