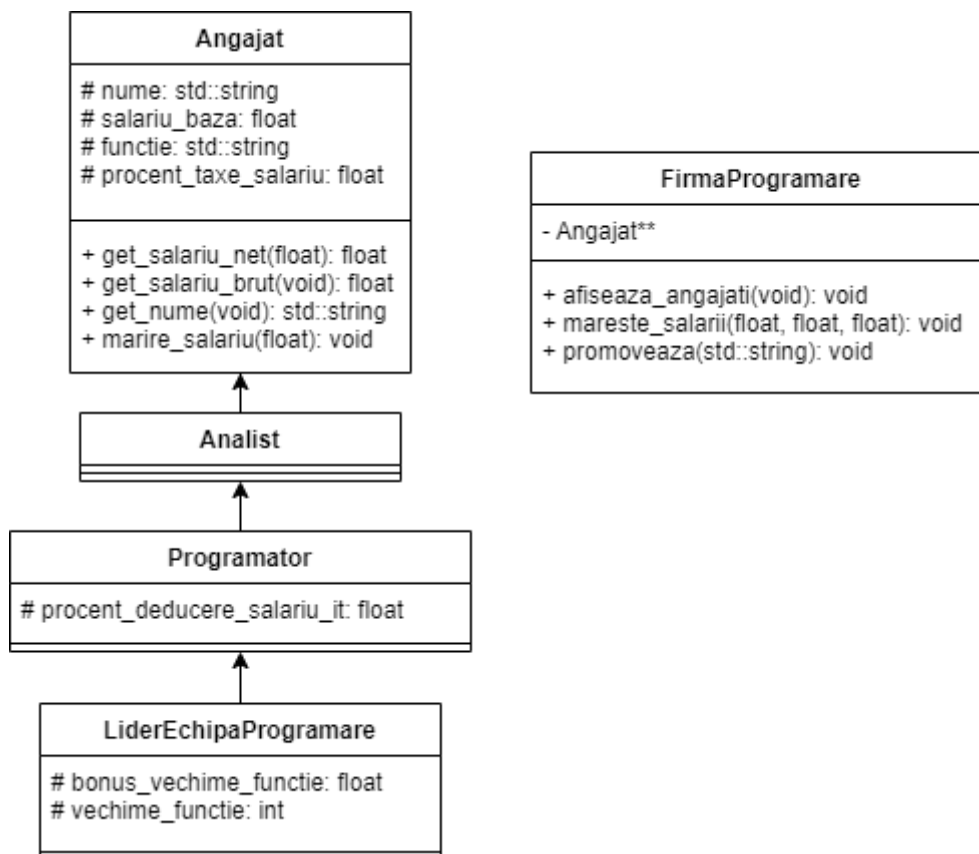


Să se scrie o aplicație care modelează organigrama cu rolurile angajaților într-o firmă IT și ajută la managementul salariilor. Pentru a ajuta realizarea programului, este pusă la dispoziție diagrama UML de mai jos. Rolurile posibile în cadrul firmei sunt identice cu numele claselor din diagrama UML (Analist, Programator, LiderEchipaProgramare):



Se citesc de la tastatură, pe linii diferite, următoarele informații:

- Numărul de angajati care vor fi introduși în aplicație
- Pentru fiecare angajat, pe linii diferite:
  - Numele angajatului
  - Rolul angajatului in firmă
  - Salariul lui de bază (inainte de taxe=brut)
  - Daca angajatul este „LiderEchipaProgramare”, atunci se va citi și vechimea in funcție, în ani
- O valoare întreagă în intervalul [1, 5] în funcție de care se realizează următoarele operații:
- 

1. (x1) **Afișarea angajatilor.** Pentru fiecare angajat, în ordinea citirii de la tastatură, se vor afișa, pe rânduri diferite:

- Numele
- Funcția în firmă

2. (x2) **Calcularea salariilor.** Pentru fiecare angajat, în ordinea citirii, se vor afișa salariul brut și salariul net, pe rânduri diferite. Se consideră că procentul de taxă pentru salariu este de 40% și procentul de deducere taxe pentru IT este de 10%. Bonusul de vechime in funcție este 500 lei/an brut.

- Numele
- Salariu brut: <Valoare salariu brut> (se calculează ca **salariu\_baza + bonus\_vechime\_functie \* vechime\_functie**). Bonusul de vechime in funcție se va adăuga doar acolo unde este necesar.
- Salariu Net: <Valoare salariu net> (se calculează ca **(salariu\_baza + bonus\_vechime\_functie \* vechime\_functie) \* (100 – procent\_taxe\_salariu + procent\_deducere\_taxe\_it) / 100**). Bonusul de vechime in funcție si procentul de deducere de taxe pentru IT se vor adăuga doar acolo unde este necesar.

