GRUPA PITANJA 1

1. Pod kojim nazivom mora biti snimljena datoteka sa izvornim Java kodom?

Mora biti snimljena sa ekstenzijom .java

2. U kojim slučajevima se javljaju sintaksne greške u kodu?

Sintaksne greske se javljaju pri prevodjenju izvornog koda, I one prestavljaju gramaticke greske pri radu sa nekim programskim jezikom.

3. Šta su logičke greške u programu?

Ako je program ispravno sintaksno napisan I ispravno preveden, ne mora da znaci da ce radi jer se mogu javiti neocekivane logicke greske koje sejavljaju u radu programa i detektuju se I resavaju tek pokretanjem programa I njegovim testiranjem.

5. Šta je definisano programskim blokovima?

U programskim blokovima definisana je grupa instrukcija,metoda I podataka. Pocinju I zatvaraju se viticastom zagradom.

6. Kako se prevodi i pokreće Java program sa komandne linije?

Kada se izvrsava Java program, JVM prvo uzima bajtkod klase u memoriji, upotrebom programa koji se naziva class loader. Posle preuzimanja klase, JVM upotrebljava program koji se naziva verifiktorom bajtkoda radi provere ispravnosti bajtkoda. Prevodi se kompajlerom koji se poziva sa javac I onda prevodi izvorni u bajtkod sa komandom java. Pokrece se takodje komandom java.

7. Koje su prednosti programiranja primenom integrisanog razvojnog okruženja IDE?

1)Mogucnost upotrebe skracenica, precica, menija ,hintova(pomoc),podvlacenje sintaksno neispravnog koda ; intelliSense mehanizma(kucanjem koda pritiskom kombinacije taster ctr+space IDE predlozi dopunu koda-to je zapravo automatsko dopunjivanje koda)

2)Pretrazivanje koda I njegovo refaktorisanje

3)Kompajliranje I izvrsavanje koda, s tim da je integrisan kompajler

4)Wizards koji olaksavaju komplikovane zadatke

5)Kreiranje izvrsnih,instalacionih paketa

8. Koja su tri najčešće korišćena menija NetBeans IDE? Objasnite ih ukratko?

1)File - je namenjen za formiranje novih i otvaranje postojecih projekata I fajlova

2)Run - je namenjen za izvrsavanje prevedenog programa; u ovom meniju moze se pokrenuti I debugger koji sluzi za otklanjanje I testiranje gresaka

3)Edit - ima potpuno istu funkciju za rad sa tekstom programa kao istoimeni meniji u ostalim grafickim okruzenjma (npr: Undo, Redo, Copy, Paste, Find,Replace)

9. Koje NetBeans panele koristimo prilikom razvoja Java programa? Objasnite ih ukratko.

1)Glavni panel za kod(radni panel); 2)levo od toga je panel projekata I panel navigatora; 3)na dnu je izlazni panel(output). Ukoliko neki od panela nije trenutno prikazan, moze se aktivirati iz menija Window.

10. Kako se kreira, prevodi i pokreće Java program primenom integrisanog razvojnog okruženja NetBeans IDE?

Kreira se tako sto se prvo se napravi novi projekat(File-new project; izabere se kategorija Java I opcija Java application nakon cega se odredjuje ime projekta I njegova lokacija), zatim paket I onda fajl. Prevodi pokretanjem kad se pritisne Run File(precica shift+F6). Prevedeni bajtkod se nalazi pod ekstenzijom .class.

11. Šta je Java?

Java je ujedno I programski objektno-orijentisan jezik viseg nivoa I platforma(predstavlja posebno okruzenje koje pokrece aplikacije napravljene Javi: JAVA SE, JAVA EE, JAVA ME). Svaka Java platfroma sadrzi virtuelnu masinu(program za posebne hardverske i softverske platforme koji pokrece aplikacije izradjene u Javi) i programski interfejs aplikacija(predstavlja skup softverskih komponenti koje se moze koristiti za razvoj drugih softverskih komponenti ili aplikacija).

12. Koja poboljšanja uvodi Java u odnosu na C/C++?

Dostupna je na svim operativnim sistemima,tj. ne zavisi od masine na kojoj se upotrebljava; u sintaksi nema datoteka zaglavlja, pokazivaca, struktura, unija, preklapanja operatora, virtuelnih osnovnih klasa I radi lakse(nema destruktore); ne mora da se vodi racuna u memorijskom prostoru; mehanizam refleksije, kao i serijalizacija objekata, omogucavaju mnogo laksu implementaciju perzistentnih objekata, tj. objekata koji su trajno memorisano u sekundarnoj memoriji racunara; velicine primitivnih tipova podataka su unapred definisane, kao i ponasanje aritmetike koja ih koristi(ceo broj u Javi uvek je 32-bitni dok je u C/C++ 16-bitni ili 32-bitni).

13. Šta je JRE? Šta je JDK?

JRE – Java Runtime Enviroment(omogucava da se na kompjuteru pokrece Java program tj. koristi se da obezbedi runtime okruzenje.; to je implementacija JVM; sadrzi virtuelnu masinu I skup biblioteka I druge fajlove koje JVM koristi u vremenu izvrsavanja (runtime)); JDK- Java Development Kit(omogucava da se pogram pravi pokrece-sadrzi JRE tj. sadrzi za pokretanje, pracenje,razvoj,debagovanje alate I kompajler tj. za prevodjenje).

14. Kako se podešava operativni sistem za pisanje Java programa?

Pet putanja u svakom OS definise koji programi mogu da se pozivaju iz komandne linije ili iz Shell-a;tu treba da se doda putanja do foldera gde se nalazi JDK pa podfolder bin.

15. Šta čini elemente programskog jezika Java?

