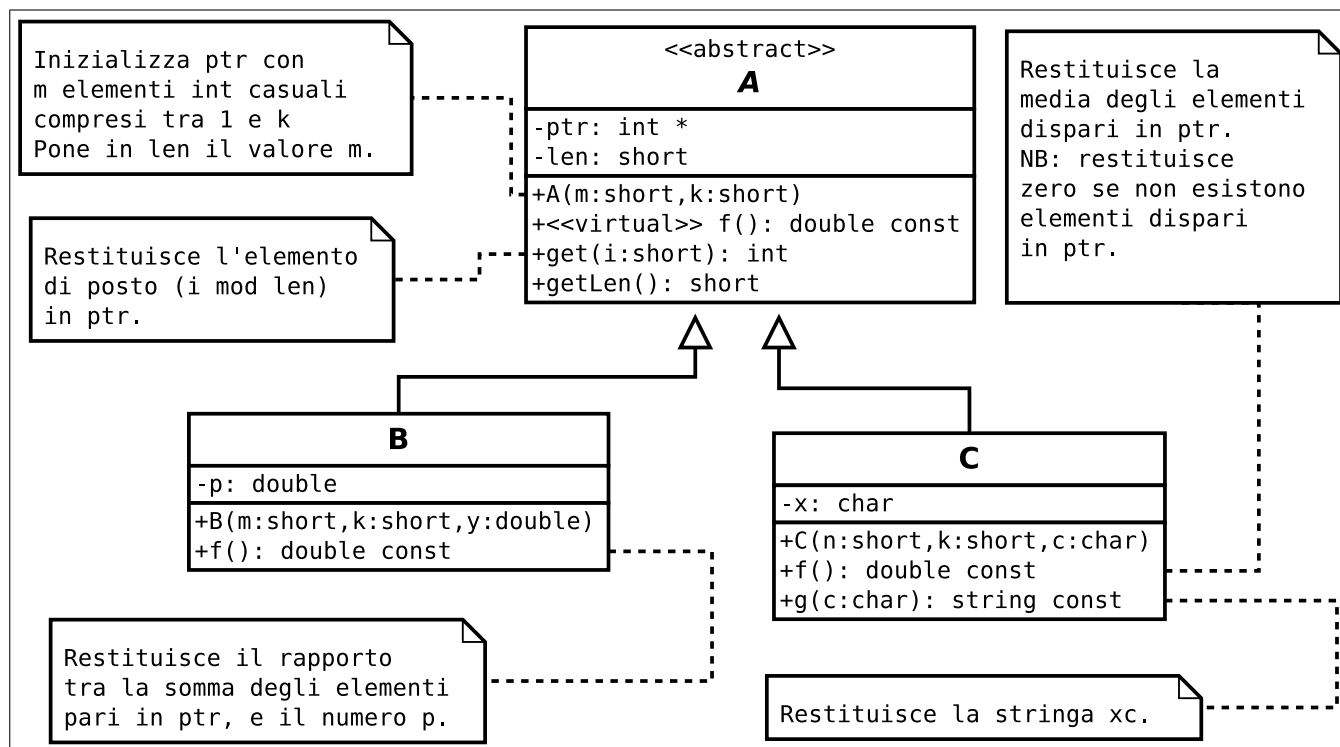


Si implementi in C++ la gerarchia ereditaria descritta dal seguente diagramma UML delle classi. NB: È necessario implementare tutti e soli i metodi indicati nel diagramma.



In un main, si generi una collezione di 50 oggetti utilizzando la sequenza di valori casuali riportata sul retro del foglio. **NB:** È inoltre possibile (e consigliato) prelevare il frammento di codice da inserire nel main a partire dalla URL indicata sul retro del foglio

Successivamente:

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

```
35)1B, ptr=[ 3 4 6 ], p=0.344766, f()=29.0052
36)1C, ptr=[ 1 1 3 2 2 5 1 2 1 ], x=j, f()=2
37)1C, ptr=[ 4 3 3 2 4 3 6 ], x=s, f()=3
```

2. si calcoli la media dei valori `f()` per tutti gli oggetti della collezione e si visualizzi la stringa concatenazione di tutte le stringhe restituite da `g('h')` per tutti gli oggetti di tipo C;
3. si implementi l'overloading dello operatore membro `[]` per la classe A e lo si utilizzi nel main in una operazione di assegnamento per un qualche oggetto della collezione.

NB: Salvare il codice sorgente nel seguente modo: **NOME.COGNOME.MATRICOLA.cpp**, quindi INVIARE IL CODICE SORGENTE MEDIANTE IL SEGUENTE (SHORT) URL: <https://goo.gl/P4toJK>.

Output di controllo:

- utente Windows: [vedi url sul retro del foglio]

```
srand(111222333);
A *vett [DIM];

for(int i=0; i<DIM; i++){
    short n=1+rand()%10;
    short m = 1+rand()%8;
    if(rand()%2==0)
        vett[i]= new B(n, m, rand()/(double) RANDMAX+0.05);
    else
        vett[i]= new C(n, m, (char) (rand()%('z' - 'a' + 1) + 'a'));
}
```

1. Frame di codice da inserire nella funzione main:

www.dmi.unict.it/~messina/didat/prog1_18_19/15_02_2019/frameD.cpp

oppure Short URL:

<https://goo.gl/Drk4wv>

2. Output di controllo utenti Windows:

www.dmi.unict.it/~messina/didat/prog1_18_19/15_02_2019/out_win_D.txt

oppure Short URL:

<https://goo.gl/Lozjhd>