Test PII - subiect 3 – dec. 2020

Să se implementeze următoarele:

O clasă Gadget ce conține:
- Un membru **producător** de tip **string**
- Un membru **pret_fara_TVA** de tip **double**
- Metoda Afișare()

O clasă Smartphone **derivată** din clasa Gadget ce conține:
- Un membru **diagonală** de tip **unsigned int**

Pentru clasa Gadget 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 gadget

Pentru clasa Smartphone 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 smartphone

Cerințe:
- Creați 2 obiecte de tip Gadget pentru a testa cei doi constructori ai clasei.
  - Pentru aceste două obiecte apelați funcția de Afișare
- Creați 2 obiecte de tip Smartphone pentru a testa cei doi constructori ai clasei.
  - Pentru aceste două obiecte apelați funcția de Afișare

Barem

1. Clasa Gadget	
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 Smartphone	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 Gadget + Afișare	1
Crearea celor 2 obiecte Smartphone + 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 binar care compară două Smartphone-uri din punctul de vedere al diagonalei.
    b. (0.5p) Testați operatorul scris în funcția main.
2.
    a. (0.25p) Scrieți o funcție membră clasei Gadget ce schimbă prețul acestuia cu un preț nou primit ca parametru.
    b. (0.25p) Apelați această funcție pentru un obiect de tip Gadget declarat anterior.
3. (0.5 p) Creați o nouă clasă Smartwatch **derivată** din clasa Gadget ce va avea:
    - Un membru **autonomie** de tip unsigned int
    - Constructor cu argumente
    - Metoda de afișare

    a. (1p) Creați un vector cu 4 obiecte de tip Smartwatch 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, crescător după prețul fără TVA.
4. (1 p) Adăugați în clasa Smartwatch un membru static privat cu ajutorul căruia să contorizați numărul instanțelor create și o metodă statică pentru a returna valoarea acestui membru. Afișați valoarea membrului de tip contor în funcția main.

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