

**Enunț**

O un producător de masini dorește să pornească o afacere și în acest sens el dorește să creeze un sistem de gestionare a modelelor de autoturisme pe care le fabrică. Producătorul dorește să realizeze aplicația în C++ folosind concepte de OOP.

Producătorul își propune să creeze mașini care funcționează pe bază de combustibili fosili (care pot fi de două tipuri: benzină și motorină), mașini electrice și mașini hibride (care pot fi alimentate atât electric, cât și cu benzină/motorina)

Despre un model de mașină, producătorul dorește să rețină următoarele informații: anul începerii producției, numele, modelul, viteza maximă și greutatea.

În plus, pentru un model de mașină pe bază de combustibili fosili se mai rețin: tipul combustibilului (benzină sau motorină) și capacitatea rezervorului. Pe de altă parte, pentru un model electric se reține adițional: capacitatea bateriei.

Autonomia unei mașini dintr-un model se calculează:

- pentru modele pe combustibili fosili: raportul dintre capacitatea rezervorului și radical de ordinul doi din greutate
- pentru modele electrice: raportul dintre capacitatea bateriei și greutate la pătrat
- pentru modele hibride: suma celor două definite mai sus

Producătorul dorește de asemenea de la program să memoreze o listă a tranzacțiilor efectuate de-a lungul timpului cu diverși clienți. O tranzacție este definită prin: numele clientului, data tranzacției și o listă de modele achiziționate.

Un exemplu de tranzacție: {Popescu, 5 5 2019, [Masina\_1, Masina\_2, Masina\_3]}, unde Masina\_i sunt obiecte/pointeri/referințe din clase reprezentând modele, **NU** nume de modele.

**Cerințe:**

- I. Să se implementeze clasele necesare astfel încât aplicația să funcționeze.
- II. Programul este prevăzut cu un meniu interactiv în care administratorul are posibilitatea de a alege dintre următoarele opțiuni:
  1. Adaugă un nou model mașină citind de la tastatură tipul mașinii și apoi câmpurile specifice ei
  2. Adaugă o tranzacție
  3. Afișează cel mai vândut model
  4. Afișează modelul cu autonomia cea mai mare
  5. Aduce o optimizare unui anumit model crescându-i viteza maximă cu un anumit procent.

**Precizări**

1. Timpul de lucru este de 90 de minute.
2. La sfârșitul timpului de lucru, studenții vor încarca pe Moodle fișierul sursă cu extensia cpp. Acesta trebuie să conțină pe primul rând un comentariu cu numele și prenumele studentului, grupa și compilatorul folosit.
3. Sursa predată trebuie să compileze. Sursele care au erori de compilare nu vor fi luate în considerare. Înainte de predarea surselor, studenții vor pune în comentariu eventualele părți din program care au erori de compilare sau nu funcționează corespunzător.
4. Se acceptă și soluții parțiale, care nu respectă toate cerințele din enunț, dar sunt funcționale. Acestea vor fi depunctate corespunzător.
5. În implementarea programului se vor utiliza cât mai multe dintre noțiunile de programare orientată pe obiecte, care au fost studiate pe parcursul semestrului și care se potrivesc cerințelor din enunț.
6. Condițiile minimale de promovare a testului sunt ca programul să fie scris cu clase și să citească modelele de mașini unul câte unul, să le memoreze datele și apoi să le afișeze.
7. Orice tentativă de fraudă se va pedepsi conform regulamentelor Universității.