1. Care este signatura completa a metodei main?

R: public static void main(String arg[]) {}

2. Cum se numeste alfabetul pe care este construit limbajul Java ?

R: unicode

3. Enumerati tipurile primitive ale limbajului Java.

R : aritmetice :{ [intregi] : byte(1),short(2),int(4),long(8), [reale]: float(4), double(8)},caracter : {char(2) }, logic: {boolean(true/false)}

4. Completati valorile variabilelor x si y dupa executia secventei :

int x = 1, y = (x++ > 1 ? – x : x–) ;

R : x=1 y=2

5. Fie declaratia int a=1, b=2, c=3, d =4; Ce va afisa pe ecran in urma executiei metodei:

System.out.println(a + b + “ “ + c + d); ?

R: 3 34

6. Care sunt modificatorii de acces ai membrilor unei clase si care este cel implicit?

R : public, abstract, final ;

7. Care este efectul modificatorului static aplicat unei variabile?

R: pt a specifica faptul ca variabila are o sg valoare comuna tuturor instantelor clasei în care este declarata. Modif valorii acestei variabile din interiorul unui obiect face ca modif sa fie vizibila din celelalte obiecte. Variabilele statice sunt initializate la încarcarea codului specific unei clase si exista chiar si daca nu exista nici o instanta a clasei respective. Din aceasta cauza, ele pot fi folosite de metodele statice. Pt fiec ob exista o zona de mem.

8. Care este efectul modificatorului final aplicat unei metode?

R: nu poate fi rescrisa în subclasele clasei în care a fost declarata

9. Ce semnificatie are cuvintul cheie this folosit ca variabila ?

R: referinta la obiectul curent.

10. Ce semnificatie are cuvintul cheie super folosit ca metoda ?

R : apel constructor supraclasa.

11. Ce reprezinta supraincarcarea metodelor unei clase ?

R : intr-o clasa por edista m multe metode cu acelasi nume dar cu arg diferite (prin tipul lor)(polimorfism).

12. Ce reprezinta un pachet si cum poate fi declarat ?

R : Un pachet consta intr-un grup de clase care au un punct de referinta comun.(ex clase ref la fisiere sunt grupate in java.io). Dorim sa scriem un pachet numit fisiere. Creem un director cu numele fisier in care punem toate clasele ce dorim sa faca parte din pachet. Aceste clase trebuie sa aiba ca prima intructiune: package fisiere.

13. Care este pachetul importat automat in orice aplicatie Java ?

R : java.lang

14. Care sunt modificatorii impliciti ai variabilelor declarate in cadrul unei interfete ?

R: static & final. public daca interfata este declarata public.

15. Care este superclasa tuturor claselor in limbajul Java?

R: Object.

16. Care este superclasa acelor clase ce descriu exceptii a caror tratare nu este obligatorie?

R : RunTimeException

17. Ce exceptie va genera urmatoarea secventa de cod :

Rectangle r[]= new rectangle[10] ;

r[0].setSize(10,10);

R: java.lang.NullPointerException

18. Care este rolul blocului finally dintr-o secventa try-catch-finally de tratare a unei exceptii?

R: Instructiunile din clauza finally sunt executate in ambele cazuri (chiar daca apare sau nu vre-o exceptie)

19. Care este diferenta intre clasele String si StringBuffer ?

R: String: pt siruri constante StringBuffer: pt sir var

20. Dati o modalitate de a converti un sir de caractere intr-un numar intreg.

R : int n=(new Integer(“123”)).intValue(); sau n = Integer.parseInt(”123”)

21. Care este pachetul care ofera suport pentru operatii de I/O?

R: java.io.\*

22. Care sunt superclasele (pt cit/scr) ale fluxurilor care lucreaza la nivel de caracter?

R: Reader / Writer

23. Care este clasa cu ajutorul careia pot fi citite linii de caractere dintr-un flux de intrare ?

String line=( new BufferedReader(new FileReader(“f.txt”))).readLine();

24. Care este clasa care ofera support pentru manipularea la nivel fizic a fisierelor si directoarelor ?

R : File

25. Ce metoda tb obligatoriu supradefinita at cind se declara un fir de executie prin extinderea clasei Thread ?

R : public void run()

27. Ce reprezinta serializarea ?

R: transf. ob (date,metode,signatura) intr-un sir de octeti si i-l salveaza intr-un fisier.

