

Exercitii

- Pentru a citi mai usor solutiile voastre va rog sa copiatii intr-un comentariu textul fiecarei probleme si sa il puneti la inceputul fiecarei rezolvari.

IMPORTANT

- Folositi OOP pentru a rezolva problemele
- Folositi exceptii.
- Folositi <https://app.diagrams.net/> pentru a realiza diagramele UML
 - Proiectati diagramele UML pe cat posibil INAINTEA redactarii programului.

Spor la treaba

Scrieti un program care va modela operatiunile unui POS.

Contul bancar

Va avea un ID de tip Guid

Va avea o metoda DepuneNumerar

Va avea o metoda ExtrageNumerar

- Va avea ca parametru suma ce se doreste a fi extrasa
- In cazul in care suma nu este disponibila, contul bancar va arunca o exceptie
- Va fi folosita la de Banca atunci cand vor fi efectuate plati

Banca

Va avea o lista de conturi curente indexate intr-un dictionar in functie de id-ul (Guid) acestora.

Va avea o metoda „CreeazaCont” care

- Va creea un nou cont bancar
- Il va adauga in lista conturilor
- Va returna id-ul contului

Va avea o metoda „EmiteCard” care va primi ca parametru id-ul contului

- In cazul in care contul nu exista va arunca o exceptie corespunzatoare
- In cazul in care au fost emise deja 2 carduri pentru acel cont va arunca o exceptie.
- Cardul emis va primi id-ul contului.

Va avea o metoda „Plateste” care va primi 2 parametri: suma si id-ul contului.

- In cazul in care contul nu exista va arunca o exceptie corespunzatoare

Card-ul

Va avea 3 metode:

- IntroduCard
 - Va afisa pe ecran un mesaj
- GetCardData
 - Returneaza id-ul contului
- ExtrageCard
 - Va afisa pe ecran un mesaj

POS-ul

Va avea o metoda „Plateste” care:

- va primi doi parametri: suma de plata si cardul
- va chema pe rand metodele
 - introdu card
 - get card data
 - va incerca sa efectueze plata in banca
 - va extrage cardul
- Se va asigura ca extragerea cardului a fost facuta si in situatia in care plata a esuat

Instantiati banca, creati conturi, depuneti bani in conturi, instantiati un POS, emiteti carduri si efectuati plati prin intermediul POS-ului

Definiti exceptiile, tratati exceptiile si afisati mesaje corespunzatoare

Suplimentar

Card-ul

Va avea propriul ID de tip guid si nu va mai contine id-ul bancii.

Contul

Va persista numarul de carduri emise.

Banca

La emiterea cardului

- Va memora intr-un dictionar id-ul contului corespunzator fiecarui id al cardului.
- In cazul in care au fost emise deja doua carduri pentru cont-ul cerut, nu va mai fi emis un nou card ci va fi aruncata o exceptie

Metoda „Plateste” va primi ca parametru ID-ul cardului si inainte de a efectua plata va incerca determinarea contului pe baza id-ului cardului

- Daca cardul nu poate fi gasit, va arunca o exceptie
- Daca contul nu poate fi gasit, va arunca o exceptie

Va avea o metoda „Connect”

- Va arunca o exceptie daca sunt mai mult de 3 conexiuni active.
- Va incrementa numarul de conexiuni active
- Va afisa un mesaj pe ecran , „Connected”

Va avea o metoda „Disconnect”

- Va decrementa numarul conexiunilor active
- Va afisa un mesaj pe ecran , „Disconnected”

Card-ul

Metoda GetCardData

- Va returna ID-ul cardului.

Pos-ul

Va avea o metoda privata „Connect” care

- Va incerca de 2 ori conectarea la Banca.
- In cazul in care conectarea a esuat, va arunca o exceptie corespunzatoare

Metoda Connect va fi apelata dupa introducerea cardului.

Dupa efectuarea platii, Pos-ul se va deconecta de la banca.

Instantiati clasele, apelati metodele.

Scrieti teste unitare pentru metoda „EmiteCard”