

Testul G1S1 si G1S2

1. Sa se scrie o functie care primeste ca parametru numele unui fisier (String) si returneaza o lista de Object, continand obiectele citite din fisier. Obiectele au fost salvate in fisier unul cate unul prin metoda standard de serializare a Java. Pentru a marca sfarsitul de fisier a fost scris un obiect de tipul "Terminator"

Aici ma interesa sa vad: deschidere fisier, objectInputStream.readObject while (!obiectCitit instanceof Terminator) add in lista si inchidere stream (pentru ca noi l-am deschis)

Solutie model:(Mihailescu Razvan)

```
public ArrayList citire(String file){
    ArrayList Obiecte = new ArrayList<>();
    try (ObjectInputStream in = new ObjectInputStream(new FileInputStream(new File(file)))){
        Object obj=in.readObject();
        while(! (obj instanceof Terminator)){
            Obiecte.add(obj);
            obj=in.readObject();
        }
    } catch (IOException | ClassNotFoundException e) {
        e.printStackTrace();
    }

    return Obiecte;
}
```

2. Se da codul din imagine. Bifati toate afirmatiile adevarate.

```
1 public Integer SUMA(Integer rezultat, int a, int b)
2 {
3     rezultat = a + b;
4 }
```

- o Java nu suporta denumiri de functie cu toate literele mari. | *No comment*
- o Avem erori de compilare deoarece rezultat este de tipul Integer iar a si b sunt int | *Nu e o problema (autoboxing)*
- o Functia este inutila si nu va avea niciun efect | *Corect! Parametrii trimisi prin valoare nu prin adresa!*
- o Functia trebuie sa returneze Integer dar nu returneaza nimic | *Corect! Lipseste return.*

Am punctat partial daca ati bifat doar 3 sau doar 4. Daca ati bifat orice altceva in afara de 3 sau 4 nu am punctat deloc.

3. Se da functia public List filtru(List lista, Predicate tester) care filtreaza lista primita ca parametru si returneaza rezultatul (un element este e

Aici ma interesa sa vad: lambda expresie, asignarea rezultatului la myList.

Solutie model:(Popa Remus-Adrian)

```
myList = filtru(myList, str->str.length()<6);
```

4. In urmatoarea bucata de cod:

```
1 public static void Main(String args[])
2 {
3     System.out.println("TEST PAO");
4 }
```

- o Avem erori de compilare deoarece Main este scris cu litera mare | *In niciun caz erori de compilare*
- o Avem erori la runtime deoarece Main este scris cu litera mare | *Se putea intelege ca daca incercam sa rulam aceasta clasa nu putem sa o rulam*
- o Apar exceptii la runtime deoarece Main este scris cu litera mare si Java este case-sensitive | *Asta era aproape identic cu ce e mai sus. Am pus*
- o Avem erori la compilare deoarece parametrul este declarat gresit, ar trebui sa fie String[] args | *No comment...si cineva chiar a bifat asta.*

Am fost foarte categoric la acest subiect.

5. Scrieti o metoda care primeste ca parametru un nume de fisier si doua numere max si count. Metoda deschide apoi acest fisier, scrie (ca text) "count

Aici ma interesa sa vad: Deschidere fisier, inchidere fisier (try with resources), generare numere si atentie la ultimul numar.

Solutie model:(Radu Ioan-Alexandru)

```
public static void funct2(String f, int max, int count) {
    try (PrintWriter out = new PrintWriter(new File(f))) {
        Random rand = new Random();
        for (int i = 0; i < count - 1; i++) {
            out.print(rand.nextInt(max));
            out.print(", ");
        }
        out.print(rand.nextInt(max));
    } catch (FileNotFoundException ex) {
        Logger.getLogger(Test.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

Desi exceptiile nu ar trebui tratate aici pentru ca ar lasa impresia ca "ne-am facut treaba" nu am scazut puncte pentru asta (la nimeni)

6. Bifati toate afirmatiile adevarate despre codul de mai jos:

