

Nume: Grigoras Adrian
Grupa: 322CC
Gradul de dificultate: Mediu

Din punctul de vedere al timpului alocat pentru rezolvare:

-am avut timp de alte teme concomitent si am avut si timp liber, deci consider ca durata de rezolvare a fost una potrivita

Modul de implementare:

Am folosit IntelliJ Idea ca IDE pentru realizarea temei

Am facut clasele cerute la primul task cu fiecare metoda necesara

O sa mentionez adaugarile metodelor sau claselor pe care le-am folosit:

-in Application am facut si Singleton Pattern, din cauza asta aparitia campului instance si a metodei getInstance

-in Consumer o metoda toString(pentru partea de interfata grafica)

-in Resume, am implementat builder pattern deci am pus campurile private si am pus functii de get pentru a lua valorile din campurile functiei. De asemenea, am facut-o intr-o clasa separata pentru a putea implementa builder-ul

-in Education o metoda ToString pentru interfata grafica

-in Experience o metoda toString pentru interfata grafica

-in User am implementat interfata Observer si am facut metoda update.

-Am adaugat o lista de notificari pentru interfata.

-Functia ani_experienta e ca sa calculez anii si in functie de numarul de luni cum e cerut in cerinta

-in Recruiter am stabilit ca poate trimite orice cerere catre Manager chiar daca nu indeplineste cerintele din meetsRequirements

-in Manager trimit niste notificari in cazul in care dupa process, un utilizator este respins, sau jobul este full

-in Company am adaugat getAllEmployees pentru usurinta la interfata grafica

-De asemenea, am o lista de observatori formata din userii care urmaresc compania respectiva

-DepartmentFactory e clasa necesara pentru crearea departamentelor cu ajutorul Factory Pattern.

-Testarea am facut-o dupa ce am citit-o cu un parser json (json-simple-1.1.1.jar)

-Ce nu era trecut in Json(companiile, reseaua sociala si joburile disponibile) le-am facut de mana din cauza ca nu am mai avut timp sa fac fisiere text din care sa citesc

-functiile din Test sunt pentru parsarea campurilor de education si jobs pentru fiecare tip de utilizator.(parseEducation si parseJobs)

-parseExperience e pentru ultimul element din Experience pentru a lua informatiile necesare pentru a stii unde munceste acum(Recruiter, Manager si Employee)

Interfata grafica:

-am implementat toate lucrurile cerute
bonusuri:

-am adaugat clase care mostenesc ProfilePage si in functie de asta apar lucruri personalizate pentru EmployeePage, RecruiterPage sau User(la User lista cu notificari, la Employee si Recruiter datele legate de ei de la companii cum ar fi salariu, numele companiei)

-am adaugat OpenPage pentru a stii cine vrea sa se logheze(Profile, Admin sau Manager)

-am adaugat clasa selectManager care arata toate companiile si din care poti sa alegi sa te loghezi la ea in functie de ce colectie de angajari vrei sa modifici

- am adaugat ConsumersPage din care pot sa aleg ce tip de consumer sunt (Employee, Recruiter sau user)
- am adaugat Login_consumer in care trebuie sa iti bagi mail-ul sa te loghezi
- am adaugat pentru paginile terminale un buton de back care te duce la openPage si lasa pe altcineva sa se logheze
- am adaugat in ManagerPage optiunea process care proceseaza toate cererile de angajare fara nevoia interventiei managerului(doar cu modul asta, userii primesc notificari. Daca managerul decide el cu accept si reject, nu vor primi notificari daca sunt respinsi).