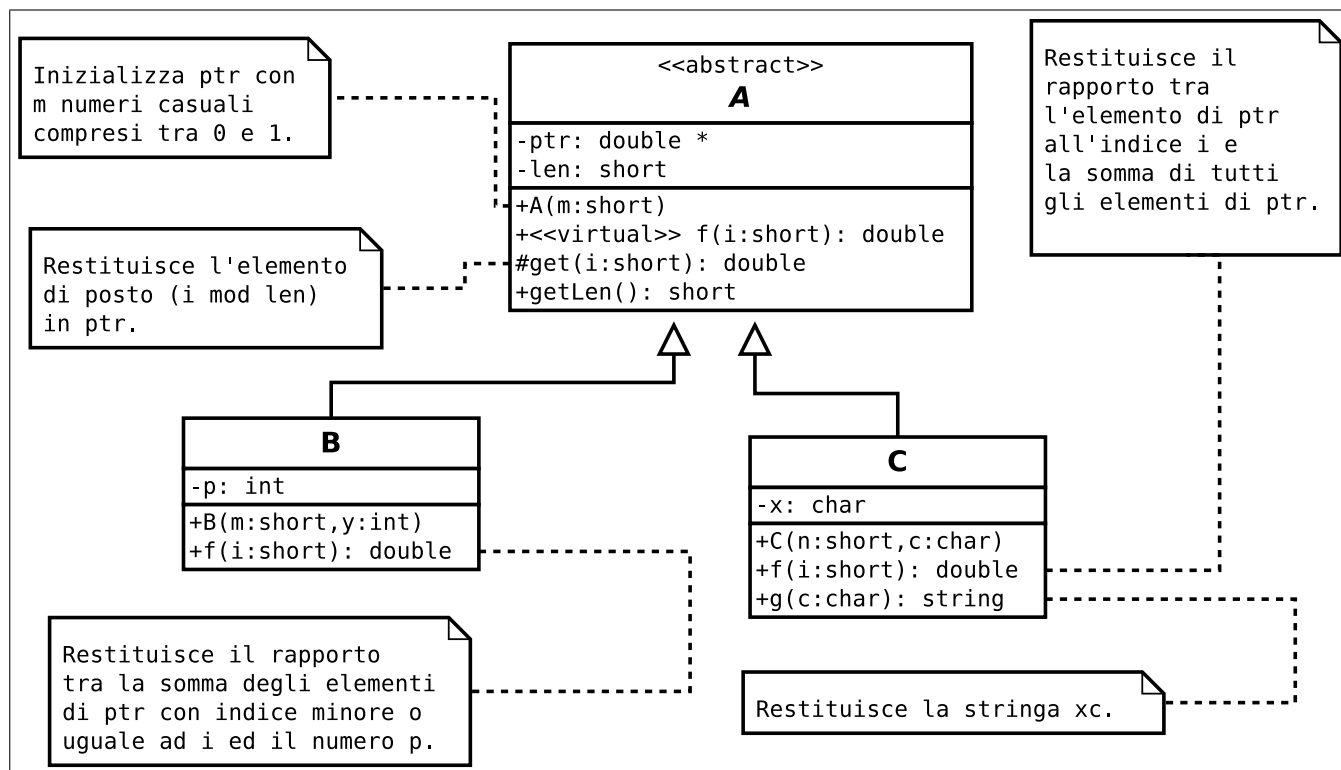


Si implementi in C++ la gerarchia ereditaria descritta dal seguente diagramma UML delle classi. NB: È necessario implementare tutti e soli i metodi indicati nel diagramma.



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:

```

36)1B, ptr=[ 0.594488 0.933246 0.111384 0.294766 0.904513 0.839223 ], p=3, f(3)=0.644628
37)1B, ptr=[ 0.541552 0.456116 ], p=4, f(3)=0.498834
38)1C, ptr=[ 0.173756 0.408945 0.0440274 0.585541 0.0726513 0.677398 ], x=f, f(3)=0.298392
  
```

2. si calcoli la media dei valori f(3) 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.

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

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

1. Frammento di codice da inserire nella funzione main:

www.dmi.unict.it/~messina/didat/prog1_18_19/29_04_2019/frame-29_04.cpp

oppure Short URL:

<https://tinyurl.com/y3vrbvaw>