

Laborator 7. Mecanismul de mostenire

1. O bancă oferă împrumuturi tuturor persoanelor care doresc să cumpere apartamente sau produse de uz casnic. Un împrumut este identificat printr-un număr unic, date despre debitor (nume, prenume și adresă), și alte date despre împrumut, cum ar fi suma împrumutată (care este egală cu costul produsului), rata și frecvența de plată. Implicit, rata este egală cu 10% din costul produsului (apartament sau produs de uz casnic) și plata se face lunar. Debitorul poate alege o altă frecvență de plată în funcție de tipul împrumutului. Dacă debitorul dorește să cumpere un apartament, acesta poate să aleagă să plătească trimestrial, dar banca adaugă la rată un comision de 3% din suma pe care trebuie s-o plătească. Dacă debitorul vrea să cumpere un produs de uz casnic, el poate alege să plătească o dată la 6 luni, numai că, în acest caz, banca adaugă un comision de 7% la rată.

Programul ar trebui să știe și să afișeze suma pe care debitorul trebuie s-o plătească. La început, suma este egală cu împrumutul și este actualizată de fiecare dată când debitorul plătește o rată (i.e., metoda `platesteRata()` este apelată). Modelul programului este prezentat în Figura 1.

Cerințe:

1. Implementați clasele din model.
2. Pentru a testa aceste clase, scrieți clasa `TestImprumut` ce conține metoda `main()` și îndeplinește următoarele operații:
 - a. Presupunem că au fost făcute două împrumuturi: unul pentru un apartament și celălalt pentru un produs de uz casnic (adică, se vor crea două obiecte, câte unul din clasa `ImprumutApartament` și `ImprumutProdusUzCasnic`). În plus, presupunem că al doilea debitor a ales să plătească o dată la 6 luni.
 - b. Apelați metodele `calculeazaRata()` și `platesteRata()`.
 - c. Afișați informațiile despre cele două împrumuturi.
2. O agenție de asigurări oferă două tipuri de polițe de asigurare: de viață și în caz de accident. O poliță de asigurare are un număr unic, datele persoanei asigurate (nume, prenume și adresă) și alte date precum suma plătită de către asigurat, frecvența de plată și suma asigurată. Numărul poliței de asigurare este generat automat de către program.

Implicit, frecvența de plată este lunar și suma implicită este de 13€. Dar asiguratul poate alege când să plătească în funcție de tipul asigurării. Dacă persoana are o poliță de viață, poate alege să plătească trimestrial, dar agenția va adăuga un comision de 2% la sumă. Dacă persoana are o poliță în caz de accident, poate alege să plătească semestrial, dar agenția va adăuga un comision de 5%.

Suma asigurată este egală cu suma tuturor ratelor plătite de asigurat și este calculată de fiecare dată când persoana plătește o rată. Programul știe când a plătit o persoană, mai precis când este apelată metoda `calculeazaSuma()`.

Modelul programului este prezentat în Figura 2.

Cerințe:

- a. Implementați clasele din model.
- b. Pentru a testa aceste clase, scrieți clasa `TestAsigurare` ce conține metoda `main()` și îndeplinește următoarele operații:

1. Crearea unei polițe de asigurare de viață și una de accident (adică, un obiect al clasei `AsigurareViață` și celălalt din clasa `AsigurareAccident`). Presupunem că a doua persoană asigurată a ales să plătească de două ori pe an.
2. Apelarea metodelor `calculeazaRata()` și `calculeazaSuma()`.
3. Afișarea informațiilor despre cele două asigurări.

