

Test PC2 - subiect 1 – dec. 2024

Să se implementeze următoarele:

(0.5p) O clasă *Jucarie* cu 2 membri:

- Un membru **nume** de tip **string**
- Un membru **pret** de tip **double**

(0.5p) O clasă *Camion* **derivată** din clasa *Jucarie* ce are un membru propriu:

- Un membru **numarBaterii** de tip **unsigned int**

Pentru clasa *Jucarie* se vor implementa următoarele metode:

- (0.5p) Constructor de inițializare (cu listă de parametri) care va inițializa membrii clasei cu valorile parametrilor
 - Acest constructor va permite și apelarea sa fără parametri, caz în care membrii se vor inițializa cu valori predefinite
- (0.5p) Metoda de afișare a tuturor datelor referitoare la jucărie

Pentru clasa *Camion* se vor implementa următoarele metode:

- (0.5p) Constructor de inițializare (cu listă de parametri) care va inițializa membrii clasei cu valorile parametrilor și apelează un constructor din clasa de bază pentru inițializarea membrilor moșteniți
 - Acest constructor va permite și apelarea sa fără parametri, caz în care membrii se vor inițializa cu valori predefinite
- (0.5p) Metoda de afișare a tuturor datelor referitoare la camion

Cerințe:

- (0.4p) Creați 2 obiecte de tip *Jucarie* pentru a testa cele două moduri de apelare a constructorului din clasă.
 - (0.1p) Pentru aceste două obiecte apelați funcția de afișare.
- (0.4p) Creați 2 obiecte de tip *Camion* pentru a testa cele două moduri de apelare a constructorului din clasă.
 - (0.1p) Pentru aceste două obiecte apelați funcția de afișare.

Barem

1. Clasa Jucărie	
1.1. Definirea corectă a datelor membre	0.5
1.2. Constructor	0.5
1.3. Metoda de afișare	0.5
2. Clasa Camion	
2.1. Implementarea corectă a conceptului de moștenire + date membre	0.5
2.2. Constructor	0.5
2.3. Metoda de afișare	0.5
Crearea celor 2 obiecte Jucărie (0.4) + Afișare (0.1)	0.5
Crearea celor 2 obiecte Camion (0.4 + Afișare (0.1)	0.5
Oficiu	1
TOTAL	5

VEZI ȘI PAGINA 2!!!

NUMAI DUPĂ implementarea cerințelor de pe pagina 1, rezolvați cerințele următoare:

1.
 - a. (0.5p) Supraîncărați un **operator binar** care compară două obiecte *Camion* din punctul de vedere al numelui acestora.
 - b. (0.5p) Testați și demonstrați funcționarea operatorul scris în funcția main.
2. Creați o nouă clasă **TIR derivată** din clasa *Camion* ce va avea:
 - Un membru **numarAxe** de tip unsigned int
 - (0.25p) Constructor de inițializare
 - (0.25p) Metoda de afișare
3.
 - a. (0.5p) Adăugați în funcția main un vector din biblioteca standard care să permită adăugarea a **patru** obiecte de tip *TIR* . Se pot folosi obiectele existente sau se pot adăuga altele noi.
 - b. (0.5p) Parcurgeți vectorul și afișați obiectele din acesta. Pentru fiecare obiect din vector trebuie să se afișeze toate proprietățile sale.
 - c. (0.5) Sortați vectorul după orice criteriu doriți utilizând funcția **sort** din biblioteca standard C++, dar menționați într-un comentariu care este acesta.
4. (1p) Dați un exemplu de utilizare a unui **operator=** supraîncărcat printr-o funcție proprie pentru una din cele 3 clase.
5. (1p) Dați un exemplu de utilizare a **pointerilor smart** pentru orice obiect sau tip de date definit anterior.

Observații:

- Se va utiliza specificatorul de acces “public” cel mult pentru metodele claselor (nerespectarea acestei prevederi atrage după sine înjumătățirea punctajului acordat).
- Fiecare clasa va avea headerul + sursele proprii, iar definirea metodelor se va realiza în fisierul sursă (.cpp)
- Clasele se pot completa, la nevoie, cu alte metode ajutătoare
- Numele membrilor claselor poate fi modificat dacă se dorește utilizarea unor convenții de notare