

Test PII - subiect 2 – dec. 2020

Să se implementeze următoarele:

O clasă Electrocasnic ce conține:

- Un membru **producător** de tip **string**
- Un membru **pret_fara_TVA** de tip **double**
- Metoda Afișare()

O clasă Frigider **derivată** din clasa Electrocasnic ce conține:

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

Pentru clasa Electrocasnic se vor implementa următoarele metode:

- Constructor fără argumente care inițializează datele de tip string cu un șir vid iar pe cele double cu 0
- Constructor de inițializare (cu listă de parametri) care va inițializa membrii clasei cu valorile parametrilor
- Destructor ce afișează un mesaj la apelarea sa
- Metoda de afișare a tuturor datelor referitoare la electrocasnic

Pentru clasa Frigider se vor implementa următoarele metode:

- Constructor fără argumente care inițializează datele de tip string cu un șir vid, iar cele de tip numeric cu 0
- 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
- Destructor ce afișează un mesaj la apelarea sa
- Metoda de afișare a tuturor datelor referitoare la frigider

Cerințe:

- Creați 2 obiecte de tip Electrocasnic pentru a testa cei doi constructori ai clasei.
 - Pentru aceste două obiecte apeleți funcția de Afișare
- Creați 2 obiecte de tip Frigider pentru a testa cei doi constructori ai clasei.
 - Pentru aceste două obiecte apeleți funcția de Afișare

Barem

1. Clasa Electrocasnic	
1.1. Definirea corectă a datelor membre	0.25
1.2. Constructor fără argumente + constructor cu argumente + destructor	0.5
1.3. Metoda de afișare	0.25
2. Clasa Frigider	0
2.1. Implementarea corectă a conceptului de moștenire + date membre	0.25
2.2. Constructor fără argumente + constructor cu argumente + destructor	0.5
2.3. Metoda de afișare	0.25
Crearea celor 2 obiecte Electrocasnic + Afișare	1
Crearea celor 2 obiecte Frigider + Afișare	1
Oficiu	1
TOTAL	5

Vezi și pagina 2!

NUMAI DUPĂ implementarea cerințelor de pe pagina 1 vă puteți apuca de cerințele de mai jos:

1.
 - a. (0.5p) Supraîncărcați un operator unar care mărește prețul fără TVA al unui obiect de tip frigider cu 100 de lei
 - b. (0.5p) Testați și demonstrați funcționarea operatorul scris în funcția main.
2.
 - a. (0.25p) Scrieți o funcție membră clasei Electrocasnic ce schimbă prețul acestuia cu un preț nou primit ca parametru.
 - b. (0.25p) Apelați această funcție pentru un obiect de tip Frigider declarat anterior.
3. (0.5 p) Creați o nouă clasă **Aspirator derivată din clasa Electrocasnic** ce va avea:
 - Un membru **putere** de tip unsigned int
 - Constructor cu argumente
 - Metoda de afișare
 - a. (1p) Creați un vector cu 4 obiecte de tip Aspirator și afișați datele corespunzătoare lor. Se va folosi tipul de dată vector din biblioteca standard, adică std::vector
 - b. (1p) Sortați elementele vectorului utilizând funcția sort. Se pot sorta elementele după ce criteriu doriți.
4. (1p) Definiți un template la funcție și utilizați-l pentru a afla maximul dintre două obiecte de tip Aspirator din punctul de vedere al puterii. Testați template-ul scris.

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 fișierul 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