28. Care sunt clasele cu ajutorul carora se pot construi fluxuri de obiecte necesare pentru serializarea, respectiv deserializarea obiectelor?

R: DataInputStream,DataOutputStream care implemn interf. DataInput/DataOutput

26. Care este rolul modificatorului synchronized plasat inaintea unei metode?

R: Pe o instanta de clasa, la un moment dat, poate lucra o sg metoda sincronizata. Daca un alt fir de executie încearca sa apeleze aceeasi metoda pe aceeasi instanta sau o alta metoda a clasei de asemenea declarata sincronizata, acest al 2-lea apel va tb sa astepte înainte de executie eliberarea instantei de catre cealalta metoda

29. Care este rolul modificatorului transient plasat inaintea unei variabile ?

R: elimina var. din procesul de serializare (daca variabila nu e declarata static)

30. Care sunt pachetele principale care ofera suportul pentru crearea interfetelor grafice (GUI) si pentru tratarea evenimentelor generate de componentele grafice?

R: java.awt.\* , java.awt.event.\*

31. Care este superclasa tuturor obiectelor care au o reprezentare vizuala (mai putin a meniurilor) ?

R: Component

32. Enumerati cel putin trei gestionari de pozitionare.

R: FlowLayout, BorderLayout, GridLayout, CardLayout (toate implem interfata LayoutManager)

33. Care sunt cele patru metode fundamentale ale appleturilor?

R: start(), stop(), destroy, paint(Graphics g)

34. Care este gestionarul de pozitionare implicit pentru applet-uri ?

R: FlowLayout

35. Care este diferenta dintr-o fereastra de dialog modala si una nemodala ?

R: modale: blocheaza accesul la fereastra parinte. Nemodale: nu.

36. Ce metoda declara interfata ActionListener ?

R: void actionPerformed(ActionEvent e)

37. Enumerati cel putin 3 evenimente generate de obiectele derivate din clasa Window.

R: WINDOW\_ACTIVATED, WINDOW\_ICONIFIED, WINDOW\_CLOSED

38. Prin ce difera o aplicatie independenta de un applet?

R: intr-un applet: nu este nevoie sa cream un obiect cu numele clasei (acesta este creat automat de browser sau applet-viewer), applet-ul nu ruleaza independent, nu contine functia main(), are in schimb functia init(), se poate incarca de pe web

39. Prin ce difera utilitarul de arhivare jar de un arhivator obisnuit ?

R: Jar = fisiere compresate care poate contine, pe linga fisiere arhivate, si director META-INF care contine un fisier manifest.mf cu informatii despre arhiva

40. Scrieti un program care primeste ca argument de la linia de comanda un numar de puncte si afiseaza nota obtinuta conform tabelului de mai jos:

Punctaj 40-36 35-31 30-26 25-21 20-16 15-11 10-6 5-0

10 9 8 7 6 5 4 L

import java.io.\*;

class test{

public static void main(String [] s) {

int puncte;

boolean gata=false;

//ca sa ies din program dau o valoare in afara intervalului

while(!gata) {

System.out.println(“Introduceti nr de puncte: “);

System.out.flush();

try {

//citesc de la tastatura

BufferedReader in = new BufferedReader(new InputStreamReader(System.in));

String linie = in.readLine();

try {

puncte = Integer.parseInt(linie);

if (puncte < 0) gata=true;

else

if (puncte>35) System.out.println(“aveti nota 10″);

else

if (puncte<=35 && puncte >=31 ) System.out.println(“aveti nota 9″);

else

if (puncte<=30 && puncte >=26 ) System.out.println(“aveti nota 8″);

else

if (puncte<=25 && puncte >=21 ) System.out.println(“aveti nota 7″);

else

if (puncte<=20 && puncte >=16 ) System.out.println(“aveti nota 6″);

else

if (puncte<=15 && puncte >=11 ) System.out.println(“aveti nota 5″);

else

if (puncte<=10 && puncte >=6 ) System.out.println(“aveti nota 4″);

else System.out.println(“nu aveti nota!!!”);

}

catch (NumberFormatException e1) {System.out.println(“Introduceti un nr intreg!!!”); }

}

catch (IOException e2) { System.out.println(“Intrare gresita de la tastatura “+e2.toString());}

}

}}

1). Cum poate fi lansata o aplicatie Java arhivata intr-un fisier cu numele demo.jar?

