

Model si repository

Pentru problema aleasa in laboratorul 1 din ProblemeLab.pdf proiectati si implementati (in Java si C#) partea de model si repository folosind baze de date relationale.

Termen de predare: saptamana 3.

Servicii si interfete grafice

1. Proiectati si implementati partea de servicii si interfata grafica a proiectului (in Java si C#): controllerul interfetei grafice apeleaza metodele din servicii, iar serviciile folosesc repositoryes (Java si C#).

Termene de predare:

- saptamana 4 intr-unul dintre limbaje (la alegerea studentului),
- saptamana 5 in celalalt limbaj.

Networking

1. Implementati partea de networking (folosind socketuri si threaduri) a proiectului (in Java si C#) si notificarea clientilor.

Termene de predare:

- saptamana 6 intr-un limbaj (la alegerea voastra)
- saptamana 7 in celalalt limbaj

Aplicatii distribuite folosind tehnologii RPC

Modificati una dintre solutiile de la tema 3 (la alegerea voastra) pentru a folosi una dintre tehnologiile RPC: RMI, Spring Remoting, sau .NET Remoting.

Termen de predare: saptamana 8.

Aplicatii distribuite cross-platform

Modificati solutiile de la Tema3_proiect (Networking) astfel incat serverul sa fie dezvoltat intr-un limbaj (la alegerea voastra) si clientul/clientii in celalalt limbaj de programare. Daca serverul este in C#, clientii vor fi in Java, iar daca serverul este in Java clientii vor fi in C#.

Observatii

1. Studentii care vor folosi pentru aceasta tema una dintre tehnologiile gRPC sau Thrift vor primi un punct in plus la nota finala (cu conditia promovarii anterioare a materiei). Codul sursa al solutiei dezvoltate folosind gRPC sau Thrift trebuie trimis prin email cadrului didactic de la curs.

2. Studentii care vor folosi pentru aceasta tema Protocol Buffers (fara gRPC) vor primi 0.5 puncte in plus la nota finala de la laborator.

Termen de predare: saptamana 9

ORM

Pentru cel putin o entitate din problema folositi un instrument ORM (Hibernate, Entity Framework, etc) pentru una dintre solutiile anterioare (la alegerea voastra).

Termen de predare: saptamana 10.

Servicii REST

1. Proiectati si implementati servicii REST pentru entitatea corespunzatoare problemei de la proiect precizata mai jos (crearea, modificarea, stergerea, cautare dupa id, afisarea tuturor entitatilor disponibile).

Problema 1: Zbor

Problema 2: Artist

Problema 3: Proba

Problema 4: Cursa

Problema 5: Proba

Problema 6: Excursie

Problema 7: Caz caritabil

Problema 8: Proba

Problema 9: Proba cu arbitrul asociat

Problema 10: Meci

Problema 11: Cursa

2. Testati serviciile folosind un client REST (extensie browser web, ex. Advanced Rest Client, Postman).

3. Scrieti un program de test in Java.

4. Scrieti un program de test in C#.

Termen de predare:

Informatica germana (an 3) - saptamana 11

Informatica romana (an 2) - saptamana 12

Client Web servicii REST

Dezvoltati o aplicatie web care permite vizualizarea tuturor resurselor de tipul entitatii primite la tema 7, respectiv adaugarea si stergerea unei noi resurse, folosind serviciile Rest de la tema 7.

Pentru dezvoltarea aplicatiei web puteti o tehnologie la alegerea voastra, care sa nu fie bazata pe Java sau .NET.

(Puteti folosi ReactJS, AngularJS, PHP, etc. dar nu puteti folosi JSP, Servlet sau ASP.NET).

Termen de predare:

Informatica linia germana (an 3) - saptamana 12

Informatica linia romana (an 2) - saptamana 14

Tema facultativa. Sisteme bazate pe mesaje

Pentru una dintre solutiile client-server anterioare (la alegerea voastra), proiectati si implementati notificarea clientilor folosind un broker de mesaje (ex. activemq, rabbitmq, etc).

Termen de predare:

Informatica germana (an 3) - saptamana 12

Informatica romana (an 2) - saptamana 14

Cine preda aceasta tema primeste un punct in plus la nota de la laborator.

Lab 1

Tema in timpul laboratorului

Pentru una dintre temele de la laboratorul de MAP (la alegerea voastra) faceti urmatoarele:

-creati un proiect Gradle

-folosind Gradle creati fisierul jar executabil asociat

-scrieti cel putin doua teste automate (folosind junit) si rulati-le folosind Gradle

-adaugati jurnalizare uneia dintre clase folosind Gradle si log4j2. Mesajele se vor salva intr-un fisier text.

Lab 2

Accesarea unei baze de date relationale

1. Creati o baza de date relationala (MySQL/Sqlite/etc, exceptand SqlServer).
2. Pentru o entitate la alegerea voastra, creati o tabela in baza de date cu cel putin 3 coloane avand o coloana de tip numar si o coloana de tip text.
3. Definiti si implementati in Java/C# repository pentru entitatea aleasa folosind baza de date (trebuie sa aveti cel putin cate o operatie de select, delete, insert si update).
4. Scrieti un program pentru a testa operatiile din repository.

Lab 3

Injectarea dependentelor

1. Pentru un proiect Java la alegere (sau proiectul de la MPP) configurati aplicatia folosind Spring pentru injectarea dependentelor. Folositi doua metode pentru configurare (fisier XML si cod Java). Proiectul trebuie sa aiba cel putin 3 bean-uri si intre 2 dintre ele sa existe dependente.

Lab 4

Aplicatii client-server

Pentru problema de la proiect implementati in Java functionalitatea de autentificare(login) folosind socketuri. Verificarea datelor se face folosind baza de date.