Štýly

https://docs.google.com/document/d/1tvY-lT7UvyimHhYdf0n4L3_jp6M2WLB4lY4MDEV6t40/edit#heading=h.d160mye1ekfn

Vzory

https://docs.google.com/document/d/1AXq5YKsH6VUn9H7Bf5SDBg-8DuSL2RvJpSZbJ-xGQ5M/edit?fbclid=IwAR0EHzmziQhuAxbM_VQtQ_yPnFyJvKdLVlrPhK-TVYvB3S_RcLREZYR6d14#heading=h.krw8coo2n3t9

Ostatné

J2EE

- základným mechanizmom, na ktorom stavia sú **transakcie**
- prenos zložitých implementačných konceptov na ramená kontaintera a resource adapterov:
    - distribuovanosť
    - transakčnosť
    - multithreading
    - deklaratívna bezpečnosť
    - resource management
- vývojár sa môže zamerať len na business logiku
- vývojár uvádza niektorý z konceptov deklaratívnym spôsobom
- oddeľuje role na bean provider, assembler a deployer
- rozdeľuje problematiku na viac vrstiev

EJB

- server side komponenty
- **multithreading:** thready striktne managuje kontainer -> nemožnosť vytvoriť thread znemožňuje asynchrónnosť a background processing, riešenie: msgDrivenBean, timerBean
- Existuju tieto beany: statless, statefull, entity, message driven, timer
- dva typy výnimiek:
    - checked
        - aplikačné výnimky
        - odvodené od java.lang.Exception
        - musia sa deklarovať v throws klauzule
        - klient môže zareagovať
    - unchecked
        - systémové výnimky
        - odvodené od java.lang.RuntimeException
        - považované za bug
        - kontainer robí automaticky rollback transakcie
- beany bežia v kontajneri
- beany podporujú dva typy spravovania transakcií

    - container-managed transactions
        - container štartuje transakciu pred volaním metódy a commituje ju po ukončené volania
        - to, ako sa má transakcia vytvoriť, definujeme transakčnými atribútmi na úrovni metódy
        - developer nevolá commit a nemôže zavolať rollback, namiesto rollback-u môže vyhodiť unchecked výnimku alebo hlasovať za rollback hlasovaním setRollbackOnly
    - bean-managed transaction
        - **JDBC transakcie** asi len ak treba wrappnúť starý Java kód, používa transaction manager priamo v ovládači pre danú DB
        - **JTA transakcie** sú v Java Transaction API, v J2EE je implementovaný transaction provider, sú nezávislé od ovládača pre danú DB
        - vývojár sa stará o begin, commit, rollback
        - nesmie volať setRollbackOnly / getRollbackOnly
        - pred ukončením volania musí stateless bean resolvnúť transakciu

Stateless

- po ukončení volania nezostáva v aplikačnom serveri žiaden údaj súvisiaci s transakciou
- každé volanie je samostatnou transakciou, po ukončení volania je DB v konzistentnom stave (transakcia je buď commitnuta alebo rollbacknuta)
- viackrokový proces je riešený ako sekvencia transakcií
- volanie je možné vykonať na ľubovoľnom serveri (ľahké clustrovanie)
- webová služba je stateless z definície
- nevýhoda: údaje, s ktorými sa pracuje treba vždy nanovo získať z DB, kedže stateless

Statefull

- stav medzi volaniami je udržiavaný v pamäti J2EE servera
- server musí riešiť afinitu, session management
- vyššia réžia
- vývojár musí riešiť:
    - možnosť vzniku kolízií pri dlhotrvajúcich transakciách
    - vyššiu pamäťovú náročnosť
    - nekompatibilita s webovými službami, keďže tie sú stateless
- výhoda: pri opakovanom spojení netreba spojenie a dáta znovu získavat

Mechanizmus transakcií

- buď celá transakcia skončí úspechom, alebo celá failne
- nemôže sa transakcia vykonať len čiastočne, nastáva rollback pri zlyhaní
- takýto prístup zjednodušuje error recovery
- zachovávajú integritu systému
- úspešne dokončená (commitnutá) transakcia sa nedá rollbacknúť
- rollbacku využíva uložený image stavu pred danou transakciou
- ACID (Atomicity, Consistency, Isolation, Durability)
- pre distribuované transakcie sa používa dvojfázový commit

- výhody:
    - zdieľanie zdrojov medzi používateľmi
    - posúva výpočet transakcií na čas, kedy sú zdroje menej vyťažené
    - menšia potreba ľudskej interakcie
    - vyššie využitie výpočtových zdrojov

Multithreading

- procesy a vlákna
- multihtreading – jeden proces má viac vlákien, tzn. viac vlákien v tom istom adresovom priestore
- využitie napr. pri serveroch, že pre každého klienta sa vytvorí jedno vlákno
- výhoda: rýchlejšie vykonanie, ostatné vlákna môžu lepšie využiť cache pamäť
- nevýhoda: vlákna môžu navzájom rušiť pri zdieľaní HW prostriedkov, synchronizácia

.NET

- Je to platforma pre XML Web services od Microsoft-u
- beží na OS od Microsoft-u
- Zjednotil viacero vývojárskych prostredí do jedného
- podoruje jazyky: C#, C++ .NET, Visual Basic .NET, JScript .NET, J#
- kompiluje jazyky do CIL (Common Intermediate Language) , tiež označovaného ako MSIL (Microsoft Intermediate Language), ktorý je CPU independent
- MSIL sa kompiluje na CPU dependent machine code pomocou JIT (Just in Time) kompilátora