

Exercitii PCLP2 – L08

EX1.

Creați clasa **Persoana** cu proprietățile **nume**, **varsta**.

Creați clasa **StudentLicenta** cu proprietățile **notaAdmitereLiceu**, **note** și clasa **StudentMaster** cu proprietățile **notaLicenta**, **note**.

Verificați dacă clasa **Persoana** poate fi moștenită de clasele **StudentLicenta** și **StudentMaster**.

Dacă da, atunci implementați moștenirea.

Creați un obiect de tip **StudentLicenta** și afișați datele sub forma:

Studentul <nume>, in varsta de <varsta> ani, este student la licenta avand nota de admitere <notaAdmitereLiceu> si media in anul curent: <media>.

Creați un obiect de tip **StudentMaster** și afișați datele sub forma:

Studentul <nume>, in varsta de <varsta> ani, este student la master avand nota proiectului de diploma <notaLicenta> si media in anul curent: <media>.

EX2.

Să se scrie un program Java care permite utilizatorului calcularea lungimii unui itinerariu prin țară specificat de utilizator. Programul va oferi un meniu cu opțiuni pentru:

a. Adăugare oraș;

Clasa **Oras** moșteneste clasa **Localitate**.

Clasa **Localitate** are proprietățile: **nume**, **nrLocuitori**, **judet**, **coordonate** (**longitudine**, **latitudine**).

Clasa **Oras** are proprietățile: **esteMunicipiu**, **esteResedinta**, **nrMonumente**.

Programul va cere numele orașului, numărul de locuitori, județul din care face parte, longitudinea și latitudinea, dacă este sau nu municipiu, dacă este sau nu reședință de județ și numărul de monumente UNESCO din administrare.

b. Eliminare oraș după nume. Numele orașului este furnizat de utilizator și identificarea acestuia în listă se va face fără distincție între litere mari și litere mici.

c. Afișarea lungimii itinerariului.

Pentru calculul distanțelor se vor folosi următoarele valori: 1 grad latitudine = 111 km, 1 grad longitudinal = 79 km și se vor considera rute liniare între orașe. Se va respecta ordinea orașelor.

Programul va fi realizat cu respectarea convențiilor specifice limbajului Java de denumire și organizare a pachetelor, claselor proprietăților și metodelor, precum și a principiului de încapsulare specifice programării orientate pe obiecte.