```
1 public class Balena extends Mamifer, Animal implements Vietate, Entitate
```

- O clasa nu poate fi declarata publica, toate clasele sunt implicit publice | *No comment, nimeni nu a bifat asta*
- Balena nu poate fi de tipul Mamifer si Animal in acelasi timp | *Singura varianta corecta era aceasta.*
- Balena nu poate fi de tipul Vietate si Entitate in acelasi timp | *Total gresit, o clasa poate implementa oricate interfete (s-ar putea sa gasii*
- Avem erori de compilare. Mostenirea multipla se declara altfel in Java | *Nu exista mostenire multipla in Java. Daca ati bifat asta si ati bifat*

7. Scrieti o functie care deschide un Socket (client) la o adresa si un port primeste ca parametru si care salveaza in doua variabile de instanta ale cl
Aici ma intereseaza sa vad (doar): deschidem socket, salvam streamuri si throws exceptii, atat!

Solutie model:(Botolan Vlad-Alexandru)

```
public void deschideSocket(int port,String adresa){
    Socket s=null;
    try{
        s = new Socket(adresa,port);
        in = new ObjectInputStream(s.getInputStream());
        out = new ObjectOutputStream(s.getOutputStream());
    }
    catch(IOException e){
        if(s.isBound()){in.close();out.close();}
    }
}
```

Pentru cei care ati declarat variabile locale in/out->0 puncte. pentru cei care ati inchis socketul...ce rost mai are metoda? Nu am depunctat faptul

8. Se da urmatoarea bucata de cod (... semnifica existenta altor metode care nu sunt specificate). Bifati toate afirmatiile care sunt adevarate:

```
1 import java.io.IOException;
2 import java.io.InputStream;
3 import java.net.Socket;
4
5 public class TestSync
6 {
7
8     private Socket socket;
9
10    public int citesteByte() throws IOException
11    {
12        InputStream in = socket.getInputStream();
13        return in.read();
14    }
15
16    ...
17
18 }
```

- Metoda citesteByte este gresita pentru ca nu inchide input streamul si socketul din care citeste | *Nu ea l-a deschis, nu ea trebuie sa il inchid*
- Metoda ar trebui sa trateze intern exceptia IOException | *Nu, pentru ca da o trateaza ar trebui sa returneze un rezultat care ar fi hardcodat/*
- Codul pare ok | *Corect*
- Avem eroare de compilare deoarece functia se numeste citeste Byte dar returneaza un int | *Din fericire nimeni nu a bifat asta*
- Adaugam synchronized la campul lista | *Nu exista asa ceva*

Explicatiile de mai sus sunt suficiente

9. Se da urmatoarea bucata de cod. Cum va puteti asigura ca nu apar erori de sincronizare in aplicatia care foloseste aceasta clasa? Daca nici una din

```
1 import java.util.List;
2
3 public class TestSync
4 {
5
6     private List<Object> lista;
7
8     public Object getElement(int pozitie)
9     {
10        return lista.get(pozitie);
11    }
12
13    public void addElement(Object element)
14    {
15        lista.add(element);
16    }
17
18    public List<Object> getLista()
19    {
20        return lista;
21    }
22
23 }
```

- Adaugam synchronized la toate metodele din functie | *Nu ajuta. (vedeti la sfarsit motivul)*
- Adaugam synchronized la metodele getElement si addElement si synchronized(Test.class) in ultima metoda | *Nu ajuta*
- Facem toate metodele statice si obiectul lista static apoi adaugam synchronized | *Nu ajuta*

- Facem metoda getListă privată și adăugăm synchronized la getElement și addElement | Corect
- Adăugăm synchronized la câmpul listă | Nu există așa ceva
- Other | Am văzut că se poate spune că din cauza metodei getListă nu se poate sincroniza, dacă vi se pare anormal să o faceți privată (sau în

Ceea ce ne încurcă în primul rând era acel getListă care practic oferea oricui acces la lista noastră deci nu aveam cum să ne asigurăm că cine cheamă

10. Scrieți o metodă "thread safe" care incrementează un câmp numit nr, de tipul int, de x ori unde x e primit ca parametru. După fiecare incrementare

Aici mă interesează să văd: synchronized pe ce trebuie (erau mai multe moduri de a face synchronized), wait în loc de sleep PE OBIECTUL PE CARE AM FĂCUT

Soluție model: (Radu Ioan-Alexandru)

