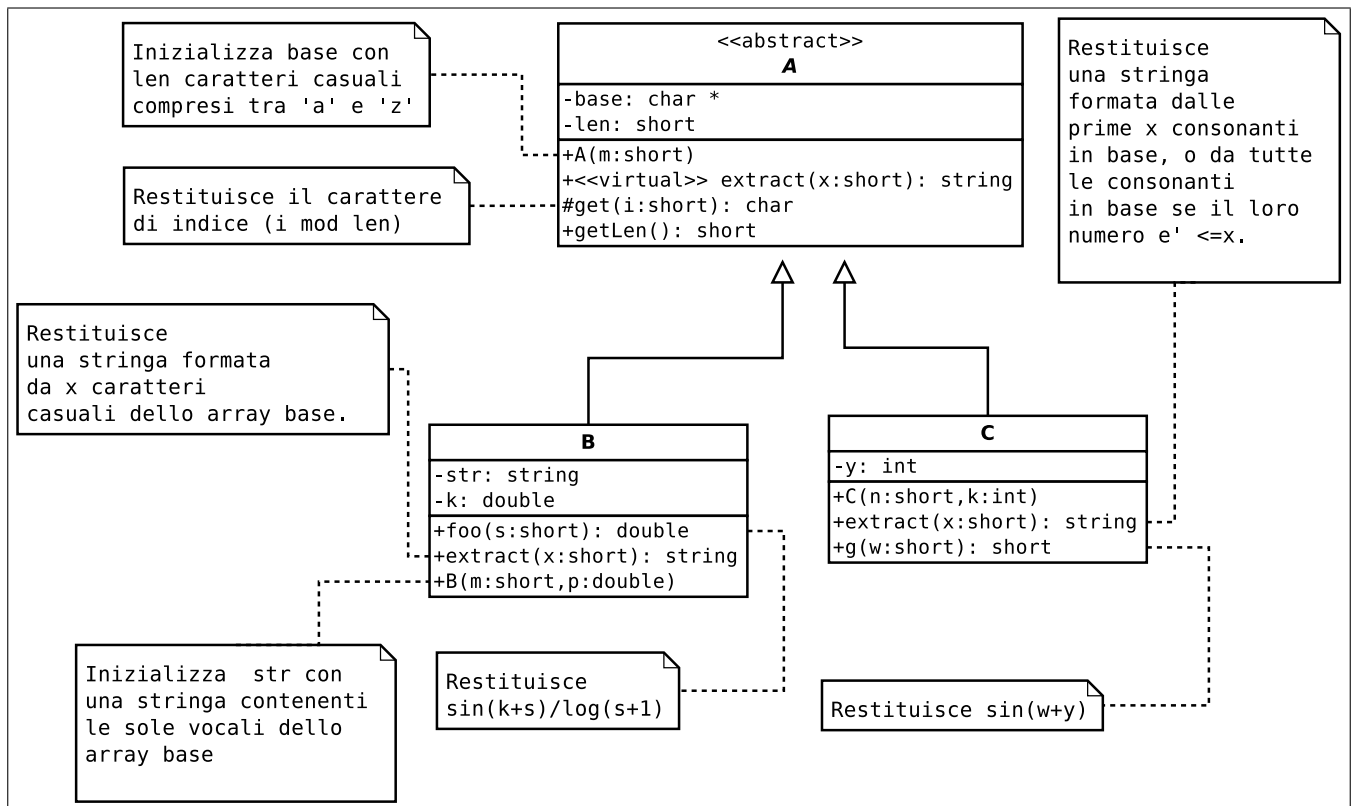


Si implementi in C++ la gerarchia ereditaria descritta dal seguente diagramma UML delle classi. NB: È necessario implementare tutti e soli i metodi indicati nel diagramma.



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:

```

41)C: base=[y t c a b z e b m d n t z d ], y=2, extract(3)=ytc
42)B: base=[k v s e p n v o m u l g ], str=eou, k=0.774659, extract(3)=vue
43)C: base=[e u h h u n j i s y b s d q h l ], y=7, extract(3)=hhn
    
```

2. si calcoli una singola media di tutti i valori `foo(5)` e `g(5)` per tutti i gli oggetti di tipo B e C.
3. si implementi l'overloading dello operatore membro postfixo `++` per la classe C e lo si utilizzi nel main.

Output di controllo: [vedi URL sul retro del foglio].

```

A *vett [DIM];

for (int i=0; i<DIM; i++){
    short n=10+rand()%10;
    switch (rand()%2){
        case 0:
            vett[i]= new C(n, rand()%20 + 1);
            break;
        case 1:
            vett[i]= new B(n, rand()/(double) RANDMAX);
            break;
    }
}

```

1. Frame di codice da inserire nella funzione main:

http://www.dmi.unict.it/~messina/didat/prog1_18_19/28_02_2019/frameC.cpp

oppure Short URL:

<https://goo.gl/FnhfZv>

2. Output di controllo utenti Windows:

http://www.dmi.unict.it/~messina/didat/prog1_18_19/28_02_2019/out_windows_C.txt

oppure Short URL:

<https://goo.gl/7NSNKv>

3. Output di controllo utenti Linux:

http://www.dmi.unict.it/~messina/didat/prog1_18_19/28_02_2019/out_linux_C.txt

oppure Short URL:

<https://goo.gl/t6ZMEa>