R : java –jar demo.jar

2). Care este rolul interfetei Observable ?

R: Observable este clasa, iar Observer este interfata! (greseala profului)

Conform java-help, “A class can implement the Observer interface when it wants to be informed of changes in observable objects.”. Raspunsul la acesta intrebare ar putea fi: Rolul interfetei Observer este de a informa despre schimbarile dintr-un object care este observat (de un obiect din clasa Observable )(are legatura cu firele de executie)

3). Cum trebuie declarata metoda top a clasei Stack (ce descrie o stiva de obiecte) asfel incat aceasta sa returneze obiectul din varful stivei, sau sa arunce o exceptie de tipul EmptyStackException daca stiva este vida ?

R: public Object top() throws Exception {

if ( this.nrElementeDinStiva>0)

return EmptyStackException;

else

return this.Elemente[nrElementeDinStiva];

}

4). Descrieti succint o modalitate de creare a unei clase din care sa poata fi instantiate fire de executie.

R: class FirExec implements Runnable{

private fir=null;

public FirExec ()

{

if (fir == null) {

fir = new Thread(this);

fir.start();

}

}

public void run() {…}

sau direct: class FirExec extends Thread { public void run() {…}}

5). Ce este o relatie de ordine totala? Cum poate fi specificata o astfel de relatie asupra obiectelor unei clase?

R: ??! Am gasit ceva: Interfata Comparable contine o functie de comparare, care implica o ordine totala intre obiectele unei colectii (java-help: This interface imposes a total ordering on the objects of each class that implements it). Pentru a fi specificata o astfel de relatie pentru obiectele unei clase trebuie ca respectiva clasa sa implemeteze interfata Comparable (deci sa aiba si functia compareTo). Definitia unei relatie de ordine totala: (conform FAI): oricare doua obiecte din multime sunt in relatie unul cu celalalt (unul si numai unul din cazurile urmatoare are loc: a<b, a=b sau a>b)

6). Care din instantierile de mai jos este (sunt) gresita (gresite):

a)List a=new List(); b)List c=new ArrayList(); c)LinkedList b=new List(); d) AbstractList d=new LinkedList();

R: a) (List e interfata; nu poate fi instantiata); c) (din acelasi motiv);

Atentie: d) e corect: LinkedList extinde AbstractList; eu la examen am citit ArrayList in loc de AbstractList si am zis ca d) e gresit;

7). Cum pot fi folosite comentariile introduse in codul sursa al unei clase Java prin secventa /\*\* ?

R: javadoc Aplicatie.java; comentariile sunt introduse in documentatia aplicatiei

8). Care din instantierile de mai jos este (sunt) gresita (gresite):

a)String a=”xy”; b) String b=new String(”xy”);

c)String c={’x’, ’y’}; d)String d=”x”+”y”;

R: c)

9). Care este signatura completa a metodei main din clasa principala a unei aplicatii Java?

R: public static void main(String args[])

10). Ce se va afisa la executia urmatoarei secvente de cod:

RuntimeException a=new RuntimeException();

System.out.println(a.getClass().getSuperclass().getSuperclass());

R: class java.lang.Throwable (cred ca ideea era ca se apeleaza Throwable.toString() pentru afisare, ceea ce determina afisarea textului class <nume\_pachet>.<nume\_clasa>)(nu stiu daca se puncta raspunsul simplu: Throwable)

11). Scrieti secventa care scrie intr-un fisier cu numele ’test.txt’ o linie formata din sirul ’Test’.

R: try{

FileWriter fw=new FileWriter(“test.txt”);

fw.write(“Test\n”);

fw.close(); //e foarte important!