3. (x2) **Primirea de mărire.** Se hotărăște acordarea de mărire pentru angajați, depinzând de funcția în firmă. Se vor citi, de pe același rând, pentru fiecare funcție, numele funcției în firma și procentul de mărire a salariului de bază, sub forma:

- functie\_in\_firma procent\_marire\_salariu

Se vor afișa pe ecran atât noile salarii ale angajaților cât și cele vechi, în ordinea citirii de la tastatură, sub forma:

- Nume
- Salariu de baza vechi: <salariu\_baza (inainte de aplicarea maririi)>
- Salariu de baza nou: <salariu\_baza (dupa aplicarea maririi)>

4. (x2) **Promovări.** Se citește de la tastatură numele unui angajat, care urmează să fie promovat. Pe rândul următor, se va citi funcția la care va fi promovat. La apelarea fiecărui destructor, se va afișa un mesaj de forma: "Destructor <nume\_clasa>". Se vor afișa apoi pe ecran, în ordinea citirii, pentru fiecare angajat:

- Numele
- Funcția în firmă

Observații:

1. Se vor folosi clasele și funcțiile prezentate în diagrama UML. În caz de nevoie, se pot adăuga variabile/funcții/clase suplimentare
2. Se vor respecta principiile încapsulării și abstractizării - rezolvările strict procedurale nu se vor lua în considerare
3. Nu se vor folosi clase/funcții friend

#### 1. Afisarea angajatilor

| Input  | Output   |
|--|--|
| 3<br>Ion Cristian<br>Programator<br>8000<br>Mihai Mihailescu<br>Analist<br>6000<br>Bogdan Vezure<br>LiderEchipaProgramare<br>14000<br>6<br>1 | Ion Cristian<br><br>Programator<br>Mihai Mihailescu<br>Analist<br>Bogdan Vezure<br>LiderEchipaProgramare |

#### 2. Calcularea salariilor

| Input  | Output   |
|--|--|
| 2<br>Ion Cristian<br>Programator<br>8000<br>Mihai Mihailescu<br>Analist<br>6000<br>2 | Ion Cristian<br>Salariu brut: 8000<br>Salariu Net: 5600<br>Mihai Mihailescu<br>Salariu brut: 6000<br>Salariu Net: 3600 |

3. Calcularea salariilor x2

| Input  | Output   |
|--|--|
| 2<br>Ion Cristian<br>LiderEchipaProgramare<br>15000<br>1<br>Robert Popescu<br>LiderEchipaProgramare<br>12600<br>8<br>2 | Ion Cristian<br>Salariu brut: 15500<br>Salariu Net: 10850<br>Robert Popescu<br>Salariu brut: 16600<br>Salariu Net: 11620 |

4. Primirea de mariri

| Input   | Output   |
|---|--|
| 2<br>Ion Cristian<br>Programator<br>8000<br>Robert Popescu<br>LiderEchipaProgramare<br>12600<br>8<br>3<br>Analist 35<br>Programator 10<br>LiderEchipaProgramare 5 | Ion Cristian<br>Salariu de baza vechi: 8000<br>Salariu de baza nou: 8800<br>Robert Popescu<br>Salariu de baza vechi: 12600<br>Salariu de baza nou: 13230 |

5. Primirea de mariri v2

| Input  | Output   |
|--|--|
| 2<br>Ion Cristian<br>Analist<br>4060<br>Robert Popescu<br>LiderEchipaProgramare<br>12600<br>8<br>3<br>Analist 15<br>Programator 0<br>LiderEchipaProgramare 105 | Ion Cristian<br>Salariu de baza vechi: 4060<br>Salariu de baza nou: 4669<br>Robert Popescu<br>Salariu de baza vechi: 12600<br>Salariu de baza nou: 25830 |

6. Promovari

| Input   | Output   |
|---|--|
| 2<br>Ion Cristian<br>Analist<br>4060<br>Robert Popescu<br>LiderEchipaProgramare<br>12600<br>8<br>4<br>Ion Cristian<br>Programator | Destructor Analist<br>Destructor Angajat<br>Ion Cristian<br>Programator<br>Robert Popescu<br>LiderEchipaProgramare |

7. Promovari v2

| Input  | Output  |
|--|---|
| 2<br>Ion Cristian<br>Analist<br>4060<br>Robert Popescu<br>LiderEchipaProgramare<br>12600<br>8<br>Mihai Mihailescu<br>Programator<br>5698<br>4<br>Mihai Mihailescu<br>LiderEchipaProgramare | Destructor Programator<br>Destructor Analist<br>Destructor Angajat<br>Ion Cristian<br>Analist<br>Robert Popescu<br>LiderEchipaProgramare<br>Mihai Mihailescu<br>LiderEchipaProgramare |