**Examen teoretic la SD (restanta 2020)**

1. **Explicati controlul invers.**

Controlul invers este fenomenul prin care programatorul pierde o buna parte din controlul aplicatiei atunci cand utilizeaza framework-uri. Acesta apeleaza metode simple, de multe ori cu o granularitate grosiera, ale framework-urilor in vederea rezolvarii unor task-uri. Astfel, programatorul cedeaza o parte din controlul sau framework-ului (astfel aparand “controlul invers”)

1. **Spark poate fi folosit doar pe date structurate/tabelare sau doar pe formate nestructurate? Explicati.**

In laboratoarele 13 si 14 am vazut Spark folosit pe data tabelare (cu Spark SQL) si structurate (de exemplu pe liniile unui fisier text unde pe fiecare linie este serializat cate un obiect). Daca trebuie sa aleg una din cele 2 variante, atunci o aleg pe prima. Totusi, cred ca Spark s-ar putea folosi si pe date nestructurate: daca am folosi Spark in Python si am procesa un flux de string-uri care contin obiecte (de tipuri diferite) serializate (de exemplu in formatul JSON). Pyton ar permite deserializarea acestora (fiecare obiect intr-un dict[str, object]) si procesarea acestora (intr-o anumita masura deoarece functiile de tip map, flat\_map etc trebuie sa tina aiba grija la tipul, adica atributele, obiectului).

1. **Prezentati ciclul de viata al unei fasole sesiune cu stare.**

Initial, fasolea nu exista (fiind in starea “Doesn’t Exist”). Ciclul fasolei este initiat de client, atunci cand obtine o referinta la acesta. Containerul injecteaza dependentele fasolei, apeleaza metodele adnotate cu @PostConstruct iar fasolea devine gata de utilizare (intrand in starea “Ready”). Apoi, in timpul in care fasolea este in starea “Ready” containterul se poate decide sa o pasivizeze (trecand-o in starea “Passive”). La final, cand clientul nu mai are nevoie de ea, apeleaza metoda Remove. Containerul reactioneaza la acest eveniment apeland metodele @PreDestroy si permitand Garbage Collector-ului sa elibereze memoria ocupata de aceasta.

1. **Cate forme de injectare a dependentelor avem in Spring si prin ce difera fata de Oracle Sun?**

Tipuri:

* constructor based: cand scrii:

class MyClass(private val myService: MyServiceInterface) // Fara nici o adnotare speciala

* setter based: cand scrii:

@Autowired

private lateinit var myService: MyServiceInterface

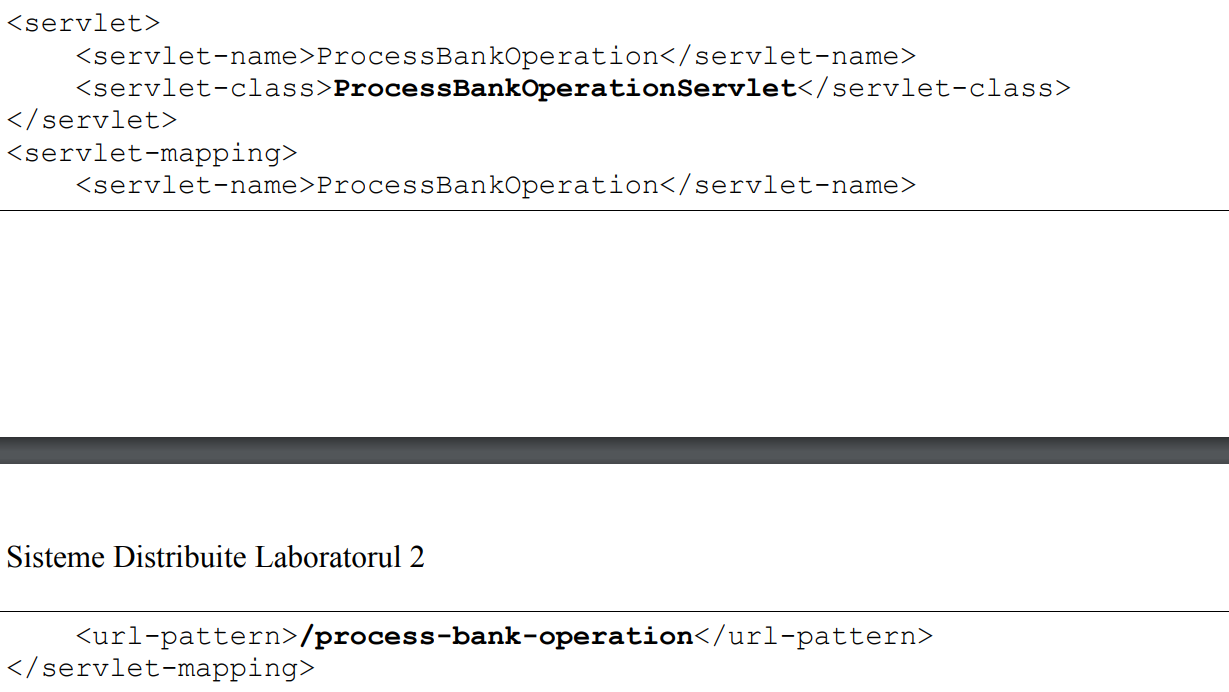
1. **Cum este plasata din punct de vedere al arhitecturii securitatea in Spring?**

