

Trotineta Electrica

Nota 1

Să se creeze un proiect în Microsoft Visual Studio care să nu genereze erori de compilare.

Nota 2

Pentru clasa **TrotinetaElectrica**, se au în vedere câmpurile:

- **model** – șir de caractere, string;
- **nivelBaterie** – nivelul bateriei (float)
- **consumPerKm** – valoare reală constantă ce indică consumul per km (unități de energie) ;
- **kmPerSesiuneInchiriere** – vector alocat dinamic cu valori reale ce înregistrează km parcurși pe sesiunea de închiriere;
- **nrSesiuniInchiriere** – întreg care indică numărul de sesiuni de închiriere
- **NIVEL_MAXIM_BATERIE** – atribut static întreg ce reprezintă limita maximă de energie ce poate fi stocată de baterie (unități de energie)

Atributele clasei se definesc în zona privată a clasei.

Să se implementeze **constructorul implicit** și metodele **getModel()**, **getNrSesiuniInchiriere()**, **getNivelBaterie()** și să se testeze în main(). Atributele clasei nu trebuie lăsate neinițializate.

Nota 3

Să se definească pentru clasa **TrotinetaElectrica** **constructorul cu parametri** și **destructorul**. Constructorul cu parametri conține validări pentru parametrii primiți, conform descrierilor atributelor. Să se testeze constructorul în main(). Destructorul se implementează astfel încât să nu genereze memory leak-uri.

1 punct Să se implementeze metoda **utilizeazaTrotineta()**. Metoda trebuie să adauge o nouă sesiune de utilizare a trotinetei cu valoarea indicată pentru km parcurși.

1 punct Să se definească pentru clasă **constructorul de copiere**

1 punct Să se implementeze metoda **incarcaTrotineta()** și metoda **kmDeParcurs()**. Metoda **incarcaTrotineta()** permite încărcarea trotinetei și primește ca parametri minutele (int) și rataIncarcarePerMinut (float). Se va actualiza nivelul bateriei în funcție de durata încărcării și rata acesteia. Valorile primite trebuie să fie pozitive, în caz contrar se afișează mesaj în consolă de avertizare. Metoda **kmDeParcurs()** verifică câți km mai pot fi parcurși cu trotineta.

1 punct Să se supraîncarce pentru clasă **operatorul de atribuire (=)** fără a genera memory leaks.

1 punct Să se supraîncarce **operatorii << și >>** pentru citirea și afișarea obiectelor de tipul **TrotinetaElectrica**. Elementele sunt afișate pe aceeași linie despărțite prin spațiu.

1 punct Să se supraîncarce **operatorul +** pentru a încărca o nouă sesiune de utilizare a trotinetei. Operatorul trebuie să permită o operație de tipul **10 + trotineta**.

1 punct Să se supraîncarce **operatorul funcție** pentru a permite determinarea sumei km parcurși pentru mai multe sesiuni de închiriere al căror index este dat. Dacă valoarea indexului nu este corectă se scrie mesaj în consolă. Exemplu utilizare **trotineta(2,4)** va returna suma km pentru sesiunile 2, 3 și 4.