}

catch(IOException e) {}

12). Care este motivul pentru secventa de mai jos nu va fi compilata ?

interface Hello{

String sayHello(); }

class HelloImp1 implements Hello{

String sayHello() {return ”Hello”;}}

R: in clasa HelloImo1 functia trebuie declarata public (modificatorul de acces implicit in interfata pentru metode e public, iar pentru clasa e package); corect deci

public String sayHello() {return ”Hello”;}

13). Cum trebuie declarata metoda aduna a clasei Complex astfel incat sa poata fi apelata sub forma Complex c=a.aduna(b);, unde a,b sunt variabile de tip Complex ?

R: Complex aduna(Complex b){

return new Complex (this.real+b.real,this.imag+b.imag); }

14). Scrieti instructiunea care creeaza o conexiune la o baza de date inregistrata in ODBC sub numele ’dbDemo’.

R: Class.forName(“sun.jdbc.odbc.JdbcOdbcDriver”); //incarcare driver in memorie

Connection c=DriverManager.getConnection(“jdbc:odbc:geografie”);

15). Mentionati cel putin doi modificatori ai metodei parseInt din clasa Integer, tinand cont de forma in care aceasta este folosita.

R: public si static (se apeleaza cu Integer.parseInt; pentru a avea acces la ea trebuie sa fie publica; o metoda statica poate fi apelata sub forma <nume\_clasa>.<metoda>)

16). Care este mecanismul de oprire a unui fir de executie care trebuie sa ruleze pe o perioada nedeterminata, fara a folosi metoda invechita stop ?

R: folosind o variabila de terminare:

public void run() {

while ( running ) {…}

}

iar din exteriorul functiei se face running=false;

17). Scrieti secventa care afiseaza suma primelor doua argumente de la linia de comanda (presupunand ca ele exista si sunt reale).

R: public static void main (String[] args){ System.out.println(Float.parseFloat(args[0])+Float.parseFloat(args[1])); }

18). Ce se va intampla la executia urmatoarei secvente?

int test = 0, infinit=0;

try { infinit=1/0; test++; }

catch(Exception e) { test++; }

finally {test++; }

System.out.println(”test=” + test + ” infinit=” + infinit);

R: test=2 infinit=0 (la 1/0 da exceptie; infinit nu se modifica; sare la catch (deci primul test++ nu se executa) si apoi in finally (care se executa indiferent daca apar exceptii sau nu)

19). Va fi compilata secventa de mai jos ? Daca da, care este mesajul afisat la executie ?

class A {

A(int x) {System.out.println(”apel constructor A”); } }

class B extends A {

B() {System.out.println(”apel constructor B”); } }

public class C {

public static void main(String args[]) { B b = new B(); } }

R: deoarece B extends A, B() apeleaza constructorul implicit A(), dar acesta nu exista (datorita existentei deja a unui constructor in clasa A) (cel implicit e creat doar daca nu exista nici un alt constructor in clasa A).

20). Scrieti metodele getter si setter pentru variabila de tip Point cu numele location.

R: Point getLocation(); double getX(); double getY();

void setLocation(float x,float y); void setLocation(int x,int y);

void setLocation(Point p) (cred ca asta trebuia; numele location e al variabilei sau a metodelor ? Daca e al metodei, atunci se exclud metodele getX(), getY() (singurele care le-am scris la examen))

21). In urma compilarii unui fisier ce contine o clasa publica cu numele Test a rezultat, pe langa fisierul Test.class si un fisier cu numele Test$1.class. Care este explicatia acestui fapt?

R: clasa Test contine un obiect anonim dintr-o clasa al carei cod e scris imediat dupa declaratia obiectului. Vezi pb.34: codul compilat pentru obiectul de tip ActionListener va fi inclus intr-un fisier cu $1.class

22). Declarati si instantiati o matrice M cu doua linii si trei coloane cu valorile M(i,j)=10\*i+j;

R: int M[][]=new int[2][3];

for(int i=0;i<2;i++)

for(int j=0;j<3;j++)

M[i][j]=10\*i+j;

23). Fie a, b doua variabile booleene. Care este valoarea de adevar a expresiei:

!(a && b) == (!a && !b) ?