```
public void funct4(int x) {
    try {
        for (int i = 0; i < x; i++) {
            synchronized (this) {
                ++nr;
            }
            Thread.sleep(100);
        }
    } catch (InterruptedException ex) {
        Logger.getLogger(Test.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

Sleep în synchronized, interzis. wait fără synchronized(this), interzis. Mulți v-ați complicat fără rost.

11. Într-o aplicație web avem nevoie de o pagină în care să se afișeze mesajul "Acces nepermis" utilizatorilor care nu au în sesiunea lor salvată valoare

Aici mă interesează să văd: Boolean val = session.getAttribute(...) și apoi folosirea acestui val. Era copy paste din laboratorul 9 din care mai ștergeți

Soluție model: (Eva Ștefan)

```
<%
    Boolean currentUser=(Boolean) session.getAttribute(admin);
    if(currentUser==false)
    {
%>
<h1>Acces nepermis</h1>
<%
    }
    else
    {
%>
<h1>Acces permis</h1>
```

Era copy paste! Mergea și un null check acolo dar nu am depunctat pentru asta. Cei care ați folosit JSTL...dacă ați fi folosit corect ați fi fost pu

12. Ce este greșit în codul de mai jos? Scrieți toate greșelile pe care le observați.

```
4      public Connection connection;
5
6      public void testJDBC() throws SQLException
7      {
8          PreparedStatement st = connection.prepareStatement("SELECT * FROM users WHERE password=?");
9          ResultSet rs = st.executeQuery();
10         while(rs.next())
11         {
12             System.out.println(rs.getString("username"));
13         }
14     }
```

Trei lucruri:

- Nu se închide statement
- Nu se închide resultset
- Nu este setat parametrul din query

NU sunt greșeli:

- Nu se închide Connection (nu noi am deschis-o nu noi o închidem)
- Aruncăm excepția mai sus (dacă nu reușim să facem treaba, cel care ne-a apelat trebuie să știe cumva!)

13. Se dau clasele din imagine. Ce trebuie (obligatoriu) să facem ca să nu mai avem probleme?

```

public class ClasaA
{
    public int nr;
}

public class ClasaB extends ClasaA implements Serializable
{
    public String nr;
}

public class ClasaC extends ClasaB
{
    public Date date;
}

public class Date
{
}

```

- Sa marcam ClasaA ca implementeaza Serializable | *Nu. Clasa A nu e serizalizabila si nici nu e nevoie sa fie*
- Sa marcam clasa Date ca implementeaza Serializable | *Corect! Date apare intr-o clasa Serializable, deci trebuie sa fie Serializable*
- Sa marcam ClasaC ca implementeaza Serializable | *Nu. C extends B care implements Serializable, deci e Serializable si ea.*
- Nu este nevoie sa facem nimic, nu avem probleme | *Date trebuie marcata Serializable*
- Avand in vedere ierarhia claselor, nu putem face nimic pentru a rezolva problemele de serializare | *Ba da, vezi mai sus*

Desi nu este recomandata mostenirea claselor Serializable, trebuie sa fiti in permanenta constienti de modul in care Interfetele sunt "mostenite". *^*

Foarte slabe rezultate la acest test. Admit si faptul ca e posibil sa fi fost cel mai dificil, asa mi-au spus mai multi studenti.