Program cini niz deklarativnih(njima se definisu neki elementi programa) i izvrsnih naredbi(njima se izvode elementarne obrade podataka), a njih čine niz leksickih simbola. Leksicki simboli ili tokeni predstavljaju nizove znakova I oni se dele na: identifikatore, lterale, ključne reči, operatore, separatore, komentare(posebna vrsta praznina koje prevodilac zanemaruje). Leksicki simboli mogu da se pisu spojenu ili sa odredjedjenim brojem praznina u koje spadaju: znak za razmak, tabulacija, vertikalna tabulacija, prelazak u novi red, prelazak na novu stranu.

16. Šta su identifikatori, a šta ključne reči?

Identifikatori (moraju da pocinju malim slovom,$ ili donjom crtom I razmaci izmedju reci nisu dozvoljeni s tim da se druga rec uvek zapocinje velikim slovom)su imena promenljivih,klasa,objekata,memorijskih lokacija,datoteka,metoda,atributa .Kljucne reci su rezervisane reci u pr. jeziku koje ne mogu da se koriste kao identifikatori: public,new,void,if,else,svi primitivni tipovi itd.

17. Šta su i kako se dele tipovi podataka? Navedite neke tipove podatka u Javi.

Tip podataka definise velicinu i organizaciju podataka, opseg mogucih vrednosti i skup operacija koje se mogu obaviti nad tim vrednostima. Primitivni tipovi podataka(ne mogu da se rastave na manje elemente koji bi mogli nezavisno da se obradjuju, zato se kaže da oni nemaju strukturu I ugradjeni su u programske jezike; promenljive primitivnih tipova podataka se cuvaju po vrednosti): celi brojevi,realni brojevi, logicke vrednosti, znakovi alfabeta.

-byte,short,int,long,float,double,char,boolean

Klasni(objektni) tipovi podataka(sastoje se od nekoliko elemenata koji mogu da se nezavisno obradjuju): nizovi I klase.

-String,nizovi

18. Po čemu je specifičan tip String? Da li se radi o prostom tipu podatka?

Predefinisani klasni tip u Java datoteci, specifican po tome kad se kreira objekat tipa String ne mora da se poziva konstruktor vec moze direktno; pocinje I zavrsava se sa “ ; ima ugradjen operator sabiranja i dodele vrednosti(ostali objekti nemaju).

19. Šta su promenljive, a šta konstante? Kako se deklarišu u Java programima?

Promenljiva je ime lokacije u glavnoj memoriji koja ima odredjeni tip koji cuva vrednost I cija vrednost moze da se menja u toku izvrsenja programa. Konstanta je promenljiva cija vrednost ne moze da se menja, I da bi ona bila konstantna stavlja se kljucna rec final ispred identifikatora koji je sastavljen samo od velikih slova.

20. Šta predstavlja blok naredbi?

Naredba je osnovna jedinica obrade u programima. Blok naredbi je niz naredbi koje se izvrsavaju jedna za drugom. Pise se unutar viticastih zagrada{}.

21. Zbog čega su značajni komentari u Java Programima? Koje vrste komentara poznajete? 4. Šta su komentari i zašto se koriste u Java programima?

Komentar predstavlja neki proizvoljni tekst u programu koji objasnjava neke delove programa drugim programerima radi lakseg snalazenja. U Javi postoje tri vrsta komentara: 1)kratak komentar koji sadrzi tekst samo u jednoj liniji pod //; 2) duzi komentar u nekoliko linija koji pocinje /* I zavrsava se */ ; 3)dokumentacioni(sluzi sa pisanje dokumentacije unutar samog koda I za razliku od obicnih komentara, oni mogu sadrzati HTML tagove i specijalne reci koje počinju znakom @.)

22. Koje vrste operatora poznajete?

Operatori mogu da se primenjuju na jedan operand (unarni operatori), dva operanda (binarni operatori) ili tri operanda (ternarni operatori). Unarni: mogu kao prefiks ili postfiks – brojac++(inkrementiranje) ili ---- brojac(dekrementiranje) , logicki(NOT). Binarni: rade sa dva operanda I to su: aritmeticki(sabiranje,oduzimanje..), relacioni(< > == itd.), logicki(AND, OR..), operator dodele .Ternarni radi sa tri operanda I oznaava se stavljanjem znaka pitanja između prvog i drugog operanda i dve tačke između drugog i trećeg operanda

23. Šta su operatori?

Operatori predstavljaju operacije koje se izvrsavaju nad operandima pri cemu daju odredjeni rezultat.

24. Objasnite kako se vrši unos i prikazivanje podataka u konzolnim Java programima? 43. Kako se učitavaju stringovi sa tastature? 44. Objasnite formatiranje izlaza primenom naredbe printf().

Unos se najcesce realizuje kreiranjem objekta klase Scanner preko operatora new. Ima ugradjene metode za unos primitivnih tipova podataka(nextInt,nextDouble..) I klase String(next-ucitava String do prvog blanka,nextLine-ucitava celu liniju), moze citati I iz datoteke. System.in predstavlja standardni ulaz za podatke, dok System.out standardni izlaz za podatke(System je klasa, out je staticki atribut). Ima ugradjene metode print(stampa se sve u jednom redu), println(stampa se jedno ispod drugog), printf(prvi argument je formatirani string (objekat tipa String) kojim se odredjuje format izlaznih podataka,koj pocinje znakom %,a zavrsava se slovom d za cele brojeve, f za realne I s za stringove,c je za char. Izmedju njih moze da bude naveden broj koji oznacava koliko ce mesta zauzeti broj, npr. %2f znaci da zauzima tacno 2 mesta, a %2.2f znaci da zauzima 2 mesta sa 2 decimale).

25. Šta predstavlja paket u Java programima?

Paket je kolekcija klasa(folder) koja sluzi nekoj vrsti posla I zbog toga cini funkcionalnu celinu, ali nema nikakvo ponasanje vec sluzi samo kao kontejner za te klase.

26. Šta su relacioni operatori? 27. Koje relacione operatore poznajete? 28. Kako se vrši poređenje objekata? Objasnite metodu equals().

