Medius d.o.o.
Tehnološki park 21, SI-1000 Ljubljana
info@medius.si
www.medius.si

Naloga za zaposlitev

Uvod

Cilj naloge je preveriti poznavanje Jakarta EE in sestavljanje čim bolj optimalnega algoritma za reševanje danega problema.

Tehnologije

Naloga mora biti rešena v programskem jeziku Java (verzija 17 ali višje). Za upravljanje s knjižnicam naj se uporabi Maven, za strežnik pa Quarkus.

Naloga

Z uporabo Jakarta EE naredi aplikacijo, ki bo omogočala uporabnikom kreiranje nove igre Lights Out, ter možnost dodajanja rešitev.
Aplikacija mora izpostaviti naslednje REST API-je ter njihovo dokumentacijo (na primer generirana s swaggerjem) :

- GET /players vrne vse shranjene uporabnike
- GET /players/{username} vrne uporabnike z *username*
- POST /players ustvari novega uporabnika
- GET /problems vrne vse probleme
- GET /problems/creator/{username} vrne vse probleme, ki jih je naredil uporabnik s uporabniškim imenom *username*
- GET /problems/{id} vrne problem z id
- POST /problems ustvari nov problem
- GET /solutions vrne vse rešitve
- GET /solutions/solver/{username} vrne vse rešitve, ki jih je oddal uporabnik z uporabniškim imenom *username*
- GET /solutions/problem/{id} vrne vse rešitve za problem z *id*
- POST /solutions ustvari nov solution

Znati mora rešiti Lights Out problem (pravila so pod Podrobnosti -> Lights Out), ki se pošlje, saj želimo shraniti zgolj tiste, ki so rešljivi. Se pravi, ko uporabnik ustvari nov

Medius d.o.o.
Tehnološki park 21, SI-1000 Ljubljana
info@medius.si
www.medius.si

problem, ga program proba rešiti. Če je problem rešljiv, ga shranimo in obvestimo uporabnika o uspešnosti, drugače problema ne shranimo in obvestimo uporabnika, da problem ni rešljiv. Iz logov mora biti razvidno kako hitro deluje solver ter v koliko potezah je prišel do rešitve oz. v primeru da je rešitev nerešljiva izpiše opozorilo, da ta problem nima rešitve. Ko uporabnik pošlje svojo rešitev za dani problem, program preveri, če je rešitev pravilna. Če je pravilna, jo shrani in vrne odgovor o uspešnosti, drugače pa jo zavrže in obvesti uporabnika, da rešitev ni pravilna. Ustvariti je potrebno tudi bazo kamor se shranjujejo problemi ter rešitve. Osnoven opis baze in kaj mora minimalno vsebovati je opisano pod Podrobnosti -> Opis baze.

Izvorna koda naj bo dostopna preko git repozitorija, nam pa se posreduje povezavo preko katere bomo lahko dostopali do kode. Za objavo repozitorija se lahko uporabi poljubno rešitev (Gitlab, Github, ...).

Podrobnosti

Opis baze:

Baza mora biti sestavljena iz vsaj 4 tabel: player, problem, solution, solution_step.
Player mora vsebovati username, ter starost.
Problem mora imeti polje za opis problema (binarna predstavitev, kot seznam,..) ter sekundarni ključ na playerja, ki je ustvaril problem.
Solution mora imeti dva sekundarna ključa enega na problem drugega na player.
Solution step je sestavljen iz sekundarnega ključa na solution, poteze in vrstni red poteze.
Poljubno se lahko dodajo še druge tabele ter polja.

REST API:

Vsak API mora vrniti odgovor v obliki JSONa, prav tako POST klici sprejmejo objekt v obliki JSONa. Kaj točno sprejme in vrne API se odločite sami vendar naj bo smiselno in povezano glede na tabele v bazi.

Solver:

Solver mora v zglednem času poiskati rešitev oz. javiti, da rešitev ne obstaja, saj želimo uporabniku javiti ali je uspešno ustvaril problem ali ne (nerešljivih problemov ne shranjujemo). Do rešitve lahko pridete s poljubnim algoritmom, vendar pa mora

Medius d.o.o.
Tehnološki park 21, SI-1000 Ljubljana
info@medius.si
www.medius.si

biti algoritem pravilen in se mora končati za vse možne primere. Omejite se na polja velikosti med 3x3 in 8x8. Pravila za problem najdete pod Lights Out.

Lights Out:

Je igra kjer je potrebno vse vrednosti v binarni nxn matriki (matrika ima vrednosti 0 ali 1) spraviti na same 1. Kadarkoli je možno “pritisniti” na katero koli polje v matriki. Ob “pritisku”, izbrano polje spremeni vrednost (iz 1 v 0 ali iz 0 v 1). Prav tako spremeni vrednost polje levo, desno, zgoraj in spodaj od “pritisnjenega” polja (če je to mogoče).
Primer začetnega stanja:

0	0	1
0	1	0
0	0	1

Primer “pritiska”:

0	0	1
0	1	0
0	0	1

S “pritiskom” na rdeče označeno polje dobimo:

0	0	0
0	0	1
0	0	0