# Moștenire simplă

Azi ninja-ul nostru va dori să învețe ce este aceea moștenire în C++. Ca atare și-a luat uniforma albă la purtător pentru că azi își va studia sabia și va înțelege că face parte din categoria armelor specifice antichității.


NullPointerNinja vrea să învețe ceea ce i-a fost promis acum câteva săptămâni și anume să înțeleagă ce este specificatorul de acces protected și cum se poate aplica acesta pentru membri unei clase. Vrea de asemenea să înțeleagă relația de tip **„is a”** între două clase și cum se stabilește cine va fi superclasa și cine va avea rolul de subclasă. Așadar vom considera următoarele clase în limbajul C++:

```
class Arma
{
    double pret;
    char* material;
};

class Sabie
{
    int lungime;
    char* tipLama;
};
```

## Cerințe

1) Să se stabilească corect relația de tip „is a” între cele două clase.

2) Să se implementeze constructorii fără parametri și cei cu toți parametrii pentru cele 2 clase.

3) Să se implementeze constructorii de copiere și operatorii de asignare pentru ambele clase (deep copy).

4) Să se implementeze corect destructorii pentru a putea asigura corect eliberarea resurselor celor 2 clase pentru membrii alocați dinamic.

5) Să se implementeze operatorii de afișare pentru cele 2 clase.

6) În main să se testeze relația de tip „is a” prin crearea unui obiect de tip Sabie și apelarea metodelor (se pot implementa pentru demonstrație getteri și setteri).

7) Să se creeze un vector alocat dinamic care conține 5 săbii.

8) Să se sorteze acest vector descrescător după câmpul lungime. (Se va supraîncărca operatorul de comparație <).

9) Eliberați memoria ocupată de acest vector.