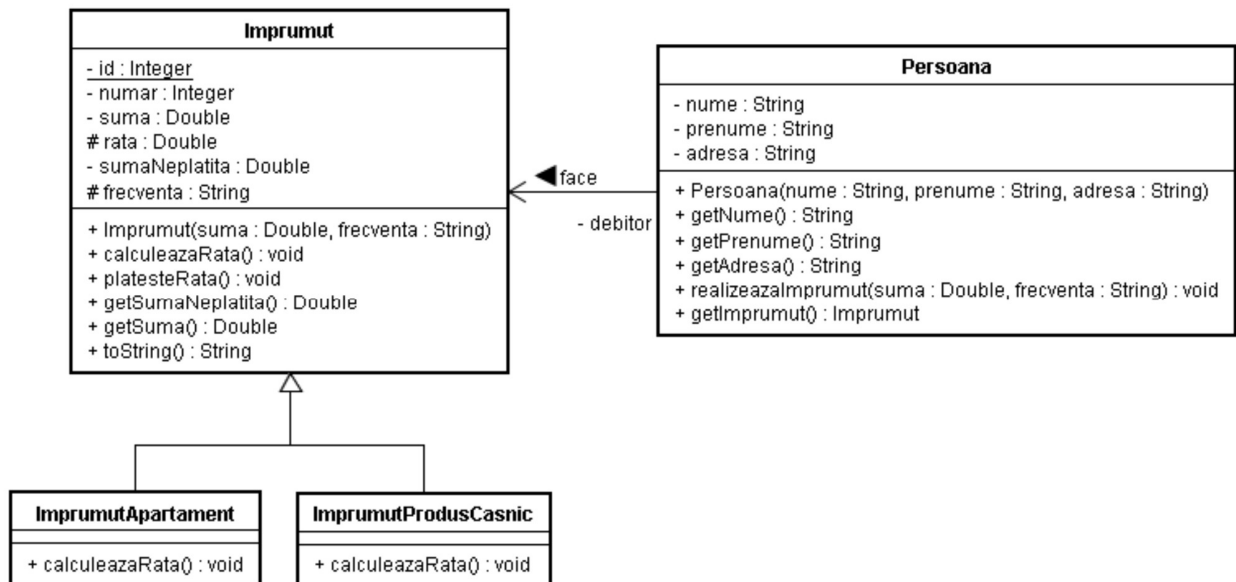


Figura 1. Diagrama UML de clase a problemei 1

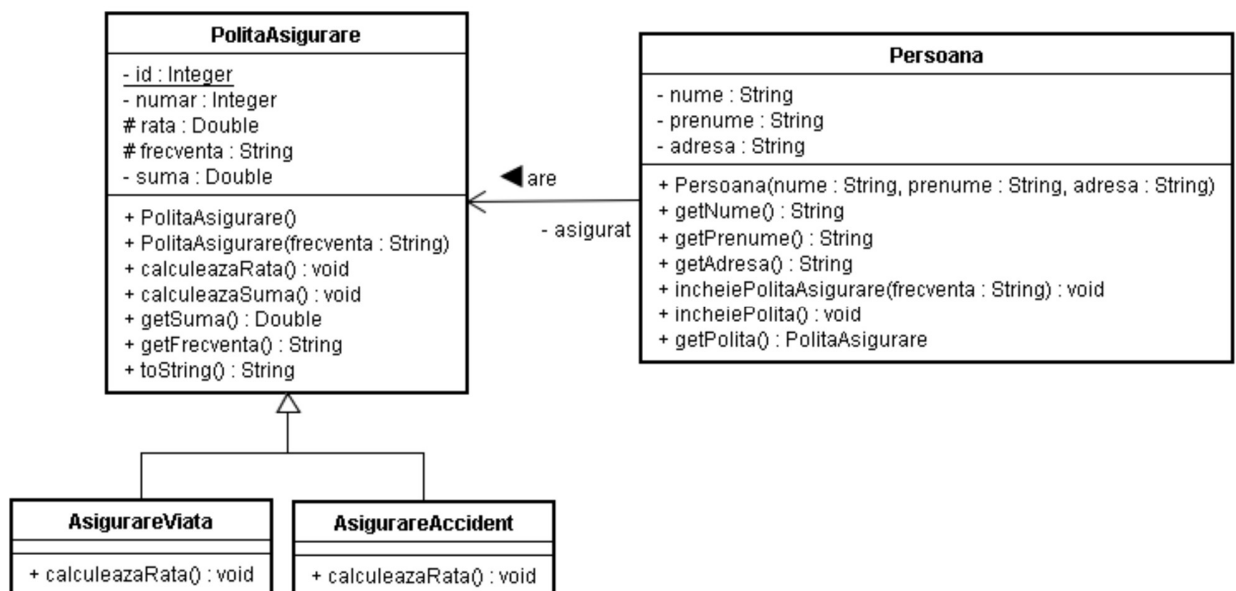


Figura 2. Diagrama UML de clase a problemei 2