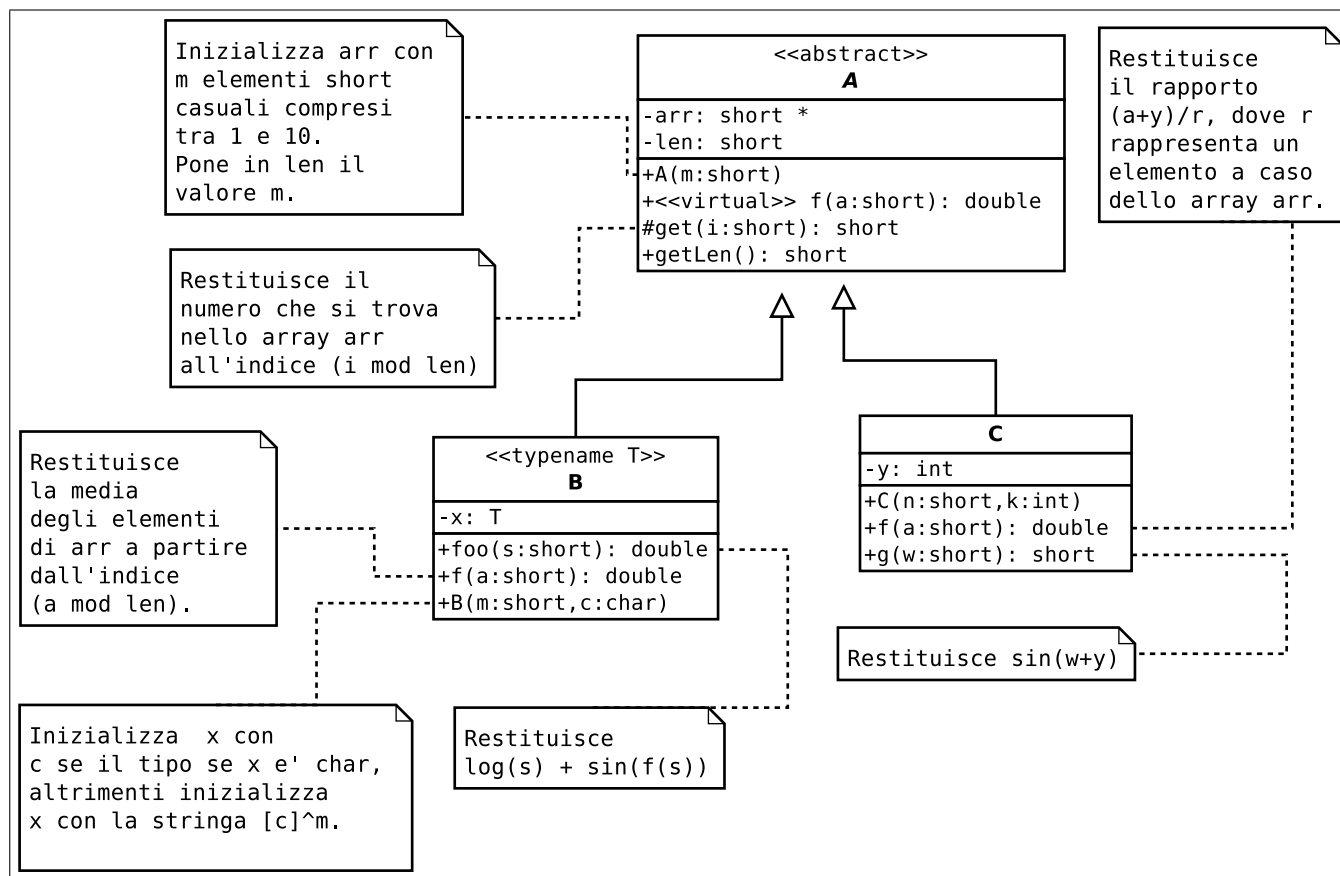


Si implementi in C++ la gerarchia ereditaria descritta dal seguente diagramma UML delle classi. NB: È necessario implementare tutti e soli i metodi indicati nel diagramma.



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:

```

39)B<char> arr=[4 6 7 5 4 2 5 ], x=p, f(3)=3.2
40)B<string> arr=[8 8 10 7 1 8 4 9 ], x=zzzzzzzz, f(3)=4.83333
41)C arr=[1 3 2 6 ], y=6, f(3)=3
    
```

2. si calcoli il massimo valore `f(3)` per tutti gli oggetti della collezione e la media dei valori `g(5)` per tutti gli oggetti di tipo C;
3. si implementi l'overloading dello operatore membro prefisso `++` per la classe C e lo si utilizzi nel main.

Output di controllo: [vedi URL sul retro del foglio].

```

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 C(n, rand()%10 + 1);
            break;
        case 1:
            vett[i]= new B<string>(n, rand()%('z' - 'a' + 1) + 'a');
            break;
        case 2:
            vett[i]= new B<char>(n, rand()%('z' - 'a' + 1) + 'a');
    }
}

```

1. Frame di codice da inserire nella funzione main:

www.dmi.unict.it/~messina/didat/prog1_18_19/28_02_2019/frameA.cpp

oppure Short URL:

<https://goo.gl/nERVsj>

2. Output di controllo utenti Windows:

www.dmi.unict.it/~messina/didat/prog1_18_19/28_02_2019/out_windows_A.txt

oppure Short URL:

<https://goo.gl/KeimjP>

3. Output di controllo utenti Linux:

http://www.dmi.unict.it/~messina/didat/prog1_18_19/28_02_2019/out_linux_A.txt

oppure Short URL:

<https://goo.gl/Cmfgr6>