Operatori koji uporedjuju vredosti dva operanda. Mogu pored brojeva da uporedjuju I char, boolean.Objekti ne mogu da se porede sa == jer bi se onda poredili po reference, a da bi se po vrednosti uporedjivali to se radi preko ugradjene metode equals() definisanom u klasi Object. Sami odredjujemo po cemu ce se porediti objekti. Primeri: < , >, <= , >=, !=, ==.

29. Navedite opšti oblik naredbe if. Kada se izvršava naredba ili blok naredbi vezan za naredbu if? 30. Napišite if iskaz koji povećava plaćanje za 3% ako je rezultat veći od 90. 31.slicno kao 30. 32.-||-

```
If(uslov){      za 30. uslov ce biti rez>90

Vise naredba; a ako je samo jedna moze bez viticastih zagrada;    za 30. naredba ce biti rez *=  1.03;

}
```

Naredbe se izvrsavaju za naredbu if ukoliko je tacan uslov, ako nije preskacu se.

33. Kada se izvršava naredba ili blok naredbi vezan za klauzulu else i if - else strukturi? 34. Kada ćete koristiti if else naredbu? 35. Kada se izvršava naredba ili blok naredbi vezan za klauzulu else if?

Izvrsavaju se naredbe ukoliko je uslov netacan. Else if se koristi kada je prvobitni if uslov netacan ali je potrebno da se ispita jos neki dodatni uslov kada se to desi.

36. Kada se koristi naredba grananja switch? 37. Objasnite ulogu naredbe break u višestrukom grananju.

switch(naziv promenljive u zavinosti od koje se ispituje neki uslov){

case 1: naredba;

break;

case 2: naredba; ….

break;

default: naredba;

}

Moze da se koristi umesto if-else granjanja, sasvim je svejedno. Cim se nadje odgovarajuci case, iskaz break prekida tok trenutne petlje, izlazi Iz tela petlje I prelazi na sledeci kod iza petlje.

38. Šta je uslovni operator (?) i kako se koristi?

Uslovni ili ternarni operator istu funkciju vrsi kao I if grananje zbog cega se moze lako zameniti upravo tim nacinom grananja sto se u praksi uvek I radi, osim u nekim izuzetnim slucajevima.

Koristi se: logicki izraz? Izraz1 : izraz2

GRUPA PITANJA 2

39. Opišite ulogu klase Math u Java programima? Da li se kreira Math objekat za pozive metoda?

Klasa Math se koristi za resavanje matematickih problema, ova klasa ima staticke metode koje obezbedjuju napredne matemticke operacije. Ne kreira se objekat jer se pozivaju samo staticke metode sledecim nacinom: Math.abs() ili naziv bilo koje druge metode koja nam je potrebna.

40. Pokažite polimorfizam na primeru metoda klase Math.

Math.max(2,5) → vraca 5 ; Math.max(2.5,5.5)→ vraca 5.5 (Polimorfizam oznacava metode koje se pozivaju na isti nacin ali daju razlicite rezultate u zavinosti od parametara koji se koriste pri pozivu). Kada su prosledjeni parametri razlicitog tipa Java automatski vrsi konverziju iz nizeg u visi tip podataka I onda vraca vrednost tog viseg tipa.

41. Objasnite redosled izvršavanja matematičkih operacija u Java programima.

Prvo mnozenje I deljenje, zatim sabiranje I oduzimanje. Pritom postuju se zagrade koje se upotrebljavaju,tj. prvo se vrsi izraz u unutrasnjim pa onda u spoljasnjim zagradama.

42. Objasnite eksplicitnu i implicitnu konverziju primitivnih tipova podataka.

Implicitna konverzija je kad se cast-uje iz manjeg tipa podataka u veci, to se radi automatski I ne mora da se stavlja operator kastovanja npr. double x = 5. Eksplicitna konverzija je iz veceg u manji I tu mogu da se izgube odredjene vrednosti npr. (int)3.14 on vraca 3.

45. Koje tipove petlji poznajete u Java jeziku? Napišite ih u opštem obliku. Pokažite jedan jednostavan Java primer urađen primenom različitih vrsta petlji. 46. Napravite razliku između while i do-while petlje. Pokažite na primeru. 47. Po čemu se razlikuju for i for-each petlja?

While(izraz){

Naredba 1,2..; Naredba/naredbe se izvrsavaju sve dok je uslov(izraz)ispunjen tj. tacan.
}

Do naredba 1,2{

}while(izraz);

Prvo se izvrsavaju naredbe(telo petlje) pa se proverava uslov. Za razliku od while petlje, do-while petlja ce se izvrsiti barem jedanput, dok while petlja ne mora ni jednom.

Kada je potrebno izvrsiti telo petlje za sve vrednosti koristi se for petlja ili for-each, tj. kada znamo tacno koliko puta treba da se petlja izvrsi. Isti problem se moze resiti for ili while petljom. For se razlikuje od for –each po tome sto se for sastoji iz tri dela, dok se for-each sastoji iz dva dela: element : niz ili lista.

For(inicijalizacija; logicki izraz; zavrsnica){

Naredba 1,2..;
}

48. Pokažite na primeru i objasnite primenu ugnježdenih petlji.

49. Šta su metode? Kako se definišu metode? Kako se pozivaju metode? 54. Objasnite kako metoda vraća vrednost?

Metode su potprogrami koji sluze za razlaganje slozenijeg programa na manje delove, kako bi se pojednostavio program I lakse se procitao I protumacio. Definisu se modifikatorom vidljivosti(public,private,protected,default, a moze I neki dodatni modifikatorom npr. static) pa tipom vrednosti koju vraca(ako ne vraca nijednu vrednost onda je void I vraca zadatu vrednost preko return), pa identifikatorom metode I na kraju parametrima u zagradi koji se metodi prosledjuju na dalju obradu u telu metode. Metode se pozivaju imenom I proslede se konkretni parametri kao literal. *Literal je konkretna vrednost zadata u programu(npr. literal tipa int je 5).*

50. Šta su modifikatori pristupa? Objasnite ih pojedinačno.