R: e valoarea de adevar a expresiei a==b; daca a si b au aceeasi valoare de adevar, valoarea expresiei este true, in caz contrar e false. (ce legatura are intrebarea asta cu java? Nu e nici o capcana; am testat cele 4 cazuri posibile pentru uplul (a,b) si rezultatul a fost conform asteptarilor)

24). Ce exceptie va genera urmatoarea secventa de cod?

java.awt.Panel p[]=new java.awt.Panel[2];

p[0].setLocation(0,0);

R: NullPointerException; p[0] trebuie initializat (p[0]=new java.awt.Panel() si apoi p[0].setLocation(0,0))

25). Ce intelegeti prin termenul portabilitate ?

R: Java este independenta de platforma de lucru (de sistemul de operare)

26). Care este diferenta intre clasele TreeMap si HashMap ?

R: In TreeMap, spre deosebire de HashMap, reprezentarea e sub forma de arbore, iar elementele sunt ordonate crescator.

27). Care va fi valoarea lui z dupa atribuirile de mai jos ?

int x=5, y=3, z=(x++ / 2 < –y ? –x : y++);

R: z=2 (x=6; y=3; 5/2=2 < –y=2)

28). Enumerati cel putin cinci pachete disponibile in JDK 1.3.

R: java.applet; java.awt; java.beans; java.io; java.lang; java.math; java.net; java.rmi; java.security; java.sql; java.text; java.util (am intrat in C:\jdk1.3\src\java)

29). Dati cel putin cinci exemple de componente AWT sau Swing.

R: Button, Canvas, Checkbox, Label, List (de exemplu)

30). Cum poate fi controlat accesul concurent al mai multor fire de executie la o resursa comuna gestionata prin intermediul unei metode ?

R: Acea metoda se declara synchronized; un singur fir de executie va putea folosi la un moment dat resursa comuna (indiferent daca apeleaza sau nu metoda respectiva)

31). Care este rolul clasei ResultSetMetaData ?

R: Prelucrarea rezultatelor dintr-un ResultSet.

32). Care sunt erorile ce vor fi detectate la compilarea fisierului A.java ce contine urmatoarele declaratii de clase:

public class A {}

public class B extends A {}

class C extends A, B {}

R: clasa C nu poate extinde doua clase (eroarea de la compilare: dupa extends A trebuie {, adica trebuie corpul clasei C)

clasa B trebuie declarata intr-un fisier separat (B.java); intr-un fisier java poate exista o singura clasa publica (cea cu main-ul)

33). Care este rolul metodei forName din clasa java.lang.Class ?

R: Conform java-help:”Returns the Class object associated with the class or interface with the given string name.”; returneaza un obiect de tip Class corespunzator clasei sau interfetei al carei nume este dat drept argument functiei.

34). Cum pot fi tratate evenimentele generate de un meniu ?

R: meniu.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {...}});

35). Care din instantierile de mai jos este (sunt) gresita (gresite):

a) int a[10]=new int[]; b) double []b=new double[10];

c) char c[]={”O”,”K”}; d) Class[]=new Class[]{String.class};

R: a) (evident), c) (tipuri incompatibile:”O” e String) si d) (nu exista String.class)

36). Care este rolul metodei toString din clasa Object?

R: returneaza o reprezentare sub forma de sir de caractere a obiectului de catre care a fost apelata

37). Dati o solutie pentru un applet sa poata scrie informatii intr-un fisier de pe masina de unde provine appletul, avand in vedere restrictiile de securitate ale appleturilor.

R: editarea unui fisier de permisiuni (vezi penultimul curs)

38). Scrieti instructiunea care deschide, dintr-un applet, pagina Web de la adresa ”http://www.infoiasi.ro”

R:try{

URL doc = new URL("http://www.infoiasi.ro");

getAppletContext().showDocument(doc);

}

catch(MalformedURLException e) {}

39). Dati o solutie pentru crearea unei clase publice care sa nu poata fi instantiata.

