

## CERINTE:

1. Creați clasa *Vehicul* cu metodele abstracte *verificaMotor* și *verificaRoti*.
  2. Definiți excepțiile custom *ProblemeMotorException* și *ProblemeRotiException* care afișează în consolă mesajele “Alerta motor”, respectiv “Alerta roți”.
  3. Creați clasele *Bicicleta* și *Autovehicul*, care mostenesc clasa *Vehicul*.
  4. Definiți clasa *PompaDeBenzina* cu atributul *cantitateDisponibila* de tip întreg. Utilizați paradigma singleton pentru această clasă. Cantitatea inițială este X.
  5. În clasa *Vehicul* definiți metoda cu tipuri generice *pornesteAutovehicul*. Metoda se poate apela numai pentru autovehicule. Metoda efectuează verificarea motorului și verificarea roților. Dacă motorul sau roțile au probleme, se aruncă excepțiile corespunzătoare, definite la punctul 2.
  6. Definiți clasele *Autoturism* și *Camion*, știind că autoturismul are 4 roți, iar camionul 6. Autoturismul și camionul trebuie să poată alimenta de la *PompaDeBenzina*, iar cantitatea disponibilă trebuie să scadă după fiecare alimentare.
  7. Definiți firele de execuție *AutoturismThread* și *CamionThread* prin extinderea clasei *Thread*. Fiecare fir de execuție trebuie să apeleze metoda *pornesteAutovehiculul*. În cadrul fiecărui fir de execuție, autovehiculul trebuie alimentat cu benzină. Asigurați-vă ca firele de execuție nu vor arunca excepții.
- 
8. Adăugați clasa *Main*. În metoda *main* instanțiați alternativ câte un *Autoturism* și *Camion* pornind alternativ câte un fir de execuție *AutoturismThread* și *CamionThread*, până când se termină *cantitateaDisponibila* din *PompaDeBenzina*. După ce se termină, afișați “Nu mai e benzină”. Folosiți valori random pentru numărul de roți al fiecărui autoturism (între 1 și 4) și camion (între 1 și 6). Știm că un autoturism alimentează cu 3 și un camion cu 5.