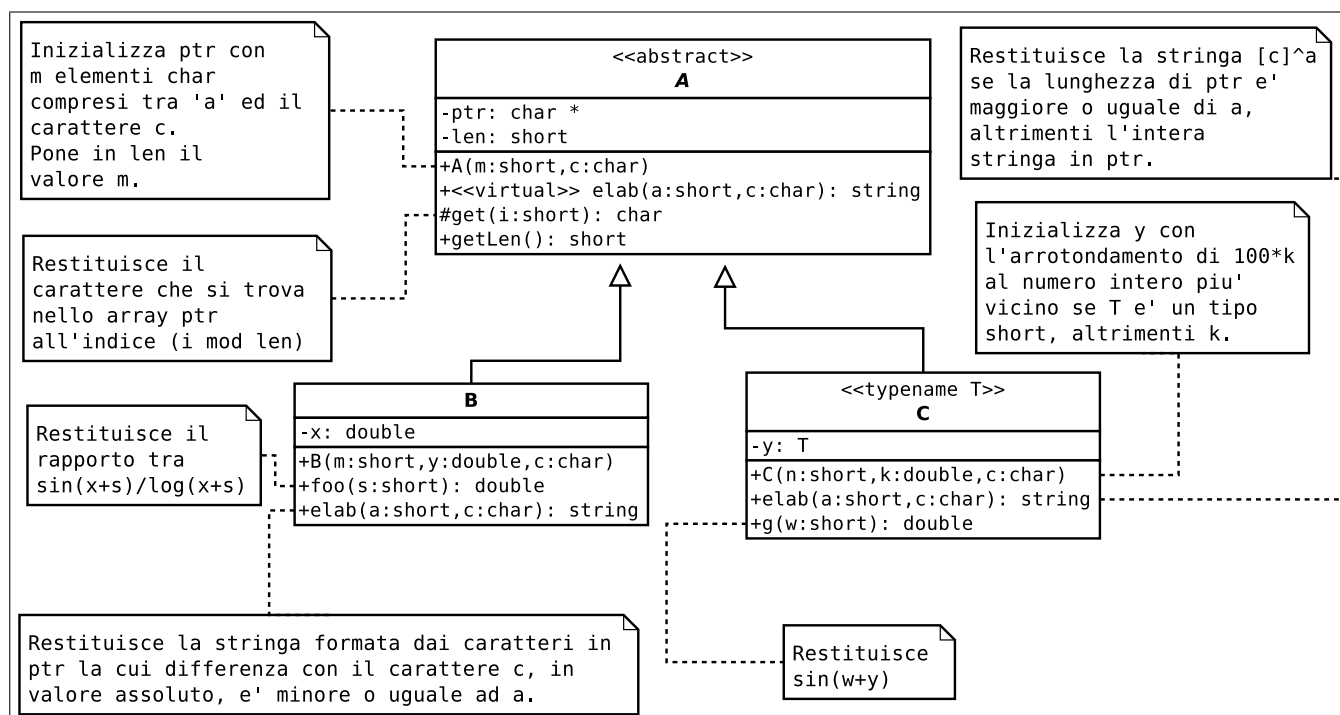


Si implementi in C++ la gerarchia ereditaria descritta dal seguente diagramma UML delle classi. NB: È necessario implementare tutti e soli i metodi indicati nel diagramma.



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:

```

23) B ptr=[f f x b k k ], x=0.00745354, elab(5, z)=x
24) C<double> ptr=[v o e v u ], y=0.0485132, elab(5, z)=zzzzz
25) C<short> ptr=[g q ], y=58, elab(5, z)=gq
    
```

2. si calcoli la media dei valori foo(5) per tutti gli oggetti di tipo B, e la media dei valori g(5) per tutti gli oggetti di tipo C<short>;
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, (double) rand()/RANDMAX, rand()%('z' - 'a' + 1) + 'a');  
             break;  
         case 1:  
             vett[i]= new C<double>(n, (double) rand()/RANDMAX, rand()%('z' - 'a' + 1) + 'a');  
             break;  
         case 2:  
             vett[i]= new C<short>(n, (double) rand()/RANDMAX, 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/frameB.cpp](http://www.dmi.unict.it/~messina/didat/prog1_18_19/28_02_2019/frameB.cpp)

oppure Short URL:

<https://goo.gl/GR1ytV>