R: o declaram abstracta: public abstract class numeClasa {}

40). Scrieti o metoda care returneaza un vector cu primele n numere ale sirului lui Fibonacci (n este primit ca argument)

R: public int[] Fibonacci(int n) throws Exception {

int v[]=new int[n];

if(n<0)

throw new Exception();

v[0]=1;

if(1<=n)

v[1]=1;

if(2<=n)

for(int i=2;i<n;i++)

v[i]=v[i-1]+v[i-2]; return v;}

1. Care dintre variantele urmatoare este gresita si de ce ?

Varianta A :

class A

{

int x;

void x() throws RuntimeException

{

throw new Throwable();

}

}

Varianta B :

class M extends NullPointerException

{

}

class A

{

void x() throws M

{

throw new RuntimeException();

}

}

2. Care este rezultatul urmatorului cod (explicati) :

int a[]=new int[3];

a={1,2,3};

System.out.println(a);

3. Care este rezultatul urmatorului cod (explicati) :

int a[]={1,2,3};

{

char b,a;

b=a[1];

a[1]=b;

}

System.out.println(a[1]);

4. Care este rezultatul urmatorului cod (explicati) :

String a=”1 “;

StringBuffer b=new StringBuffer(“2 “);

Vector d=new Vector();

d.add(a);

d.add(b);

((String)(d.elementAt(0))).concat(“salut”);

((StringBuffer)(d.elementAt(1))).append(“salut”);

System.out.println(d.elementAt(0));

System.out.println(d.elementAt(1));

5. Care este rezultatul urmatorului cod (explicati) :

class T

{

int a;

static int a(int a)

{

a=10;

System.out.println(a);

System.out.println(this.a);

return 0;

}

public static void main(String[] a)

{

a(4);

}

}

6. Care este rezultatul urmatorului cod (explicati) :

class T

{

int a;

int r;

T(int n,int a)

{

if(n==0) r=1;

else

if(n==1) r=a;

else

{

this(n-1,a);

r=r\*a;

}

}

public String toString()

{

return “”+r;

}

}

class X

{

public static void main(String[] a)

{

System.out.println(new T(10,2));

}

}

7. Este vreo greseala in codul urmator (explicati) ?

class T

{

private int k;

T()

{

k=0;

}

T(int a)

{

k=a;

}

}

class C extends T

{

private int L;

C(int a)

{

L=a;

}

}

8. Este vreo greseala in codul urmator (explicati) ?

class T

{

T() throws RuntimeException

{

throw new RuntimeException();

}

}

12. Ce rezultat are urmatorul cod (explicati) :

class a\_class

{

static final double PI;

static

{

PI=3.14;

System.out.println(PI);

}

}

class T

{

public static void main(String[] a)

{

new a\_class();

}

}

13. Care dintre urmatoarele variante genereaza eroare :

Varianta A :

class M

{

class M1

{

static int a;

}

}

Varianta B :

class M

{

static class M1

{

static int b;

}

}

14. Este vreo eroare in codul urmator (explicati) ?

class M

{

class M1

{

int a;

}

}

class T

{

public static void main(String []a)

{

M.M1 some=new M.M1();

}

}

18. Exista vreo eroare in codul urmator ?

class T

{

final int a()

{

return 0;

}

protected int a(int b)

{

return b;

}

}

19. Care este superclasa clasei Object ?

ın Java orice clas˘a are o superclas˘a ¸si numai una. Evident, trebuie s˘a existe o except¸ie

de la aceast˘a regul˘a ¸si anume clasa care reprezint˘a r˘ad˘acina ierarhiei format˘a

de relat¸iile de mo¸stenire dintre clase. Aceasta este clasa Object.Clasa Object este ¸si superclasa implicit˘a a claselor care nu speciﬁc˘a o anumit˘a superclas˘a. Clasa Object este cea mai general˘a dintre clase, orice obiect ﬁind, direct sau indirect, descendent al acestei clase. Fiind p˘arintele tuturor, Object deﬁne¸ste ¸si implementeaz˘a comportamentul comun al tuturor celorlalte clase Java.

