

CERINTE:

1. Creeati clasa *Vehicul* cu metodele abstracte *verificaMotor* si *verificaRoti*.
2. Definiti exceptiile custom *ProblemeMotorException* si *ProblemeRotiException* care afiseaza in consola mesajele "Alerta motor", respectiv "Alerta roti".
3. Creeati clasele *Bicicleta* si *Autovehicul*, care mostenesc clasa *Vehicul*.
4. Definiti clasa *PompaDeBenzina* cu atributul *cantitateDisponibila* de tip intreg. Utilizati paradigma singleton pentru aceasta clasa. Cantitatea initiala este X.
5. In clasa *Vehicul* definiti metoda cu tipuri generice *pornesteAutovehicul*. Metoda se poate apela numai pentru autovehicule. Metoda efectueaza verificarea motorului si verificarea rotilor. Daca motorul sau rotile au probleme, se arunca exceptiile corespunzatoare, definite la punctul 2.
6. Definiti clasele *Autoturism* si *Camion*, stiind ca autoturismul are 4 roti, iar camionul 6. Autoturismul si camionul trebuie sa poata alimenta de la *PompaDeBenzina*, iar cantitatea disponibila trebuie sa scada dupa fiecare alimentare.
7. Definiti firele de executie *AutoturismThread* si *CamionThread* prin extinderea clasei *Thread*. Fiecare fir de executie trebuie sa apeleze metoda *pornesteAutovehicul*. In cadrul fiecarui fir de executie, autovehiculul trebuie alimentat cu benzina. Asigurati-vă ca firele de executie nu vor arunca exceptii.

8. Adaugati clasa *Main*. In metoda *main* instantiatii alternativ cate un *Autoturism* si *Camion* pornind alternativ cate un fir de executie *AutoturismThread* si *CamionThread*, pana cand se termina *cantitateaDisponibila* din *PompaDeBenzina*. Dupa ce se termina, afisati "Nu mai e benzina". Folositi valori random pentru numarul de roti al fiecarui autoturism (intre 1 si 4) si camion (intre 1 si 6). Stim ca un autoturism alimenteaza cu 3 si un camion cu 5.