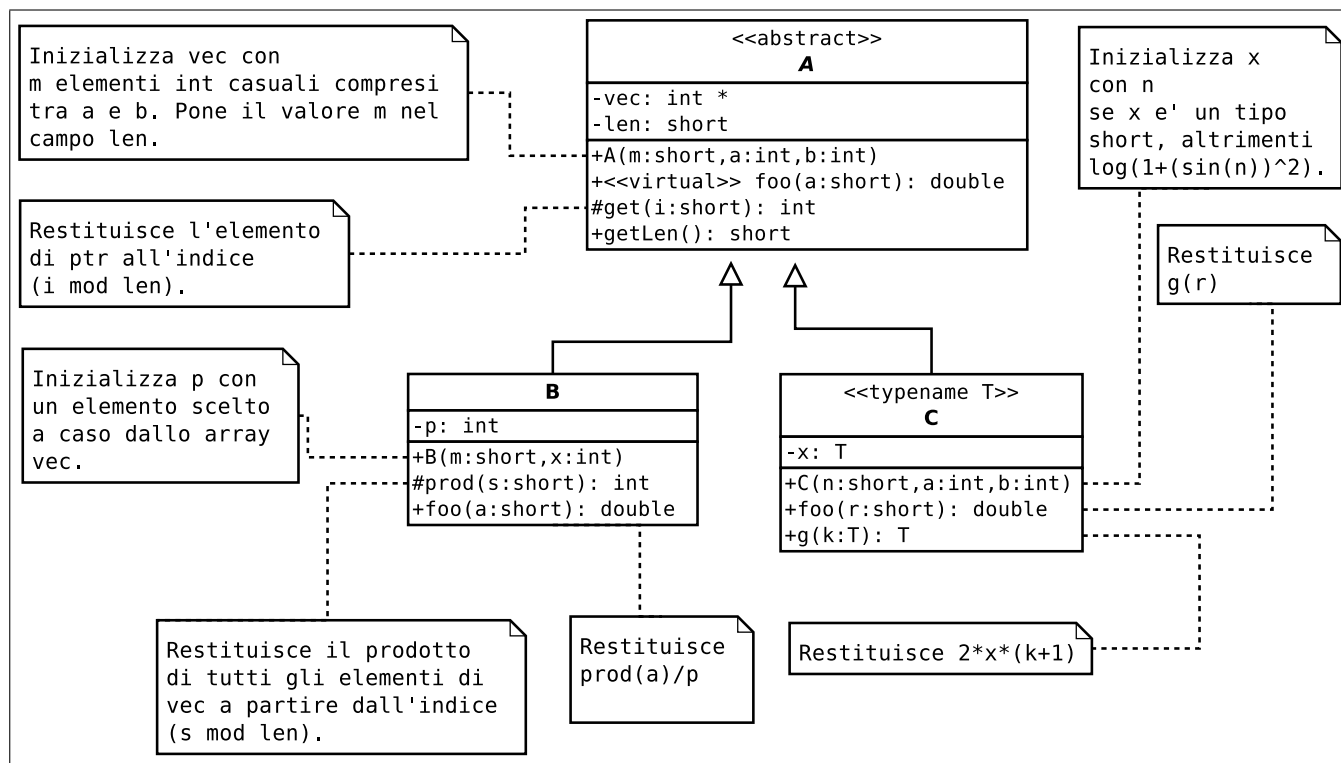


Si implementi in C++ la gerarchia ereditaria descritta dal seguente diagramma UML delle classi. NB: È necessario implementare tutti e soli i metodi indicati nel diagramma.



In un main, si generi una collezione di 50 oggetti utilizzando la sequenza di valori casuali riportata sul retro del foglio. **NB:** È possibile scaricare il frammento di codice da inserire nel main a partire dalle URL indicate sul retro del foglio

Successivamente:

1. si visualizzi la collezione mediante l'overloading dell'operatore <<, ad esempio:

31)1CIde, vec=[4 5 9 5 9 9 9 5 7 6], x=0.259251, foo(3)=2.07401

32)1B, vec=[8 14 6 9 9 9 5 8 12 11], p=9, foo(3)=0

33)1CIse, vec=[7 5], x=2, foo(3)=16

2. si calcoli il massimo valore foo(3) per tutti gli oggetti della collezione e la media dei valori g(5) per per tutti gli oggetti di tipo C<double>;
3. si implementi l'overloading dello operatore membro "[[]]" per la classe A e lo si utilizzi nel main in una istruzione di assegnamento.

```
srand(111222333);

A *vett [DIM];

for(int i=0; i<DIM; i++) {
    short n=1+rand()%10;
    switch(rand()%3) {
        case 0:
            vett[i]= new B(n, rand()%5 + 1, rand()%10 + 5);
            break;
        case 1:
            vett[i]= new C<double>(n, rand()%5 + 1, rand()%10 + 5);
            break;
        case 2:
            vett[i]= new C<short>(n, rand()%5 + 1, rand()%10 + 5);
    }
}
```

1. Frame di codice da inserire nella funzione main:

www.dmi.unict.it/~messina/didat/prog1_18_19/30_04_2019/frame-30_04.cpp

oppure Short URL:

<https://tinyurl.com/y3q9ttgy>