**VAVA P#8**

🡺 **Odovzdávať budeme 10. alebo 11. mája**

🡺 PDFko s GITlogom – datum, meno, comment z commitu; odovzdat vsetky projekty, zdrojaky, konfiguraky; žiadne classfily ANI KNIZNICE; zip sa bude volat puskas\_radosky.zip

🡺 Prezentujeme vo dvojici – ukazeme appku, zdrojaky, pokecame si 😊 – dohodneme si, ako to rozdelime

🡺 DOK – netreba generovat Javadoc, ale MAJU TAM BYT JAVADOC KOMENTY

🡺 Napisat si preferenciu, ci chceme prezentovat v stredu alebo stvrtok

🡺 Prezentovat budeme max 10 min

🡺 Staci to mat lokalne v Eclipse, ale my to spravime

🡺 Viacvrstovy model – server – klient

🡺 Session beany – kratko zijuce, **nezdielane** – nemaju garantovane prezitie neocakavanych situacii

**ENTITY BEANS**

🡺 API

🡺 Lookup – komunikacia s beanou

🡺 Dependency injection – viem z 1 beany zavolat druhu

@EJB – nemusim ani pouzit metodu lookup()

🡺 Enterprise archive EAR – napr. ked potrebujem na server dostat kniznicu pri generovani QR kodu – urcim tym, kt. beanu chcem zavolat

🡺 Vdaka tomu viem v klientovi mat len interface jeden a vsetko nad tym bezi (cisty kod)

🡺 Tj ked budeme pouzivat kniznicu pre pdf/html/qr kod – **treba to na server dostat cez ten ear!!!**

**🡺 TRANSAKCIE – ACID (atomicka, konzizstetná, izolvoaná, data tam do dalsej tr. budu)**

🡺 Transakcie – CONTAINER & BEAN MANAGED

🡺 Container – automaticky spracovav transakcie – automaticky commitne, ak vsetko ide ako ma, inak exception a rollback transakcie

🡺 BACHA – napr. sa mi moze rollbacknut logovani, co nechcem – transakčné atribúty (Required...)

🡺 **RequiresNew** – transakcia bude v kazdom pripade (aj pri chybe)

🡺 Session beany sa mozu volat aj navzajom

@TransactionAttribte(vybrat\_z\_ponuky)

🡺 Ale štandardne také niečo nepotrebujem – ked tak len RequiresNew

🡺 Concurrency control – pessimistic & optimistic

🡺 Pessimistic – stopercentné, ale velky overhead

🡺 Optimistic – pri commitovani zmien sa kontroluju verzie zaznamov – ak je poziadavka na zmenu z nizsej verzie, neumozni mu to -🡺 DEFAULT v JPA

🡺 DISTRIBUOVANE TRANSAKCIE – dvojfazovy commit – az ked vsetci potvrdia

**JPA – JAVA Persistence API**

🡺 Entity beany su dlhozijuce – preziju restart (lebo DB)

🡺 Implementacia cez XML alebo anotacie

🡺 POJO classy

🡺 Podobne ako triggere v DB

🡺 Persistence context – medzi DB a pamatou

🡺 Konfiguracia vo WildFly – standalone.XML

🡺 NamedQuery, NativeQuery – obchadzam OR mapovanie

🡺 Vediet rozdiel medzi session a entity bean, coho su sucastou, ACID, optimisticke a peimisticke logovaie, distribuovane transkacie poznat