

TEST PRACTIC – Pregătire

OBSERVAȚII: Se depunțează **DOUĂ PUNCTE** lipsa de separare a definițiilor și implementărilor în fișierele header și sursă pentru fiecare clasă în parte.

1p - Oficiu

Cerinte:

1. **(1p) Se cere implementarea clasei de bază *Senzor*** care conține **3 câmpuri private** (un *nume* de tip ***string***, un *tip* de tip ***string*** și *pret* de tip ***double***), iar ca metode:
 - Constructor fără parametri care inițializează câmpurile.
 - Constructor cu parametri (implementat prin listă de inițializare). **Nu se vor combina cei doi constructori într-un singur constructor.**
 - Funcție de afișare.
 - Supraîncărcarea operatorului `==` pentru a putea compara doi senzori de același tip (implementat prin intermediul unei funcții globale)
 2. **(1.5p) Se cere implementarea clasei derivată *Termocuplu*** care conține **2 câmpuri private** (*toleranța* de tip ***double*** și *valoare* de tip ***double***), iar ca metode:
 - Constructor fără parametri care inițializează câmpurile (se va utiliza constructorul clasei de bază).
 - Constructor cu parametri (implementat prin listă de inițializare). **Nu se vor combina cei doi constructori într-un singur constructor.**
 - Funcție de afișare.
 - Supraîncărcarea operatorului `<<` pentru afișare (va apela în interior funcția creată anterior).
 3. **(2.5p) Se cere implementarea clasei *Vehicul*** care conține **2 câmpuri private** (*nrltems* de tip ***unsigned int*** și *senzoristică* de tip ***vector< Termocuplu >***), iar ca metode:
 - Constructor cu parametri (cu valori implicite) care inițializează câmpurile
 - O metodă de adăugare a unui element.
 - O metodă de scoatere a unui element.
 - O metodă de sortare a elementelor în funcție de nume.
 - O funcție de afișare a tuturor elementelor din vector.
 - O funcție de determinarea a **temperaturii medii** a senzorilor, precizând și **gradul de încredere** în funcție de toleranța medie (dacă un senzor are toleranța ± 5 , atunci gradul de încredere este de 95%).
 4. **(0.5p) Implementarea unei variabile statice private (în oricare dintre clasele menționate anterior).**
 5. **(1p) Implementarea unui template la funcție de care să permită o apelare de tipul:**
 - (pseudocod) `Senzor s = combine(Senzor w, double d);`
 - (pseudocod) `Termocuplu t = combine(Termocuplu k, double s);`
-

În main:

1. **(0.25p)** Adăugarea a două obiecte de tip **Senzor** prin apelarea constructorilor definiți și afișarea acestora.
2. **(0.5p)** Adăugarea a 4 obiecte de tip **Termocuplu** prin apelarea constructorilor definiți și afișarea acestora (prin intermediul operatorului supraîncărcat).
3. **(0.25p)** Demonstrarea supraîncărcării operatorului de comparare din **Senzor**.
4. **(0.75p)** Crearea unui vector de **Termocuplu**, popularea acestuia cu cele 4 elemente definite anterior și adăugarea lui într-o instanță a clasei **Vehicul**. După asta se va scoate un element din vector și se vor adăuga altele două la final. Se va afișa statusul vectorului din obiectul de tip **Vehicul** la fiecare pas.
5. **(0.25p)** Calculul și afișarea temperaturii medii a senzorilor din vehicul.
6. **(0.25p)** Demonstrarea utilizării variabilei statice.
7. **(0.25p)** Apelarea funcției template **combine**.