Modifikatoru pristupa su zapravo kontrola pristupa koja definise ko sve moze da koristi odredjenu metodu, tj u kom ce opsegu(scope) sve da vazi. Moze da bude public – u svakom paketu I klasi metoda moze da se pozove tj. bilo koj Java program moze da je pozove; private – metoda moze biti pozvan samo u klasi u kojoj se nalazi tj. od primerka njegovog objekta; protected – metoda moze biti pozvana u svim klasama koje se nalaze u odredjenom paketu, van tog paketa ne moze; default – isto kao protected, samo ako se nista ne zada onda je default tj. on se ne pise.

51. Na osnovu povratnog tipa kako delimo metode? Tj. na osnovu prosledjenih elemenata

Na one koje rade po vredosti(prosledjeni paramtri su primitivni tipovi) I one koje rade po referenci(prosledjeni parametric su objekti).

52. Na osnovu načina pozivanja kako delimo metode?

Staticke metode(pozivaju se po nazivu klasa I pripadaju celoj klasi , a pristupaju samo statickim podacima klase) I objektne(pozivaju se po nazivu objekta i ne pripadaju celoj klasi vec definisu ponasanje odredjenog objekta).

53. Navedite prednosti primene metoda u programima.

Jedna metoda se moze na vise mesta koristiti(reusability), samim tim skracen je kod I bolja je funkcionalnost I univerzalnost koda jer ce odredjena metoda raditi za sve slucajeve gde se prosledjuje isti tip parametara a ne samo za jedan slucaj.

55. Šta je preklapanje metoda (overloading)?

Metode sa istim imenom(identifikatorom) ali razlicitim potpisom(potpis – redosled,tip,broj prosledjenih parametara). Kada postoji overloading, Java prevodilac zna koju metodu da koristi. Koristice onu metodu ciji stvarni parametric odgovaraju deklarisanim. Primer: konstruktori.

56. Šta je redefinisanje metoda (overriding)?

Kada postoji metoda iz superklase koja je potrebno da se koristi u subklasi tj. da se redefinise odnosno da joj se promeni ponasanje. Vrsi se na sledeci nacin: prepise se potpis, doda se @Override anotacija iznad,I napise se nova implementacija , a moze I da se zadrzi stara I doda nova. Primer: toString metoda iz klase Object se redefinise u nekoj subklasi.

57. Šta je jednodimenzionalni niz? 58. Kako se konstruiše niz? 59. Kako se pristupa članovima niza? 60. Ukoliko se ne dodeli vrednost svim članovima niza, koje podrazumevane vrednosti dobijaju ti članovi? 63. Objasnite učitavanje i obradu jednodimezionalnog niza.

Skup vrednosti istog tipa cija je velicina(duzina niza) fiksno zadata odmah. Pojedinacne promenljive se zovu elementi niza , a njihov redni broj se zove indeks. Clanovima niza se pristupa imeNiza[indeks] pomocu operatora za pristupa tj. []. Ako je primitivni tip niza podrazumevana vrednost pre dodeljivanja je 0, a ako je objektna se javlja runtime exception tj. nema vrednost. Definise se prvo, pa se ucitava pomocu Scanner-a.

I nacin: tipPodataka[] imeNiza = new tipPodataka[duzinaNiza];

II nacin: tipPodataka[] imeNiza = {elementi};

61. Objasnite na koji način klasa Arrays olakšava rad sa nizovima?

Klasa Arrays je pomocna klasa iz paketa java.utils I ona ima razne ugradjene staticke metode koje olaksavaju rad sa nizovima npr. toString(), sort(), copyOf().

62. Navedite nedostatke nizova.

Fiksiranje duzina niza(zbog toga se ne mogu dodati promenljive nizu,ne mogu se ni otkloniti – tj. ne moze se dinamicki upravljati memorijom).

64. Algoritam sortiranja jednodimenzionalnog niza.

```
for(int i = 0; i < niz.length - 1; i++){
        for(int j = I + 1; j < niz.length; j++){
                if(prviNiz[i] > prviNiz[j]){
                int pomocnaPromenljiva = niz[i];
                niz[i] = niz[j];
                niz[j] = pomocnaPromenljiva;
                }
        }
}
```

GRUPA PITANJA 3

65. Čemu služe dvodimenzionalni nizovi? 66. Kako se definiše dvodimenzionalni niz? 68. Deklaracija i inicijalizacija dvodimenzionalnog niza - objasnite!!! 69. Objasnite učitavanje i obradu dvodimezionalnog niza.

Za smestanje podataka koji su u obliku tabele ili matrice. Primer: tabela u fudbalu ko je s kim igrao I koji su rezultati.

I nacin: tipPodataka [] [] imeNiza = new tipPodataka[vrste] [kolone];

II nacin: : tipPodataka [] [] imeNiza = { [1,2,3,4],[5,6,7,8]} ;

```
for(int i = 0; i < matrica.length; i++){
        for(int j = 0; j < matrica[i].length; j++){
          matrica[i][j] = scanner.nextInt();          za stampanje: System.out.print(matrica[i][j] + " ");
          }
                                                                  System.out.println("");
}
```

67. Šta je matrica?

Dvodimenzionalni niz ciji elementi predstavljaju niz celobrojnih nizova.

70. Unos dvodimenzionalnih nizova u metode - objasnite!

Dvodimenzionalni niz se u metode prosledjuje kao parametar(bice tipPodataka[][] imeNiza – ime moze da bude promenljiva a u pozivu se prosledjuje konkretna matrica) tj. kao referenca niza. Matrica se kao I niz cuva u memoriji po referenci.

71. Šta je objektno-orijentisano programiranje? 72. Šta je objekat? Objasnite primenu objektnog tipa podataka. 75. Od kojih elemenata se sastoji klasa? 85. Šta je softverski objekat? Navednite primere softverskih objekata.

