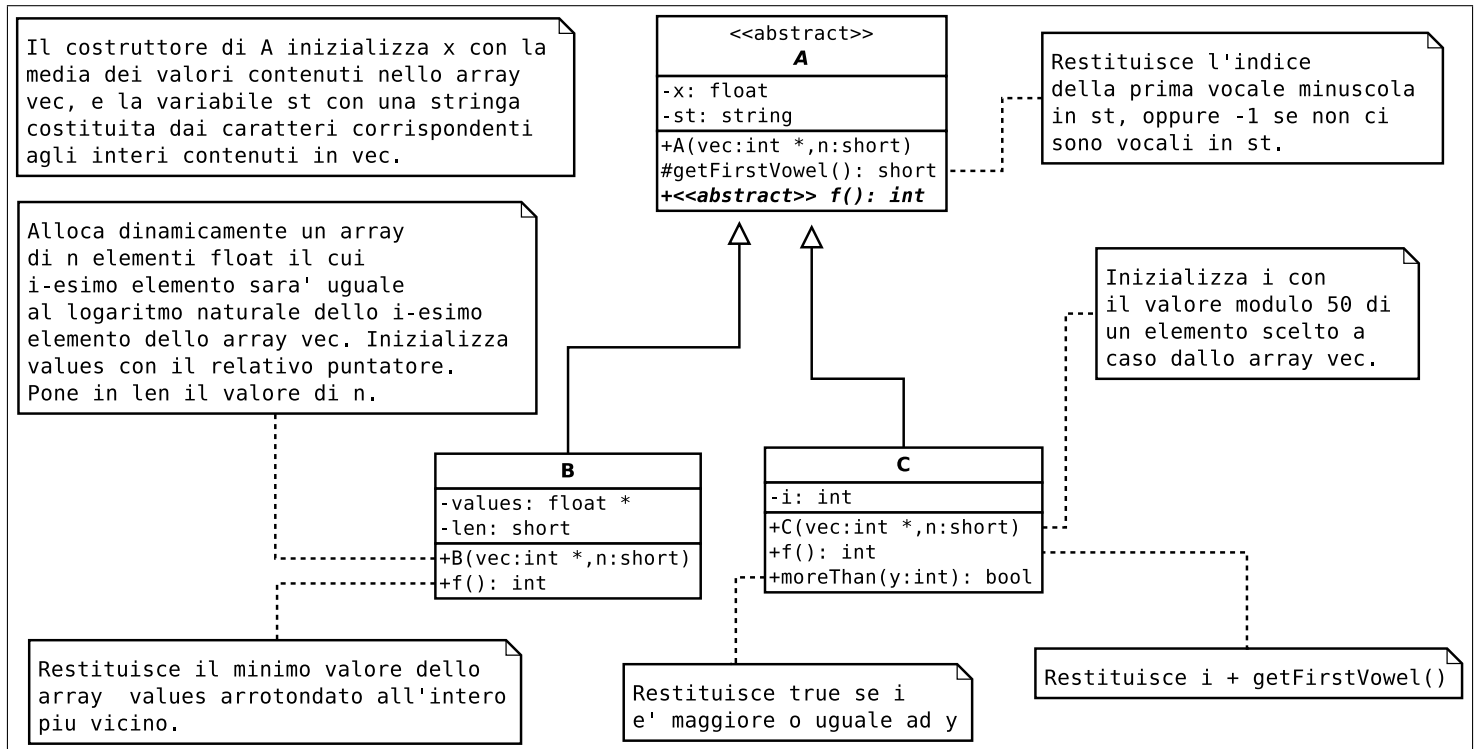


Università di Catania
 Dipartimento di Matematica e Informatica
 Corso di Studio in Informatica, A.A. 2017-2018
 Prova di laboratorio di Programmazione I (9 CFU).
 COMPITO B
 16 Marzo 2018

VERSIONE PDF: <http://www.dmi.unict.it/~messina/didat/16032018/compB.pdf>

Si implementi in C++ la gerarchia ereditaria descritta dal seguente diagramma UML delle classi. N.B.: 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 a pagina 2.

(NB: È possibile scaricare il frame di codice da inserire nel main al seguente URL:

<http://www.dmi.unict.it/~messina/didat/16032018/frameB.cpp>).

Successivamente, relativamente a questa collezione:

1. si visualizzi la collezione mediante l'overloading dell'operatore <<, esempio (non per controllo):

```

Class 1C: x=106.6, st=anyjdapgd { i=21
  f()=21
Class 1C: x=108.625, st=gytiajdy { i=16
  f()=19
Class 1B: x=106.8, st=scvai { values=[4.74493 4.59512 4.77068 4.57471 4.65396 ], len = 5
  f()=5
  
```

2. si calcoli il massimo, il minimo e la media aritmetica dei valori restituiti dal metodo f().
3. si calcolino tutti gli indici degli oggetti per cui il moreThan(10) restituisce true.

Output di controllo:

- Windows: http://www.dmi.unict.it/~messina/didat/16032018/outB_windows.txt
- Linux: http://www.dmi.unict.it/~messina/didat/16032018/outB_linux.txt

```
srand(111222333);
A *vett [DIM];
short l, select;
int *vec;

for(int i=0; i<DIM; i++){
    l = rand()%8 + 5;
    select = rand()%2;
    vec = new int [1];

    for(int c=0; c<1; c++)
        vec[c] = 97 + rand()%25;

    if(select)
        vett[i] = new B(vec, l);
    else
        vett[i] = new C(vec, l);
}
```