Implementati o clasa abstracta Echipa care modeleaza o echipa in cadrul unei firme. Echipa are neaparat un lider si o lista de membri(pentru lista sa se folsoeasca structuri de date potrivite din API-ul de colectii, adica ArrayList etc). La aceasta echipa se poate adauga oricand un membru cu metoda Boolean addMember(Membru member). Metoda o sa returneze true daca membrul are loc in echipa, false in caz contrar. Orice Echipa are un nume si un numar maxim de membri. Numele si locul maxim de membri o sa fie setat prin constructor.

Liderul este setat dupa ce echipa este instantiata utilizand metoda boolean setLeader(Membru liderNou). Pentru ca un membru sa poate sa fie pe post de lider, acesta trebuie sa aiba minim 5 ani de experienta. In cazul in care membrul primit indeplineste conditia sa fie lider, metoda returneaza true, altfel false.

Din echipa se pot da afara membrii prin metoda Member removeMember(Membru member). Se va returna membrul scos daca acesta a fost gasit in lista de membrii, altfel se va returna null.

Fiecare echipa are a o metoda toString() care va returna un obiect de tip String de forma: “Lider echipa: <nume-prenume\_lider>, Membri

O ultima metoda va fi metoda abstracta double getCost() care returneaza in functie de tipul echipei costurile asociate acestei echipe.

Clasa Echipa va avea si settere pentru numarul maxim de membri si nume dar doar liderul de echipa va putea sa schimbe aceste date. Daca un membru simplu o sa incerce sa faca asta, o sa se afiseze un mesaj pe ecran sa il instiinteze ca nu are voie si nu o sa se faca schimbarea. Verificarea dintre leader si membrul primit ca parametru sa se faca utilizand metoda equals care trebuie sa fie implementata in clasa Membru.

Liderul si membri vor fi amandoi de tipul Membru. Clasa Membru modeleaza un membru. Fiecare membru are un nume, prenume, varsta, salariu si un numar de ani de experienta in domeniu.

Exista 2 tipuri de echipe: DevTeam, HR. Ambele echipe extind clasa abstracta Echipa si o sa suprascrie metoda getCost().

Echipa DevTeam o sa calculeze astfel costurile: pentru lider costurile sunt 2500 la care se adauga anii de experienta inmultiti cu 250(2500 + ani\_exp\*250). Pentru un membru simplu, costul asociat acestuia va fi de 1500 la care se adauga un bonus procentual in functie de experienta: daca are mai sub 2 ani, procentul este 0%, daca are intre 2 si 5 ani de experienta, procentul este de 25% iar daca are peste 5 ani de experienta, procentul este de 50%. Metoda va returna costurile totale asociate echipei.

Echipa HR o sa calculeze astfel costurile: pentru lider costurile sunt 1350 la care se adauga 300 pentru fiecare an de experienta. Pentru un membru simplu, costul asociat acestuia va fi de 1000 la care se adauga un bonus procentual identic cu cel de la echipa DevTeam.

In clasa Main sa se faca 2 echipa, una de DevTeam si alta de HR, sa se incerce adaugarea unui lider care nu este eligibil, apoi sa se adauge un lider eligibil. Sa se adauge macar 5 angajati unici la fiecare echipa si sa se afisaze costurile totale pentru fiecare echipa. Sa se afiseze si echipele utilizand metoda toString(). Sa se stearga un membru si sa se afiseze membrul sters, apoi sa se afiseze inca o data echipa si costurile associate ei pentru verificarea ca s-a facut corect stergerea.