Programska paradigma po kojoj se softverski sistem gleda kao skup manjih celina-objekata (promenljiva klasnog tipa, primeri: student, banka, krug…)koji raspolazu skupom podataka I ugradjenim metodama iz klase preko cijeg sablona je instanciran koje obradjuju te podatke. Objekat moze da sprovede akcije nad sobom tj da iskoristi neku metodu ili da zahteva od drugog objekta da on izvrsi akciju, I na taj nacin medjusobnom interakcijom resava se kompletan programerski problem. Cuvaju se po referenci u memoriji(samo se primitivni tipovi cuvaju po vrednosti). Objekat ima identifikator, stanje(atributi ili polja), ponasanje(metode). Klasa se sastoji kao I objekat od identifikatora, atributa I ponasanja.

86. Šta je objekat, a šta referenca na objekat?

Referenca je pokazivac gde je smesten objekat u memoriji(ime objekta se nalazi u delu memorije koji se naziva stack, a sadrzaj objekta na hip).

73. Šta je klasa? Kako se grafički (UML) predstavlja klasa?

Klasa je sablon(opsti opis objekata)koja ima sadrzaj I strukturu objekta po kojoj se instanciraju objekti. Ona u stvarnosti ne postoji za razliku od objekata. Graficki se predstavlja kao pravougaonik koji je sastavljen iz tri celine: prva sadrzi identifikator klase, druga atribute I treca sadrzi ime metoda(ispred njenog imena stoje simboli za vidljivost: + je public, - je private, # je protected) pa dve tacke I onda povratni tip koji ona vraca.

74. Objasnite kako se kreira klasa? 77. Opšti oblik predstavljanja klase - objasnite!

Modifikator pristupa(opciono) class nazivKlase{}

Kako se zove klasa tako mora I da se zove fajl u kome se nalazi.

76. Objasnite razliku i primenu statičkih i objektnih članova klase.

Staticki clanovi(atributi) klase su isti za sve objekte te klase, dok je za objektne obrnuto. Vrednosti statickih clanova se nalaze unutar same klase , dok objektnih aktuelnih unutar objekta.Staticka metoda ne zavisi od konrektnog stanja objekta (vazi za sve objekte) vec od parametara.

78. Šta su lokalne, a šta globalne promenljive? Objasnite oblast važenja promenljive.

U Javi postoji 4 oblasti vazenja(scope): najslabija je unutar petlje koja se nalazi u metodi-lokalna promenljiva(ne mora u petlji da bude,bilo gde definisana unutar metode); parametar metode(vazi samo u metodi); atribut unutar klase(vazi u celoj klasi tj. u objektu klase)-globalna promenljiva; najjaci je staticki atribut koji vazi u svim objektima klase.

Alociranje promenljive je odredjivanje memorijskog prostora za smestanje vrednosti neke promenljive, dealociranje je praznjenje tog prostora.

79. Objasnite ključne reči this i super.

Rec this oznacava trenutni objekat te klase kojoj pripada koji se poziva. U nekim slucajevima je nepotrebno koristiti kljucnu rec this ali se u praksi koristi uglavnom uvek jer se na taj nacin jasno razlikuju objketni atrubuti klase od lokalnih promenljivih metoda. Super je pokazivac na na nadklasu(koristi se za pozivanje metoda I konstruktora superklase).

80. Šta su omotačke klase i koja je njihova namena?

Omotacke klase su odredjene klase koje umota primitivne tipove podataka I na taj nacin ih pretvara u objekte kako bi oni koristili ugradjene metode koje poseduju omotacke klase jer primitivni tipovi nemaju ugradjene metode I funkcionalnosti. Za sva 8 primitivna tipa podataka postoji omotacka(wrapper)klasa: Integer,Double…Namena im se ogleda u tome da olaksava rad sa primitivnim tipovima podataka pomocu svojih ugradjenih metoda I statickih konstanti npr. min_value; max_value; parseInt(pretvara String u int-ova metoda postoji za sve ostale primitivne tipove)…

81. Da li je String prost tip podataka? Objasnite ulogu klase String u Java programima.

String je klasni tip podataka, jednom napravljen String ne moze da se menja vec moze da se napravi novi kome ce se dodeliti ista referenca. Ima 13 vrsta konstruktora ali ne mora da ih poziva vec moze direktno da se napravi objekat klase String. Uloga se ogleda u tome da se mogu cuvati reci, recenice I moze se vrsiti razna obrada sa tim tekstulanim tipovima podataka pomocu raznih ugradjenih metoda u klasi String. Klasa String se nalazi u paketu java.lang(I zato se nikad ne importuje).

82. Šta je učaurivanje (enkapsulacija) podataka? Objasnite ukratko ovaj objektno - orijentisani kocept.

Enkapsulacija je podesavanje modifikatora atributa klase na private, a pristup njihovim vrednostima vrsi se preko javnih metoda gettera(vraca vrednost) I settera(postavlja vrednost) koji su deo javnog interfejsa klase. Svrha ovakvog rada je zastita podataka(vrednosti mogu da se menjaju samo u kontrolisanim uslovima, I odredjeni detalji klase se mogu sakriti ukoliko je nepozeljno da ih drugi programeri vide).

83. Konverzija između stringova i nizova karaktera - objasnite!

Niz karaktera(char) I String nije isto, konverzija se radi pomocu metode Arrays.toCharArray().

84. Konverzija oznaka i numeričkih vrednosti u stringove - objasnite!

Radi se na sledeci nacin: pozove se wrapper klasa Integer, pa se pozove metode iz nje Integer.toBinaryString() predstavlja broj u binarnoj reperezentaciji Stringa, postoji ui u oktalnoj I u heksadecimalnoj– naravno isto vazi za ostale tipove numerickih podataka.

87. Šta je konstruktor? Kako se konstruiše objekat? Da li konstruktor može da bude preklopljen - objasnite?

