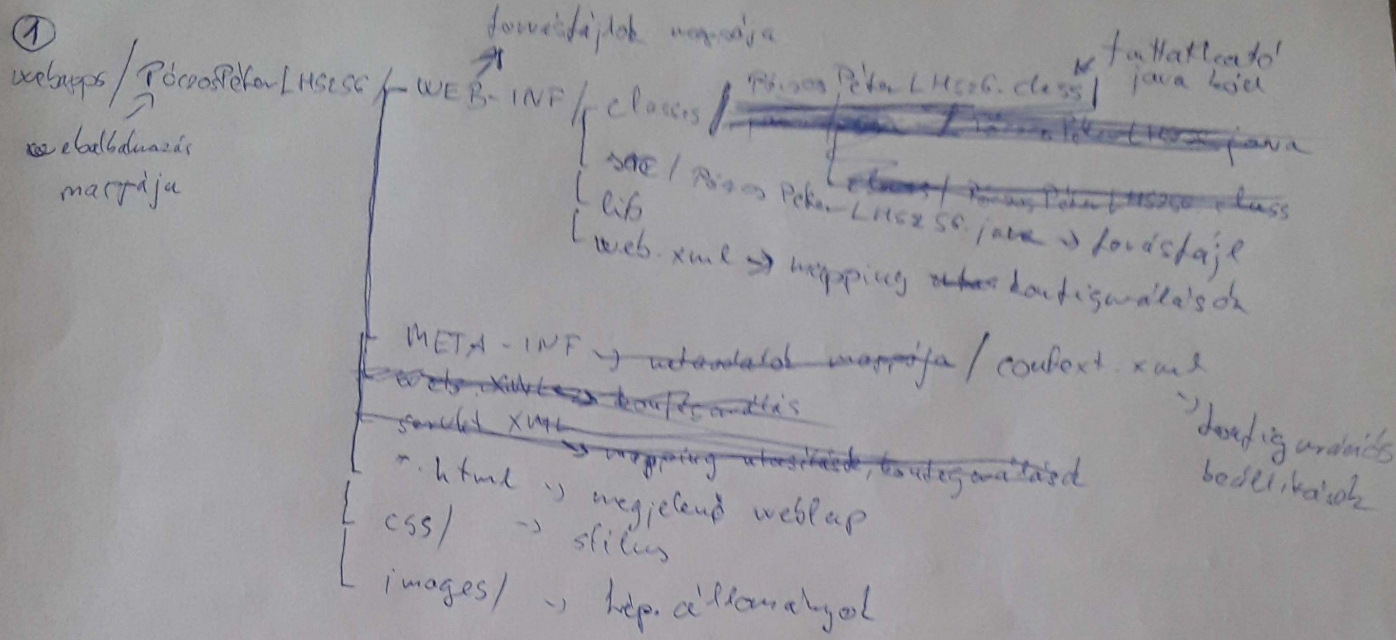
Pósos Péter
LHS2SG
2021.01.14



web.xml -> szervlet mapping létrejöttéhez

PósosPéterLHS2SG.java:

import javax.servlet.*;

@WebServlet(name = "", UrlPatterns = {"/home/"})

Public class PósosPéterLHS2SG extends HttpServlet {

~~PrintWriter out = response.getWriter();~~

~~try{~~

~~out.println("<html><head>");~~

~~out.println("</head><body> PósosPéter </body></html>");~~

@Override

public void DoGet(HttpServletRequest request, HttpServletResponse response) {

PrintWriter bi = response.getWriter();

try { out.println("<html><head></head><body> PósosPéter
</body></html>");

}

finally{

out.close();

}

Tóth Péter
LHISEC
2021.01.14.

(2) 1. Lokális verziókövető rendszerek
- Lokális tár
- fájl alapú példa: SCSS, RCS

2. Centralizált verziókövető rendszerek
- több fejlesztő, párhuzamosság
- fájl alapú
- merge before commit példa: CVS, ~~SVN~~ SVN

3. Elosztott verziókövető rendszerek
- decentralizált
- minden klines rendelkezik a teljes tárolóval
- verziókezelés lokálisan történik
- kommunikáció: peer to peer
- commit before merge

példa: Mercurial, Git, Bazaar

Parancsok:

git config -> repository configurálása
git status -> állapotlekérdezés
git add . -> állományok hozzáadása } helyi tárolás
git commit -> commitolás
git push -> távoli tárolóval szinkronizálás
git pull -> - " -