(Am gasit pe net, cine stie ce se dorea a se raspunde la intrebare defapt…)

Intre client si Servlet (controllerul din MVC) poate exista un lant de filtre care sa asigure securitatea. [DETALII](https://docs.spring.io/spring-security/reference/servlet/architecture.html)

1. **Care este rolul fisierului web.xml in abordarea JEE?**

In fisierul web.xml se declara servletii (pentru a deveni eligibil pentru procesul de mapare la un URL) si pentru maparea acestora la anumite URL-uri.



1. **Care este relatia afaceri-infrastructura suport?**

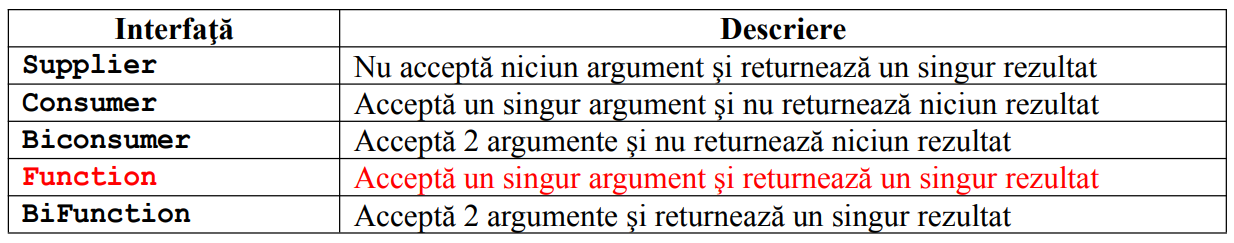
(Raspuns improvizat, nu cred ca e ok)

Infrastructura suport ar trebui sa urmeze gandirea afacerii, astfel infrastructura poate fi proiectata tinand de afacere si pentru a se putea dezvolta ambele in acelasi timp.

1. **De ce este contestat SOA din 2013?**
2. **Explicati problema dependentelor organizationale.**
3. **Ce este anuntarea incetarii functionarii si la ce ne trebuie?**

Anuntarea incetarii functionarii este un model de proiectare specific SOA care presupune ca furnizorii de servicii sa anunte din timp incetarea functionarii serviciilor ce urmeaza a fi desfiintate pentru ca acei clienti ce depindeau de ele sa poata sa gaseasca altele.

1. **In abordarea propusa de Micronaut pentru interfata Function, este posibila trimiterea a 100 de parametri functiei?**



Interfata function are o metoda ce trebuie suprascrisa (implementata) care contine un singur parametru deci teoretic nu se pot trimite 100 de parametri. Totusi, acel unic parametru poate sa fie un obiect (nu neaparat unul primitiv) si poate incapsula acei 100 de parametri deci, cumva, s-ar putea trimite.

1. **Explicati virtualizarea hibrida**

Virtualizarea hibrida este o combinatie dintre virtualizarea completa si paravirtualizare.

1. **Explicati intermediarul invers & atasamentul**

Tratarea unei cereri se realizeaza in urmatorii pasi: cererea vine la Load Balancer (ex: Kubernetes). Acesta verifica daca raspunsul pentru ea este in cache. Cache-ul este intr-un container de pe aceeasi masina (deci latenta de accesare este mai mica, descoperirea simpla si disponibilitatea exact aceeasi ca a LB-ului). Daca se gaseste raspuns pentru cerere in cache atunci acesta este returnat (HIT). Daca nu se gaseste atunci cererea este inaintata spre POD-ul destinatie (MISS). POD-ul trateaza cererea (depunand un efort computational superior gasirii raspunsului in cache) si returneaza raspunsul. LB-ul, pe langa ca returneaza inainteaza raspunsul de la POD la client, stocheaza raspunsul in cache.

1. **Ce este o platforma hiperconvergenta?**

O platforma hiperconvergenta este o infrastructura definita de software ce virtualizeaza toate componentele unui sistem de calcul conventional (memorie, procesare, stocare de date, acces la retea).