20. Dati un exemplu de utilizare a clasei FileDescriptor.

The **Java.io.FileDescriptor** class instances serve as an opaque handle to the underlying machine-specific structure representing an open file, an open socket, or another source or sink of bytes.Following are the important points about FileDescriptor:

* The main practical use for a file descriptor is to create a *FileInputStream* or *FileOutputStream* to contain it.

Using a FileDescriptor from getFD() and creating a FileInputStream from FileDescriptor:

**import**java.io.File;  
**import**java.io.FileDescriptor;  
**import**java.io.FileInputStream;  
**import**java.io.IOException;  
**public class**MainClass {  
  **public static void**main(String[] a) {  
    File aFile = **new**File("C:/myFile.text");  
    FileInputStream inputFile1 = **null**;   
    FileDescriptor fd = **null**;   
    **try**{  
      inputFile1 = **new**FileInputStream(aFile);  
      fd = inputFile1.getFD();   
    } **catch**(IOException e) {  
      e.printStackTrace(System.err);  
      System.exit(1);  
    }  
    FileInputStream inputFile2 = **new**FileInputStream(fd);  
  }  
}

21. Ce afiseaza urmatoarea secventa de cod :

int a=1;

int b=2;

if((0>1)&&((++a)>(–b)));

System.out.println(a+” “+b);

23. Care este rezultatul urmatorului cod (explicati) :

class C1

{

static void f()

{

System.out.println(“f1″);

}

void g()

{

System.out.println(“g1″);

}

}

class C2 extends C1

{

static void f()

{

System.out.println(“f2″);

}

void g()

{

System.out.println(“g2″);

}

}

C1 ob=new C2();

ob.f();

ob.g();

24. Este vreo eroare in codul urmator :

double b=2.3;

int c=b;

25. Este vreo eroare in codul urmator (explicati) :

int []a={1,2,3};

Object b=a;

System.out.println(b[0]+” ceva”);

byte []c=a;

System.out.println(c[0]+” ceva”);

26. Ce genereaza codul urmator (explicati) :

int x=0;

if(Double.isInfinite(2/x)) System.out.println(“Da”);

else System.out.println(“Nu”);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

27. Ce genereaza urmatorul program :

class C1

{

static void f()

{

System.out.println(“f1″);

}

void g()

{

System.out.println(“g1″);

}

}

class C2 extends C1

{

static void f()

{

System.out.println(“f2″);

}

void g()

{

System.out.println(“g2″);

}

}

C2 ob=(C2)(new C1());

ob.f();

ob.g();

a) eroare la compilare

b) exceptie la rulare

c) iesirea f2 g2

d) iesirea f1 g1

e) iesirea f1 g2

f) iesirea f2 g1

28 Este vreo eroare in codul urmator (explicati) ?

double b=2.3D;

double c=2.3d;

double e=1.2F;

float x=(float)(1.2E+1);

32. Este corecta urmatoarea constructie (explicati) ?

abstract class some

{

abstract void x();

void x(int y)

{

y=10;

}

}

33. Care dintre urmatoarele este corecta (explicati) ?

a) interface x extends java.io.Serializable, java.io.Externalizable

{

int y=2;

}

b) interface x extends Object, java.io.Externalizable

{

float b=2.3f;

}

34. Care este rezultatul urmatorului cod (explicati) :

interface x

{

int b=10;

}

class some1 implements x

{

int b=20;

}

class some2 implements x

{

}

class test

{

public static void main(String[]a)

{

some1 o1=new some1();

some2 o2=new some2();

System.out.println(o1.b+” “+o2.b);

}

}

36. Se dau :

1. fisierul some.java cu codul :

class xul

{

xul()

{

System.out.println(“Wonder …”);

}

}

2. fisierul sacx.java cu codul :

package x;

class T

{

public static void main(String[]a)

{

xul some=new xul();

}

}

3. ordinea sigura de compilare : javac some.java si apoi javac sacx.java.

Care este rezultatul ?

37. Schimband “some” cu “xul” si declarand clasa xul publica, care sunt modificarile ?

Toate interfetele de tip "listener" sunt subinterfetele lui *java.util.EventListener*. Aceasta din urma este o interfata vida.

