Andrei Alin-Ionuț 323CC
Gradul de dificultate al temei:mediu :)
La tema am lucrat 15-20 ore.
Implementarea: am folosit Intellj

Task 1 ::

Clasa Aplication:

- metoda getJobs la implementarea ei am parcurs lista de compani primita ca
parametru apoi am comparat cu numele companiilor din lista pe care o are clasa
Aplication, am gasit acele compani apoi pentru fiecare companie am parcurs
departamentele si am extras joburile care erau deschise si le-am pus intr-o
lista noua pe care am returnat-o la final;

- pe langa metodele cerute am mai implementat si metoda userDetect care imi
returneaza un utilzator daca acesta exista si null in caz contrar;

- restul metodelor nu necesita o detaliere comentariile sunt suficiente;

Clasa Consumer:

- in acceasta clasa am ales sa implementez o metoda bsf pentru parcurgerea in
latime a retelei sociale cu acceasta metoda retin pentru un utilizator toti
utulizatorii care au un grad de pritenie (se poate ajunge la ei ca intr-un
arbore) si nivelul pe care se afla;

- metoda getDegreeInFriendship se foloseste de metoda bsf pentru a detecta
cunostintele de pe toate nivele ale unui Consumer, totodata ea retine pentru
fiecare persoana din retea si nivelul pe care se afla, apoi caut indexul
utilizatorului curent si returnez elementul de pe acel index din vectorul cu
nivele;

- am ales sa am o metoda care imi calculeaza anii de experienta ai unui
Consumer, acceasta av fi folosita cand voi calcula scorul unui Consumer.

Clasa Resume:

-in aceasta clasa am folost ca si colectii sortate pentru experienta si educatie
"ArreyList" care este sortat de fiecare data cand e adaugat ceva in el dar si
cand e initializat;

Clasa Information care respecta principiul incapsularii are variabilele private
dar are metode prin care poti fi accesatesi modificare;

Clasa Education care implementeaza interfata Comparable contine pe langa constructor metoda compareTo care exectuta comparatia in functie de data de
sfarsit "stop" sau in cazul in care aceasta e null folosid data de inceput;

Clasa Experience e implementata in mod asemanator clasei Education comparatia
facandu-se folosind dara de sfarsit respectiv numele companiei

Clasa User:

-metoda media care calculeaza media tuturor parcurgand lista de studii din clasa
interna Resume;

-metoda getTostatScore care se foloseste de metoda media() si experienta()
pentru a calcula scorul;

-metoda convert() creaza un nu Employee care are datele fostului Utilizator ce
s-a angajat;

Clasa Employee: acceasta clasa extinde Consumer si cntine doar un constructor pe
langa cele 2 campuri noi;

Clasa Recrutier:are un constructor care pe langa crearea unui obiect Recrutier
adauga acel Recrutier in departamentul IT, prin parcurgerea listei de departamente din companie(am detectat compania folosind metoda statica din
clasa Aplication getCompany()) apoi am comparat numele claselor ce extind
un departament pana am gasit departamentul IT in care am adaugat acel angajat;

- metoda evaluate() care care calculeaza scorul unui utilzatoru cu ajutorul
metodei getTotalSCore() apoi creaza o noua cerere (Request) care este pusa in
lista de cereri pe care o are managerul companiei;

Clasa Manager:

-metoda process(job) parcurg lista de cereri si caut cererile pentru job-ul

cerut apoi pentru fiecare post liber caut cel mai bun candidat"max", pentru
noua lisat de cereri caut cererea cu scorul cel mai mare care indeplineste
si cerintele minime, apoi apelez metoda convert din clasa User si creez un
angajat adaugandui apoi si salariu si numele companiei, urmand apoi sa il adaug
in lista de angajati a departamentului;

Clasa Job: contine constructorul dar si metoda meetsRequirments care verifica
daca un tilizator indeplineste condiitiile minime;

Clasa Contrait: contine doar un constructor si doua variabile de tip int care
reprezinta limitele cerintelor;

Clasa Company:

-metoda move(Department source, Department destination) parcurge lista de angajati ai departamentului sursa mutand foiecare angajat in noul departament
apoi departamentul este sters

-metoda getRecruiter foloseste la ineput un functia bsf din clasa Consumer
pentru a determina arborele de cunoscuti si nivelele Utilizatorului, apoi parcurg lista de recrutieri si daca unul nu a fost vizitat il aleg deoarece
inseamna ca nu exista un nivel de prietenie, in cazul in care toti recutieri se
afla in lista de vizitati parcurg acea lista si caut recutierii cu cel mai mare
nivel, daca am gasit pentru un anumit nivel recrutieri ies in while, in cazul in
care e unul singur intorc acel recrutier in caz contrar il caut pe cel cu cel
mai mare scor si il returnez;

Clasa Departament si cele care o extind IT, Management, Marketing si Finance
contin metoda care calculeaza bugetul, acesta este calculat cu un for ce parcurge lisat de angajati si in functie de cat platesc impozit calculez bugetul;

Task 3   ::
Am folosit Singleton pattern pentru clasa Aplication

Clasa Factory implementeaza Factory pattern ea contine o metoda care returneaza
un anumit departament in functie de numele primit ca parametru;