Proiecte propuse pentru disciplina Programare avansată (Java)

ultima actualizare: 14.03.2021

- Propunerile au caracter informativ. Nu există specificaţii exacte.
- Se poate lucra în echipe de 2 persoane
- Sunt acceptate si alte limbaje ale platformei Java: Kotlin, Scala.
- Interfetele grafice pot fi create cu orice tehnologie: Swing, JavaFX, eventual Web sau Android daca fac parte dintr-un proiect ce utilizeaza servicii
- Preluarea de cod sursa din repo-uri publice se recompenseaza cu punctaj negativ
- Implementarile cele mai bune pot fi continuate ca proiect de licenta

1. **ExamNet**
   - Definirea unui format XML pentru reprezentarea unor teste formate din intrebari grila si enunturi de probleme, respectiv pentru preluarea solutiilor
   - Crearea unor servicii pentru transmiterea testelor, preluare rezultatelor
   - Optional: crearea unei interfete Web
   - Similar cu examinarea implementata pe platforma pbinfo

2. **GraphEditor**
   - Crearea unui editor pentru grafuri, folosind diverse tipuri de formate (GraphML, TGF, etc) cu posibilitatea exportului acestora in diferite formate (PDF, SVG, Tikz, etc.)
   - Implementarea unor algoritmi pentru desenarea automata a grafurilor (Tree, Circular, Spring, etc)
   - Optional: crearea de animatii
   - Similar cu yEd (https://www.yworks.com/products/yed)

3. **DiscordBot**
   - Servicii REST, Discord API
   - Crearea unui bot pentru Discord capabil sa ofere mesaje preluate prin RSS pe anumite teme (programare, Java, etc) si sa raspunda la intrebari simple
   - Pot fi folosite biblioteci RSS (Rich Site Summary or Really Simple Syndication) cum ar fi ROME

4. **WordGames**
   - Crearea unor servicii pentru implementarea unor jocuri cu cuvinte (Scrabble sau altele) pentru limba română
   - Serviciile pot fi: gestionarea unor dictionare, anagrame, accesarea unor servicii pentru obtinerea definitilor, sinonimelor (Dex), suport pentru anumite jocuri, etc.
   - Implementarea unui joc care foloseste aceste servicii cu interfata JavaFX

5. **MazeGenerator**
   - Cercetare, Algoritmica, Grafică
   - Studierea si implementarea unor algoritmi pentru generarea de labirinturi 2D sau 3D (maze, dungeon, etc), cu posibilitatea de export in diverse formate
   - http://www.astrolog.org/labyrnth/algrithm.htm

- Crearea unei interfete grafice care sa asiste procesul de generare

6. **DistributedSupervisor**
   - Google Docs Api, Git Api
   - Evidenta activitatii studentilor la seminarii si laboratoare la nivelul unui curs; se va tine cont de faptul ca pot fi mai multe grupe, pot fi mai multi profesori, pot fi mai multe unitati de lucru (saptamani); Profesorii vor tine evidenta folosind Google Spreadsh-uri personale; aplicatia trebuie sa fie capabila sa permita definirea regulilor prin care sunt culese date din fiecare si sa le centralizeze.
   - Optional: pentru fiecare student, o unitate de lucru poate avea specificat un proiect pus pe Git. Aplicatia poate verifica similaritatile dintre proiecte.

7. **DeepLearner**
   - Retele neuronale
   - Utilizarea sau crearea unui dataset (https://www.kaggle.com/datasets)
   - Crearea unei retele neuronale care sa ofere predictii, pe baza acestui dataset, folosind DeepLearning 4J (https://deeplearning4j.org)
   - Crearea unei interfete grafice desktop

8. **OPD: Optimal Financial Portfolio Design**
   - Cercetare, Algoritmica
   - Modelarea unei probleme de satisfacere a constrangerilor (CSP), utilizarea unui solver (Choco Solver, Opta Planner, etc.), crearea unei interfete desktop pentru definirea instantelor, vizualizarea rezultarelor, etc.
   - Detalii suplimentare: https://www.csplib.org/Problems/prob065
   - *Pot fi abordate si alte probleme similare de la* *CSPLib: A problem library for constraints*