1) c,d

10s – prefixul “s” nu exista in Java

( a se studia prefixele pentru tipuri numerice :”d”,”D”,”L”,”l”,”f”,”F”)

String goto – “goto” e cuvant cheie, var. nu pot purta numele cuvintelor cheie din limbaj

char firstOne = (char)65; – cast perfect valid, desi ne-necesar

1E2 – forma stiintifica a numarului 100 ( atentie, forma stiintifica este evaluata la double automat)

2) 4

atentie la operatorii pre/post-fixati

3) a

Java nu permite mostenirea multipla, doar implementarea mai multor interfete.

4) nu functioneaza pentru numere impare negative

atentie la definirea operatorului %:

[X/2] \* 2 + (x%2) = x ( am notat [] partea intreaga)

x%2 = x – [X/2] \* 2

exemplu: -7%2 = -7 – (-3)\*2 = -7 + 6 = -1

-1 != 1

5) d), nu putem face nicio presupunere legata de activitatea GC-ului, nu stim cand va dealoca memoria

6) Se va intra intr-o bucla infinita, cauzata de crearea continua de obiecte de tip Test.

de vina este campul privat Test a.

La crearea obiectului t din main(), se va crea o noua instanta a clasei Test insa inainte de apelul constructoului, se va initializa membrul “a” samd.

7) Scriem o clasa care implementeaza interfata Comparator. In cadrul acesteia, folosim o variabila membra care va fi verificata in metoda compare si se va decide daca se vor compara valorile a doua facturi ori datele la care au fost emise.

8) O adnotare pentru o metoda.

Exemplu

@Incorrect

public int sum(int a, int b){

return a+b+7;

}

9) Varianta 1 este cea corecta

Se discuta intai lipsa clasei Test din fisierul Test.java

Regula este urmatoarea: DACA fisierul Test.java contine o clasa publica, ea trebuie sa se numeasca Test.

(atentie, un fisier java poate contine MAXIM o clasa publica)

Compilatorul nu se plange, metoda() fiind deja implementata in clasa A.

10) Eroare de sintaxa

operatorul “+” nu este supraincarcat “din fabrica” pentru tipul BigInteger .

Atentie, in Java supraincarcarea operatilor NU exista.

Operatorii aritmetici pot fi folositi doar pe tipuri primitive cu EXCEPTIA claselor wrapper Integer, Float, Double, Character.

11)

try{

throw new Exception();

}

catch(Exception ex){

throw ex;

}

12) public Alg load(String clazz) throws Exception{

return (Alg)Class.forName(clazz).newInstance();

}

13) Nu se poate afla direct aceasta informatie deoarece un obiect de tipul ResultSet nu este decat un pointer catre informatii care nici macar nu au fost aduse din baza de date.

int count = 0;

while(rs.next()){

count++;

}

Atentie. Subiectul mai poate apare si sub forma: cum putem afla numarul de coloane? Pentru aceasta,vom folosi clasa ResultSetMetaData:

ResultSetMetaData md = rs.getMetaData();

md.getColumnCount() – ne da nr. de coloane

14) Se include in classPath parser.jar

15) 2

16) fluxurile I/O standard din java.io.\*

17) d), prin eliminare

Nu se intampla nimic la runtime, dpv. sintactic oricum nu s-ar putea face o astfel de verificare la compilare.

18)

public static HashMap metoda(char[][] matrix){

HashMap result = new HashMap();

for (int i=0; i < matrix.length;i++){

for (int j =0; j < matrix[0].length;j++){

Integer count = result.get(matrix[i][j]);

if ( count == null){

result.put(matrix[i][j], 1);

}

else{

result.put(matrix[i][j], count + 1);

}

}

}

return result;

}

public static void main(String[] Integer){

char[][] matrix = new char[2][2];

matrix[0][0] = 'a';

matrix[0][1] = 'b';

matrix[1][0] = 'c';

matrix[1][1] = 'b';

HashMap map = metoda(matrix);

for ( Character c : map.keySet()){

System.out.println(c + ” – ” + map.get(c));

}

}