Konstruktor je metoda(nije ni objektna ni staticka) koji se koristi za stvaranje objekata svoje klase. Konstruktor ima identifikator,parametre I modifikator pristupa isti kao I klasa u kojoj se nalazi, a nema tip podataka jer je rezultat uvek referenca na novi objekat date klase. Konstruktori se mogu pozvati samo uz operator new(new imeKlase(listaArgumenata) – na ovaj nacin kreira se objekat tj. referenca na njega). Konstruktor moze da bude preklopljen(overloaded) jer moze da postoji vise vrsta konstruktora sa istim identifikatorom a razlicitim prosledjenim parametrima(uglavnom se u praksi pisu 3: prazan-bez ikakvog parametra; preopterecen-sa svim atributima koji se nalaze u klasi; konstruktor kopije- prosledjeni parametar je objekat te klase ciji se atributi kopiraju u nov objekat).

GRUPA PITANJA 4

88. Razlika proceduralnog i objektno-orijentisanog programiranja?

Proceduralno je usmereno ka pravljenju metoda, sto znaci da napravljena metoda proceduralnim programiranjem ne moze da se koristi u svim aplikacijama jer koristi Main metod, dok u objektno-orijentisanom moze jer on nudi vecu fleksibilnost I modularnost pri razvoju softvera za visestruku upotrebu. Kod proceduralnog podaci I operacije su odvojeni sto znaci da podaci moraju da se ubacuju u metode, dok su oni u objektno-orijentisanom smesteni u jedan objekat sto oslikava stvaran svet.

89. Vrste veza između klasa, navedite i objasnite. 90. / 91. Šta je agregacija, a šta kompozicija?

Postoji 6 vrsta veza izmedju klasa:

1) Zavisnost – naslabija veza

2) Asocijacija – jedan objekat upotrebljava servise drugog

Agregacija I kompozicija su posebne vrste asocijacije I one definisu vezu “sadrzi” pri cemu jedna klasa sadrzi jedan ili vise objekata druge klase

3) Agregacija – ako se objekat jedne klase izbrise druga I dalje postoji

4) Kompozicija – iskljucivo vlasnistvo tj. kada se objekat jedne klase izbrise objekat druge ne postoji

5) Generalizacija – podklasa ima sve karakteristike klase

6) Realizacija – veza izmedju klase I njenog interfejsa

92. Primena klasa BigInteger i BigDecimal, objasnite.

Koriste se za velike vrednosti brojeva kada je potrebna velika preciznost. Date su u paketu java.math I sadrze ugradjene metode za izvrsavanje aritmetickih operacija sa velikim brojevima.

93. Klase StringBuilder i StringBuffer - primena, objasnite.

Pomocu ove dve klase moze da se doda I ubaci novi sadrzaj u String objekat sto je kod obicne String klase nemoguce uraditi(jednom formiran String objekat ne moze da se menja). StringBuffer se koristi kad je potrebno da se izvrsi vise zadataka istovremeno, jer su metode za modifikaciju bafera sinhronizovane. StringBuilder ne pravi takozvano djubre u memoriji za razliku od StringBuffera I brze radi od njega.

94. Šta je nasleđivanje? *Tri/(Cetiri) principa OO programiranja* 95. Navedite prednosti i nedostatke nasleđivanja.

To je vazna odlika objektno-orijentisanog programiranja pri cemu se definise neka opsta(superklasa) klasa koja sadrzi sve zajednicke atribute I metode vise slicnih klasa. Druge klase koje imaju ove zajednicke atribute I metode nasledjuju tu superklasu upotrebom kljucne reci extends(Java podrzava jednostruko nasledjivanje(nedostatak) I klasa moze da nasledi apstraktnu ili obicnu klasu).Ovo se radi da bi se izbeglo ponavaljanje istih atributa I metoda,I s tim da moze isti softver da se koristi vise puta-prednosti.

Principi: polimorfizam, nasledjivanje, (data hiding-posledica enkapsulacije tj. sakrivanje podataka kada se podesi modifikator private), enkapsulacija

96. Objasnite odnos superklase i potklase.

Sve sto se definise u superklasi se automatski nasledjuje u potklasi I potklasa moze da doda svoje neke specificne metode I atribute tj. mogu da prosiruju superklasu. Atributi ne mogu da se override, a metode mogu.

97. Upotreba konstruktora superklase u potklasi - objasnite.

Potklasa ne moze da nasledi konstruktor vec moze samo da ga u svom konstruktoru pozove pomocu kljucne reci super. U prvom redu se nalazi obicna sintaksa za konstruktor ali ispod mora odmah da stoji super.(nasledjeni atributi).

98. Šta je apstraktna klasa? Kako se kreira apstraktna klasa?

Apstraktna klasa(nema realnu implementaciju u zivotu) je superklasa koja je toliko uopstena da ne moze da kreira svoje objekte jer nema dovoljno informacija za to. . Kreiraju se tako sto im se doda u obicnu sintaksu za klasu I modifikator abstract posle modifikatora vidljivosti(public abstract class imeKlase{}).

99. Navedite osobine apstraktne klase. 100. Šta se dešava sa apstaktnim metodama superklase u potklasama?

Ona moze da bude iskljucivo apstraktna ako sadrzi apstraktne metode(u potklasama se one override). Apstraktna klasa koja ima apstraktne metode mora da bude apstraktna, ali ne mora da znaci da uvek mora da ih ima I neapstraktna klasa ne moze da ima apstraktne metode a mora da primeni sve apstraktne metode koje je nasledila od superklase. Kada je konkretna superklasa potklasa moze da bude apstraktna, a potklasa moze da nadjaca metod superklase da bi ga definisao kao apstraktni I tada ta klasa postaje apstraktna

101. Šta je interfejs? Kako se kreira interfejs?

Interfejs lici na apstraktnu klasu I cuva se kao klasa u java.file, ali se razlikuje od nje. On sadrzi zajednicka svojstva povezanih ali I nepovezanih klasa, I moze da ima samo apstraktne metode I konstante. Sve metode u interfejsu su apstraktne I javne pa ti modifikatori nema potrebe da se pisu, mogu I ne moraju da imaju prosledjene parametre I nemaju telo metode. Jedna klasa moze da implementira vise razlicitih interfejsa. Interfejs ne moze da kreira svoj objekat operatorom new kao I apstraktna klasa. Veza izmedju interfejsa I klase se definise kao nasledjivanje I ne razlikuje se od obicnog nasledjivanja izmedju klasa.

