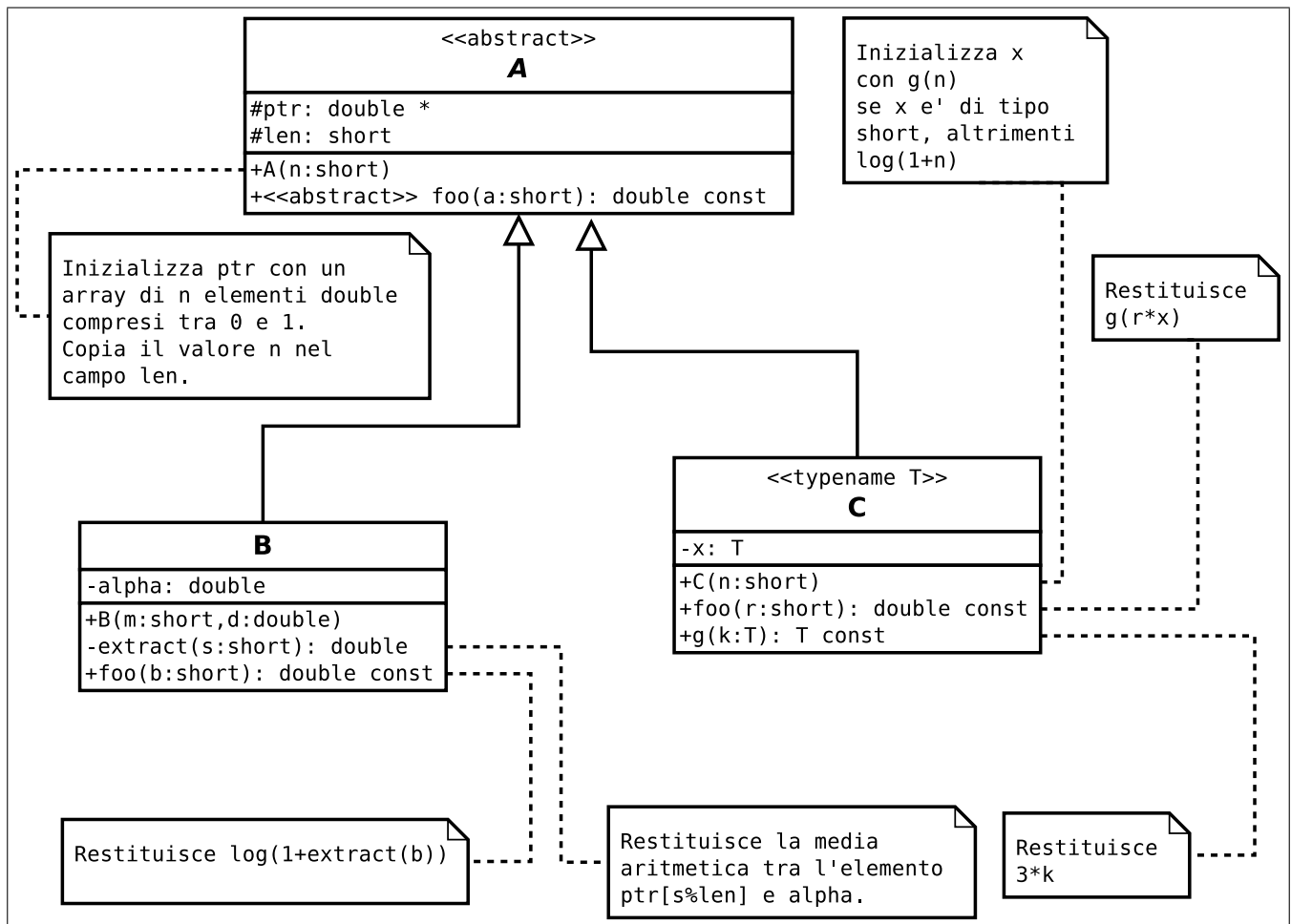


Si implementi in C++ la gerarchia ereditaria descritta dal seguente diagramma UML delle classi. NB: È necessario implementare tutti e soli i metodi indicati nel diagramma.



In un main, si generi una collezione di 30 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 dai link indicati sul retro del foglio

Successivamente:

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

```

4)B, ptr=[0.32407 0.256984 0.747291 0.487611 0.615914 ], alpha=0.05, foo(5)=0.335504
5)C<int>, ptr=[0.791312 0.78885 0.0491608 ], x=9, foo(5)=135, g(5)=15
6)B, ptr=[0.21169 0.659288 0.0233332 ], alpha=0.03, foo(5)=0.114083
    
```

2. si calcoli il massimo valore foo(5) per tutti gli oggetti della collezione e la somma g(5) per tutti gli oggetti di classe C<double>;
3. si implementi l'overloading dello operatore membro "[]" per la classe A e lo si utilizzi nel main.

Output di controllo:

- utente Windows: [vedi link sul retro del foglio]
- utente Linux: [vedi link sul retro del foglio]

```
srand(328832748);  
for(int i=0; i<DIM; i++) {  
    short n=1+rand()%5;  
    switch(rand()%3) {  
        case 0: vett[i]= new B(n, n/100.0); break;  
        case 1: vett[i]= new C<double>(n); break;  
        case 2: vett[i]= new C<int>(n);  
    }  
}
```

1. Frame di codice per la funzione main:

<http://www.dmi.unict.it/~messina/didat/14122018/frame.cpp>

oppure Short URL:

<https://goo.gl/igKXJT>

2. Output di controllo utenti Windows:

<http://www.dmi.unict.it/~messina/didat/14122018/windows.txt>

oppure Short URL:

<https://goo.gl/KmeuYw>

3. Output di controllo utenti Linux:

<http://www.dmi.unict.it/~messina/didat/14122018/linux.txt>

oppure Short link:

<https://goo.gl/ZDZQpz>