9. **MDVSP: Multi Depot Vehicle Scheduling Problem**
   - Cercetare, Algoritmica
   - Studierea problemi MDVSP si implementarea unui scenariu practic, de exemplu: o firma de transport persoane are un numar de garaje, fiecare garaj are un numar de masini si trebuie sa planifice calatoriile pentru urmatoarea zi; fiecare calatorie are definite locatiile de start si destinatie, intervalul de timp in care trebuie sa fie onorata; aplicatia trebuie sa planifice calatoriile, minimizand distanta totala parcursa (sau numarul de masini folosite);crearea unei interfete desktop.
   - Detalii suplimentare: la cerere

10. **VRP: Vehicle Routing Problem**
    - Cercetare, Algoritmica
    - Studierea problemi VRP si implementarea unui scenariu practic, de exemplu: o firma de curierat trebuie sa livreze pachete achizitionate online; se stiu dimensiunile/greutatile pachetelor, capacitatile masinilor aflate la dispozitie, locatiile clientilor si preferintele acestora legate de timp; aplicatia trebuie sa

planifice calatoriile, minimizand distanta totala parcursa; crearea unei interfete desktop.
- Detalii suplimentare: la cerere

11. GoClient (pentru pasionatii jocului de Go (game))
- Crearea unei aplicatii client, cu interfata grafica, pentru jocul de Go, intre doua persoane, in retea.
- Integrarea cu 'engine'-uri performante pentru jocul de Go, cum ar fi KataGo https://github.com/lightvector/KataGo/ sau GnuGo https://www.gnu.org/software/gnugo/
- Poate fi continuat cu: suport pentru editarea de probleme, studierea Joseki-urilor, etc.

12. Backgammon (pentru pasionatii jocului de table:)
- Crearea unei aplicatii client, cu interfta grafica care sa permita jocul intre doua persoane in retea
- Crearea unui algoritm (AI) capabil sa joace la un nivel rezonabil
- Poate fi continuat cu studierea algoritmilor bazati pe retele neuronale

13. SortingNetworks
- Crearea unei aplicatii cu interfta grafica care sa permita crearea, testarea, analiza, vizualizarea retelelor de comparatori si de sortare. https://en.wikipedia.org/wiki/Sorting_network
- Implementarea unor algoritmi generici de construire, cum ar fi Batcher odd–even mergesort

14. RecommendationSystem
- Crearea unui sistem de recomandari pentru carti/filme/etc,
- Utilizarea unui sistem de gestiune a datelor bazat pe grafuri Neo4j Graph Platform – The Leader in Graph Databases
- Crearea de servicii REST si interfeta Web

15. ArgumentationFramework
- Cercetare, Constraint Programming
- Studierea conceptelor din teoria argumentarii https://en.wikipedia.org/wiki/Argumentation_framework
- Implementarea unei biblioteci cu algoritmi pentru analiza unui sistem de argumentare
- Social Abstract Argumentation, On the acceptability of arguments

16. Traffic Accident Detection
- Cercetare, Stream Processing
- Simularea unor sensori de trafic prin generearea datelor din trafic.
- Folosirea Java Stream API pentru construirea unor interogări și detectarea anomaliilor
- Events stream processing (Apache Kafka + Apache Flink)

17. Correct an address

- Write an algorithm that corrects the fields country, state, city of a postal address. Example: Country: RO, State: New York, City: Iasi will become
  Country: RO, State: Iasi, City: Iasi
- the algorithm needs to have unit tests and integration tests for performance and precision
- ideally the algorithm will work for all countries in the world and a few languages
- Expose a REST api using spring boot that will receive a postal address and return the corrected result
- Deploy the application as a docker container in aws/heroku or other using a continuous deployment pipeline

18. TODO
19. TODO
20. TODO