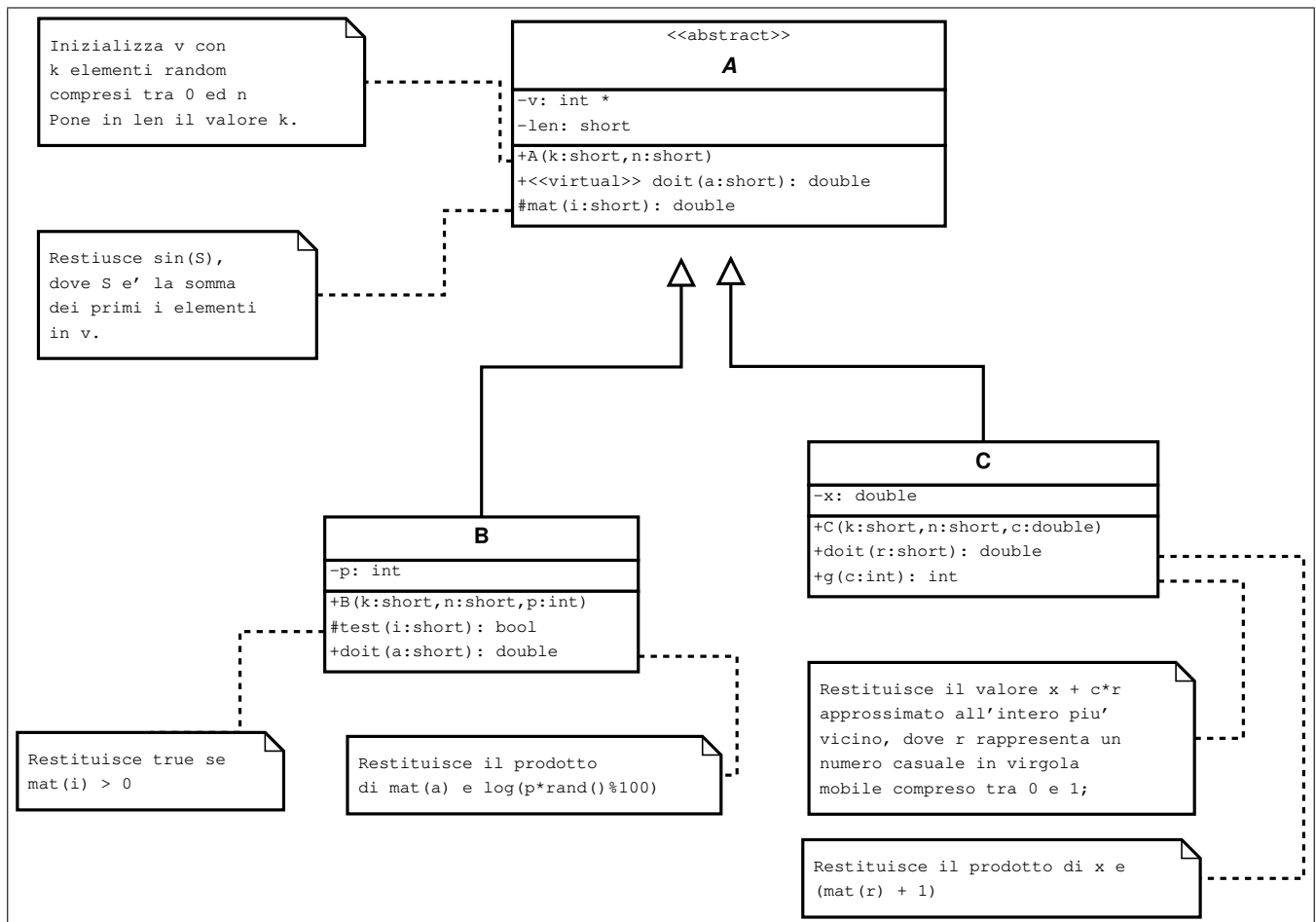


Università di Catania
Dipartimento di Matematica e Informatica
Corso di Studio in Informatica, A.A. 2019-2020
Prova di laboratorio di Programmazione I (9 CFU).
10 Febbraio 2020 - C

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:

```

44) 1B v = [ 1 5 5 7 ], p=3, doit(3)=-3.87116
45) 1C v = [ 4 2 4 6 0 2 4 9 ], x=10, doit(3)=-4.44021
46) 1C v = [ 4 5 6 8 ], x=3, doit(3)=2.95086
    
```

2. si calcoli il massimo valore `doit(3)` per tutti gli oggetti della collezione e la media dei valori `g(5)` per tutti gli oggetti di tipo C;
3. si implementi l'overloading dello operatore membro prefisso `++` per la classe B e lo si utilizzi nel main per un qualche oggetto di tipo B.

```
srand(111222333);

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

1. Frame di codice da inserire nella funzione main:

www.dmi.unict.it/~gfarinella/Prog1/10022020/frame-10_02_2020_C.cpp

oppure Short URL:

<http://tiny.cc/mreojz>

2. Output di controllo utenti Windows:

www.dmi.unict.it/~gfarinella/Prog1/10022020/out_windows_C.txt

oppure Short URL:

<http://tiny.cc/vseojz>

3. Output di controllo utenti Linux:

www.dmi.unict.it/~gfarinella/Prog1/10022020/out_linux_C.txt

oppure Short URL:

<http://tiny.cc/nteoiz>