1. **Care sunt problemele JEE in interactiunea cu containerele?**

(Nu stiu dar pot sa spun ceva problema a containerelor JEE)

Problema containerelor JEE este ca sunt masive si nu sunt flexibile.

(Alte probleme ale JEE daca trebuie umplut spatiul)

* Influentata de gandirea clasica in sisteme distribuite
* Modelul de thread-uri nu aduce niciun beneficiu
* Framework-ul nu suporta nativ abordarile bazate pe data streams
* Nu suporta rezilienta
* Implementata in containere masive
* Fara suport pentru imutabilitate
* Limitari in alegerea persistentei
* ESB-ul este limitativ

1. **La ce se refera “proiectat pentru schimbare”?**

(Nu stiu sigur)

Proiectat pentru schimbare se refera la proiectarea serviciilor din cadrul SOA cu o granularitate suficient de fina astfel incat sa poata fi schimbate.

1. **Explicati rolul functiei de fitness in nor?**

Este o functie care sumarizeaza cat de aproape e aplicatia de telurile alese. Se aleg cateva teluri si se urmareste echilibrarea lor.

1. **Cum a aparut notiunea de microserviciu?**

Initial, microserviciile se numeau microaplicatii. Acest termen provoca neintelegeri si a fost inlocuit de termenul microservicii.

1. **Ce este ascunderea informatiilor in cazul microserviciilor?**

Ascunderea informatiilor reprezinta ascunderea datelor si logicii de afaceri de exterior prin prezentarea doar a functionalitatilor publice prin intermediul contractului.

1. **Ce format au entitatile in etapa de mapare in abordarile MapReduce/Hadoop? Dar in etapa de reducere?**

In etapa de map entitati in formatul (key1, val1) se transforma in entitati in formatul (key2, val2).

In etapa de reducere entitati in formatul (key, [val2 list]) se transforma in formatul [val3 list]

1. **La ce se utilizeaza TOGAF?**

Standardul TOGAF se utilizeaza la proiectarea arhitecturilor software pentru intreprinderi.

1. **Ce avem pe axa z in cubul scalarii microserviciilor?**

Avem descompunerea functionala. Aceasta seamana cu axa x (cu descompunerea orizontala), diferenta fiind faptul ca se distribuie mesajele pe baza subiectelor (topics) acestora, uneori fiind relevanta si geolocatia.

1. **Care sunt spatiile de lucru utilizate de masina Docker?**

NET, PID, IPC, MNT, UTS

1. **Cum se poate realiza multiplicarea unui serviciu in Docker? (comanda de ex)**

docker compose up --scale nume\_serviciu=numar\_instante

1. **Ce probleme au containerele in nor?**
2. **Explicati descompunerea urmand regulile de afaceri pentru MSA?**

Este procesul prin care o aplicatie cu granularitate mare este descompusa in microservicii urmand principiile de proiectare ale MSA. Aceasta se poate face folosind modelul de proiectare sugrumator (strangler) care presupune extragerea din modulele din aplicatia originala si transformarea acestora in microservicii si introducerea a cate un proxy pentru noile microservicii

1. **Ce este impartirea responsabilitatilor in crearea unei interogari?**

Este un model de proiectare specific MSA. In cadrul interogarilor complexe, mai multe microservicii au cate o responsabilitate, aceasta constand in efectuarea unui pas in vederea realizarii interogarii.

1. **Prezentati modelul de proiectare urmarire distribuita.**

Modelul de proiectare urmarire distribuita este un model de proiectare specific MSA. Acesta presupune crearea unor microservicii ce au ca responsabilitate urmarirea altor microservicii. Fiecarui microserviciu urmarit i se atribuie un ID unic. Noile microservicii alcatuiesc stratul de instrumentatie.

1. **Definiti notiunea de eveniment complex in nor.**

Este un eveniment care, pe langa tipul sau, incapsuleaza si alte alte atribute (precum antetele mesajelor de la Kafka). Cel mai important atribut al unui eveniment complex, la proiectare si utilizare, este comportamentul sau.

1. **Care sunt cele 3 dimensiuni ale modelului CAP si ce inseamna fiecare?**

**C**onsistency (consistenta) = faptul ca se lucreaza mereu cu ultima versiune a bazei de date

**A**vailability (disponibilitate) = tranzactiile au o latenta scazuta

**P**artitioning (partitionare) = baza de date are o redundanta a datelor crescuta, in cazul in care se pierd mesaje aceasta continua sa functioneze astfel avand o toleranta la erori