Kreira se:

```
public interface imeInterfejsa{

        metode;

}
```

102. Navedite razlike između apstraktne klase i interfejsa? 103. Nasleđivanje klase i implementacija interfejsa - suštinske razlike - objasnite.

Apstraktne klase mogu da sadrze i nestaticke atribute, mogu da imaju I apstraktne I obicne metode za razliku od interfejsa. Klasa moze da nasledi jednu klasu ali moze da implementira vise interfejsa.

104. Šta je polimorfizam? Navedite primere polimorfizma u Java programima.

Polimorfizam je jedan od tri principa OO programiranja koji predstavlja pojavu da se jedan objekat moze javiti u vise oblika, tj. to je mogucnost da se jedna ista metoda ponasa razlicito u zavisnosti od toga koji je objekat poziva (on omogucava da se ista funkcija superklase razlicito primenjuje u potklasama). Objekat potklase moze da zameni objekat superklase. Primeri: Overloading,Overriding.

105. Šta je dinamičko povezivanje (dynamic binding) ?

Mogucnost da se objekat potklase kreira pomocu konstruktora superklase koju nasledjuje, pri cemu se odredjuje prilikom izvrsenja iz koje ce klase pozvati metodu, tj. ako ona ne postoji u potklasi onda je trazi u superklasi I to se sve zove dinamicko povezivanje. To znaci da metoda potklase nadjacava metodu superklase jer ce se ona prva pozvati.

106. Navedite značaj primene interfejsa Comparable<T>. Kako se porede objekti u Java programima?

Znacaj se ogleda u tome da mogu da se porede objekti bilo koje klase. Porede se sa equals metodom Ili kada implementiraju interfejs Comparable automatski se override metoda compareTo pomocu koje isto mogu da se porede, a mogu I pomocu metode compare kada se implementira interefejs Comparator.

GRUPA PITANJA 5

107. Eksplicitna konverzija tipa objekta - objasnite.

Konverzija je pretvaranje reference objekta u drugu referencu objekta. Implicitna konverzija(upcasting) je uvek moguca jer je objekat potklase uvek I objekat superklase(Object o = new Student()). Kada se vrsi konverzija objekta superklase u objekat potklase(downcasting) vrsi se eksplicitna konverzija objekta. Da bi konverzija bila uspesna objekat superklase mora da bude primerak(objekat) potklase.

Student s = (Student) o gde je o objekat superklase Object

108. Konverzija primitivnih tipova i konverzija tipova objekata - objasnite.

Konverzija primitivnih tipova vrednosti se razlikuje od konverzije tipa neke reference objekta. Konverzija primitivnog tipa vrednosti vraca novu vrednost:

npr. int broj = 45; byte noviBroj = (byte) broj; → sada je dodeljena nova vrednost promenljivoj noviBroj

Medjutim, konverzija tipa reference objekta ne kreira novi objekat:

npr. Object o = new Circle(); Circle c = (Circle) o; → sada reference pokazuju na isti objekat

109. Upoređivanje dva objekata metodom equals() - objasnite.

Pocetno ugradjena implementacija metode equals koja je definisana u klasi Object proverava da Ii objekti pokazuju na istu referencu sa operatorom ==, ali u potklasi tj. klasi(bilo koja klasa kada se napravi implicitno pozdrazumevano nasledjuje klasu Object) gde se poziva se ona nadjacava(override), I onda se tu proverava da Ii su jednaki po sadrzaju.

110. Memorisanje objekata primenom klase ArrayList - objasnite.

Napravi se objekat klase ArrayList(nalazi se u paketu java.util), definise se tip I onda se pozivaju potrebne metode(add,remove,contains..). Elementi ArrayList mogu da budu samo objekti. Kada se kreira lista za memorisanje objekata preko klase ArrayList njena duzina nije fiksna.

111. Navedite i objasnite preporuke za projektovanje klasa.

Kohezija , konzistentnost(treba da se sledi standardni Java stil I konvencija naziva), ucaurenje, jasnoca(klasa bi trebalo da ima interfejs koji je razumljiv I lako objasnjiv), kompletnost, objektni I staticki, nasledjivanje I agregacija, nasledjivanje I agregacija, interfejsi I apstraktne klase

112. Šta je izuzetak? Navedite neke najpoznatije izuzetke?

Izuzetak nije isto sto I greska, to je objekat koji predtavlja neku gresku ili uslov koji sprecava normalno izvrsenje programa. Izuzeci se izbacuju (stvaraju) iz nekog metoda, a onaj koji poziva taj metod, moze da uhvati (catch) izuzetak i da nesto uradi sa njim(ako program ne reaguje na izuzetak doci ce do pucanja programa, I zbog toga se uvek odredi njegova reakcija kako bi moglo normalno da se nastavi izvrsenje programa ili da se zaustavi pod kontrolisanim uslovima).

Primeri: ArrayIndexOutOfBounds, InputMismatchException, NullPointer, ArithmeticException(deljenje nulom)

113. Koje vrste izuzetaka poznajete?

Klasa Throwable je korenska klasa svih izuzetaka.

1)sistemske greske(objekti klase Error)-opisuje unutrasnje greske sistema; moraju u try catch

LinkageErorr, VirtualMachineError

2)izuzeci(objekti klase Exception)-opisuje greske prouzrokovane programom I spoljnim okolnostima, ovakav tip greske se hvata I obradjuje u programu; moraju u try catch

-IOException

3)izuzeci u fazi izvrsenja(objekti klase Runtime Exception) – opisuju greske programiranja(logicke greske); ne moraju da se stavljaju u try catch vec ih izbacuje JVM

-ArithmeticException, NullPointer, IndexOutOfBounds

114. Objasnite izbacivanje i obradu izuzetaka.

Izbacivanje je kada program otkrije gresku, on kreira objekat odgovarajuce klase izuzetaka I izbaci ga(throw). Ako se izbacuje izuzetak koji nije RunTime u potpisu metode mora da ima throws I naziv izuzetka koji se baca. Kada se izbaci izuzetak on mora da se uhvati tj. da se izvrsi njegova obrada u try catch bloku.