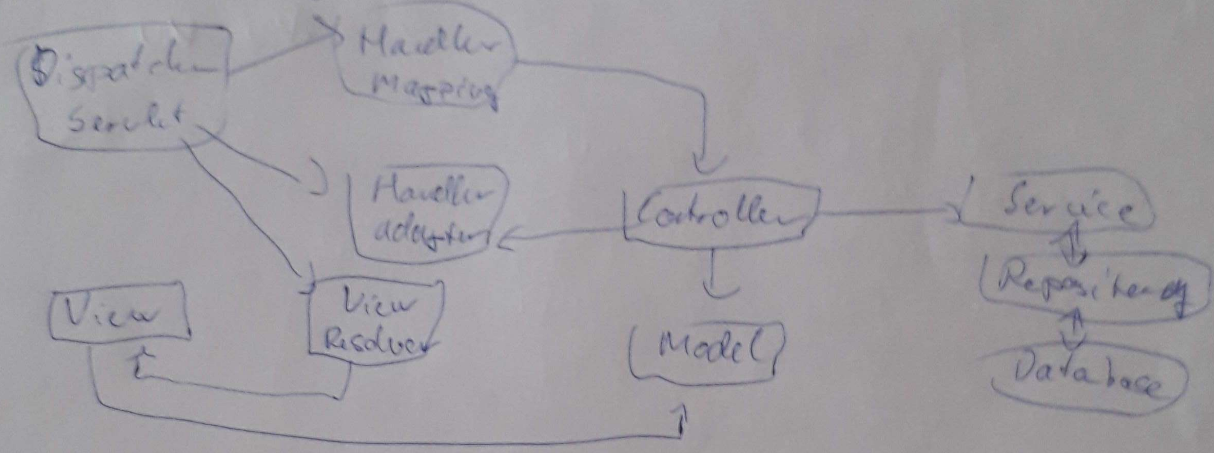
Pócsos Péter
LHS256
2021.01.14

(3) Spring MVC

Modell View Controller

A vezérlés és a megjelenés különválik, szerepenként átláthatóan, modulárisan építjük fel a webalkalmazást

~~Controller - a vezérlést tartalmazza amely az osztályokat melyet a~~



Dispatcher Servlet: a szervlet amin fut a webalkalmazás

Handler Mapping -> a controllert hívja mely kiválasztja a helyes eseménykezelőt

Handler adapter -> konverzió végzés

View resolver -> megjelenés előállítása

View -> A megjelenő weblap

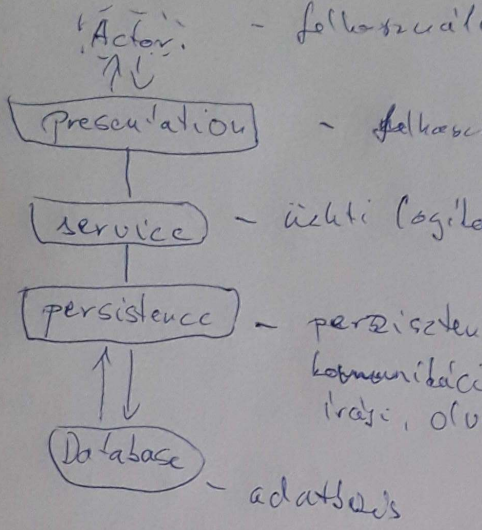
Controller -> A vezérlőegység melynek központi szerepe van rajta keresztül történik a kommunikáció a weblap és a mögöttes szerver architektúra között

Modell -> eredmények inputok kezelése

Service, Repository, database -> Mögöttes architektúra

Pódos Péter
LHS250
2021.01.14

(4) Spring Framework



Inversion of Control: A vezérlés megfordítása, a tradicionális kódfuttatás menetének a megfordítása. procedurális programoknál esetén a program sorban halad végig, hívva azokat az univerzális könyvtárakat amire szüksége van. Ezzel szemben az IoC, egy keretrendszertől kapja meg a futtatást úgynevezett, ő a keretrendszer dönti el hogy éppen melyik modul kapja meg a vezérlést. Segíti a moduláris program felépítését.

Dependency Injection: Függőség befecskendezés, erre akkor van szükség, ha a programunknak valamilyen külső, függősége van, vagyis egy objektumnak szüksége van egy másik objektumra ahhoz, hogy futni tudjon.

Módjai - Maven -> egy úgynevezett pom.xml fájl segítségével adhatjuk meg a külső függőségeket. Ezek lehetnek lokálisak vagy akár a webről elérhető állományok is.