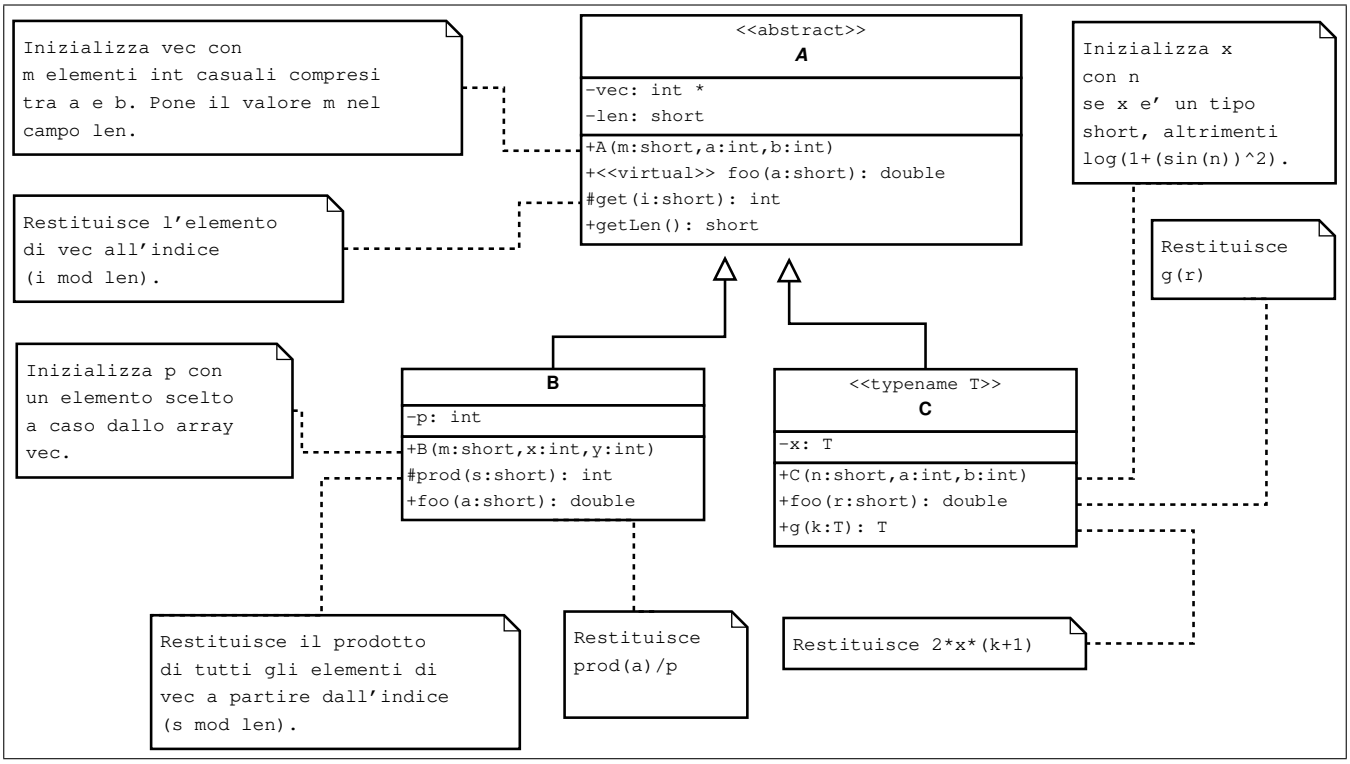
Università di Catania
Dipartimento di Matematica e Informatica
Corso di Studio in Informatica, A.A. 2018-2019
Prova di laboratorio di Programmazione I (9 CFU).
30 Aprile 2019 - B

Si implementi in C++ la gerarchia ereditaria descritta dal seguente diagramma UML delle classi. NB: É 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:

```
31)1CIdE, vec=[4 5 9 5 9 9 9 5 7 6 ], x=0.259251, foo(3)=2.07401
32)1B, vec=[8 14 6 9 9 9 5 8 12 11 ], p=9, foo(3)=0
33)1CIsE, vec=[7 5 ], x=2, foo(3)=16
```

2. si calcoli il massimo valore foo(3) per tutti gli oggetti della collezione e la media dei valori g(5) per per tutti gli oggetti di tipo C<double>;

3. si implementi l'overloading dello operatore membro "()" per la classe A e lo si utilizzi nel main. La funzione di overloading dovrà prevedere due argomenti interi $i1$ e $i2$ e restituire la somma di tutti gli elementi presenti in vec, dall'indice $i1$ all'indice $i2$.

```
    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, rand()%5 + 1, rand()%10 + 5);
          break;
        case 1:
          vett[i]= new C<double>(n, rand()%5 + 1, rand()%10 + 5);
          break;
        case 2:
          vett[i]= new C<short>(n, rand()%5 + 1, rand()%10 + 5);
      }
  }
```

1. Frame di codice da inserire nella funzione main:
   `www.dmi.unict.it/~messina/didat/prog1_18_19/30_04_2019/B/frame-30_04_B.cpp`
   oppure Short URL:
   `https://tinyurl.com/y2le3laa`