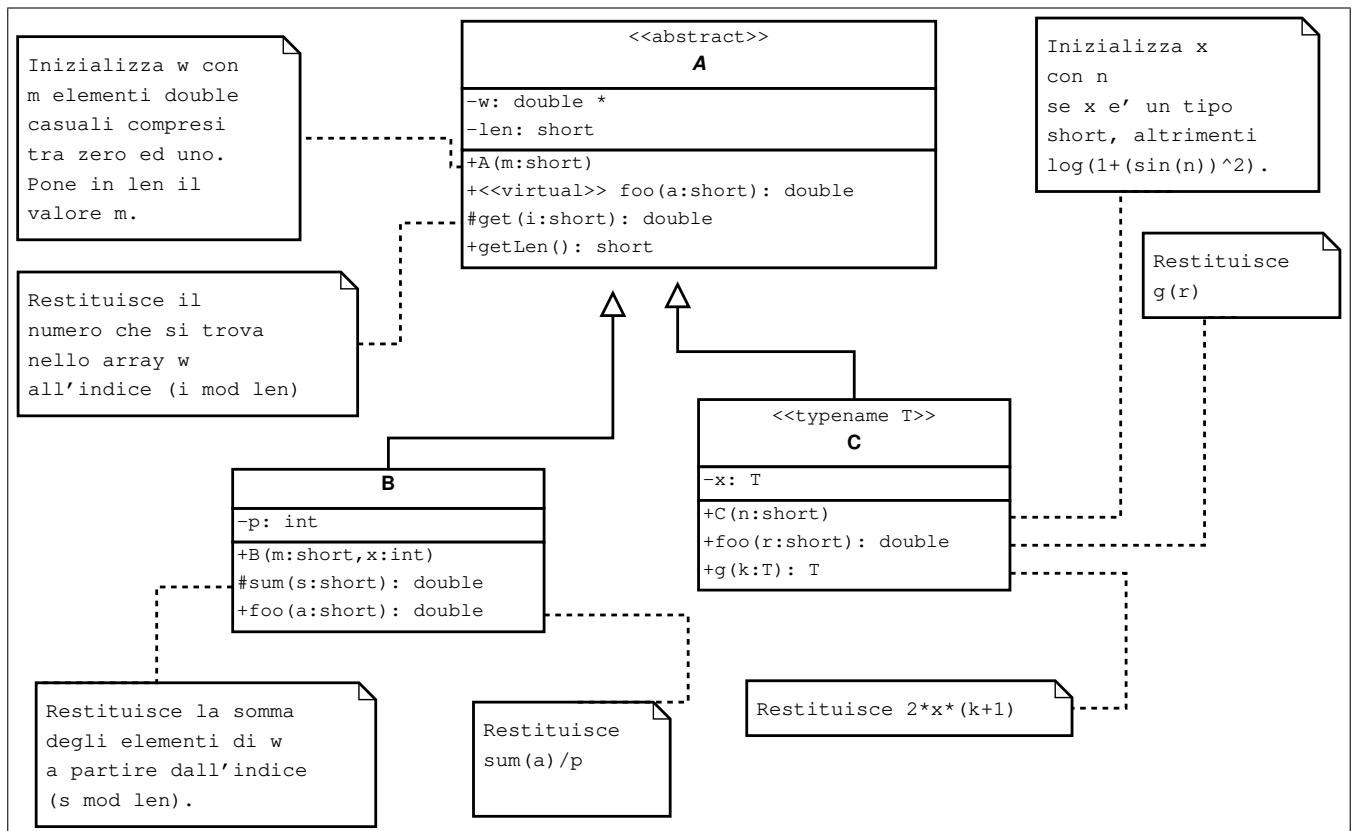


Si implementi in C++ la gerarchia ereditaria descritta dal seguente diagramma UML delle classi. NB: È necessario implementare tutti e soli i metodi indicati nel diagramma.



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:

```

30)1CI5E, w=[0.766171 0.725601 0.900516 0.814684 0.889074 0.799782 0.387099 0.141197 0.358242 ], x=9, foo(3)=72
31)1CI5E, w=[0.669342 0.803909 0.0982594 0.601108 0.327015 0.790548 0.2399 ], x=0.358815, foo(3)=2.87052
32)1B, w=[0.302023 0.22095 0.679391 0.547904 0.924594 0.295153 0.63216 ], p=7, foo(3)=0.34283
    
```

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()%10 + 1);
            break;
        case 1:
            vett[i]= new C<double>(n);
            break;
        case 2:
            vett[i]= new C<short>(n);
    }
}
```

1. Frame di codice da inserire nella funzione main:

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

oppure Short URL:

<https://goo.gl/CTL93b>