

TEST PRACTIC – Pregătire

OBSERVAȚII: Se depun puncte 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 *Item*** care conține **3 câmpuri private** (un *nume* de tip ***string***, un *type* de tip ***string*** și *value* 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 combina doua obiecte (implementat prin intermediul unei funcții globale)
 2. **(1.5p) Se cere implementarea clasei derivată *Weapon*** care conține **2 câmpuri private** (*damage* de tip ***unsigned int*** și *weight* 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.
 - Funcție de calcul a **DPS**-ului (damage per second) știind că valoarea **damage** a câmpului este valoarea per lovitură, iar frecvența loviturilor este de o lovitură la 2.5 secunde.
 - Supraîncărcarea operatorului << pentru a afișare (va apela în interior funcția creată anterior).
 3. **(2.5p) Se cere implementarea clasei *Inventar*** care conține **2 câmpuri private** (*nrlItems* de tip ***unsigned int*** și *storage* de tip ***vector<Weapon>***), 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 DPS-ului total al tuturor armelor din inventar.
 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 fuziunea obiectelor și să permită o apelare de tipul:**
 - (pseudocod) `Weapon v = combine(Weapon w, double s);`
 - (pseudocod) `Item i = combine(Item k, double s);`
-

În main:

1. **(0.25p)** Adăugarea a două obiecte de tip **Item** prin apelarea constructorilor definiți și afișarea acestora.
2. **(0.25p)** Adăugarea a două obiecte de tip **Weapon** prin apelarea constructorilor definiți și afisarea acestora.
3. **(0.5p)** Demonstrarea supraîncărcării operatorului de adunare din **Item**.
4. **(0.5p)** Crearea unui vector de **Weapon**, popularea acestuia și adăugarea lui într-o instanță a clasei **Inventar**. 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 **Inventar** la fiecare pas.
5. **(0.25p)** Calculul și afișarea DPS-ului total al elementelor din inventar.
6. **(0.25p)** Demonstrarea utilizării variabilei statice.
7. **(0.5p)** Apelarea funcției template **combine**.