115. Kada je potrebno upotrebiti izuzetke?

Izuzeci usporavaju program I zato se koriste samo kad je iskljucivo potrebno tj. ako se obradjuje izuzetak u metodi u kojoj se javlja onda nije potrebno da se izbacuju I koriste izuzeci. Za jednostavne greske nije potrebno koristiti izuzetke vec se moze ispitati ispravnost programa pomocu if iskaza. Dakle izuzeci se bacaju kada se desava nenormalno ponasanje I hvataju se I obradjuju kada metoda neke klase sa kojom se radi baca izuzetak. Koriste samo u neocekivanim uslovima za javljanje gresaka I tada se obradjuju u try catch bloku.

116. Kreiranje vlastitih klasa izuzetaka.

Kreira se samo kada nijedan postojeci u Javi ne odgovara izuzetku koji je potreban. Kreira se tako sto se nova klasa napravi koja nasledjuje Exception I pozove konstruktor sa imenom istim kao klasa, u njemu se samo stavi kljucna rec super kako bi nasledio sve parametre klase Exception. Moze da ima 4 vrste konstruktora.

117. Olančani(chained) izuzeci - objasnite.

Izbacivanje izuzetaka zajedno sa drugim izuzecima cini lanac izuzetaka. To se desava kada se izbaci novi izuzetak pored pocetnog izuzetka(postoji jedna metoda koja izbacuje izuzetak , drugametoda koja poziva prvu i hvata taj izbacen izuzetak u catch I tu ga ugnjezdi u novi izuzetak koji se izbacuje, treca metoda hvata novi izuzetak u catcj koji sadrzi oba).

118. Razlika između throw i throws - objasnite.

Throw je unutar koda kada se baca izuzetak(throw new), a throws je u potpisu. Kada postoji throw unutar koda ne mora da se pise throws u potpisu (osim ako je IOException, znaci vazi samo za RunTime).

119. Klauzule try, catch i finally - objasnite upotrebu.

Izuzetak se javlja u kodu,koji se izvrsava u bloku try,koji sadrzi kod koji se izvrsava u normalnim okolnostima, u catch se hvata izuzetak I tu se nalazi kod koji ga obradjuje. Ako se javi izuzetak u try on se odmah prebacuje na catch gde se vrsi njegova obrada. U bloku finally se nalazi kod koji se izvrsava uvek bez obzira na to da li se javlja greska ili ne.

120. Šta su datoteke?

Datoteke su skup trajno sacuvanih podataka na nekom memorijskom medijumu na jednoj logickoj adresi. Postoje tekstualne I binarne datoteke, tekstualne mogu da se otvore, menjaju I citaju, dok binarne mogu da se otvore takodje ali nista ne moze da se razume jer su heksadecimalnom zapisu. Mogu se transportovati I citati. Svaka datoteka se cuva u direktorijumu sistema datoteka koji obezbedjuje operativni sistem. Apsolutni naziv sadrzi kompletan put do datoteke, a relativni se definise u odnosu na direktorijum u koji je smestena.

121. Ulaz i izlaz tekstualnih datoteka - objasnite.

Za upisivanjeu tekstualnu datoteku uglavnom se koristi klasa PrintWriter(nalazi se u paketu java.io), nakon sto se kreira objekat te klase mogu da se pozovu njene metode print,println I printf koje vrse upis u datoteku. Nakon sto se zavrsi upis, mora da se zatvori datoteka tj. file.close() jer bi se u suprotnom mogli pogresno memorisali podaci. Mogu I druge klase sa upis da se koriste: FileWriter, BufferedWriter. Za citanje iz datoteke(stampanje u output) upotrebljava se klasa Scanner(nalazi se u paketu java.util), nakon sto se kreira objekat te klase, pozivaju se metode nextInt,nextDouble,nextLine…Mogu I druge klase da se koriste za citanje: FileReader, BufferedReader

122. izbaceno pitanje //citanje podataka sa veba

123. Čuvanje binarnih podataka u datotekama - objasnite.

Koriste se klase DataInputStream I DataOutputStream radi filtriranja bajtova. Objekti ove dve klase citaju i pisu primitivne tipove vrednosti i stringove na nacin koji je nezavisan od racunara, sto znaci da moze datoteka da se formira na jednom racunara a da se koristi na drugom

124. Čuvanje objekata u datotekama - objasnite.

Koriste se klase ObjectInputStream, ObjectOutputStream(sadrze sve funkcije kao I DataInputOutputStream pa mogu kompletno da ih zamene)-mora da se implementira Serializable(serijalizacija automatizuje proces memorisanja objekata i nizova tj.cuvanje objekata u datotekama) interfejs I onda se dobijaju binarni podaci.

125. Operacija čitanja i pisanja u datotekama sa slučajnim pristupom – objasnite

Vrsi se pomocu klase RandomAcessFile koja dozvoljava da se podaci citaju ili da se u njih vrsi upis na nekim slucajno odredjenim memorijskim lokacijama. Ova klasa primenjuje DataInput(definise metode za citanje) I DataOutput (definise metode za upisivanje)interfejse . Kada se kreira objekt klase RandomAccessFile, moze da specificira jedan od dva oblika koriscenja: r ili rw. Oblik r znači da je tok samo za citanje, a oblik rw – da je i za citanje i za zapisivanje. Datoteka sa slucajnim pristupom sadrzi niz bajtova. Ona ima pokazivac koji pomera kroz datoteku I na kom se bajtu pozicionira tu se vrsi upis ili citanje.