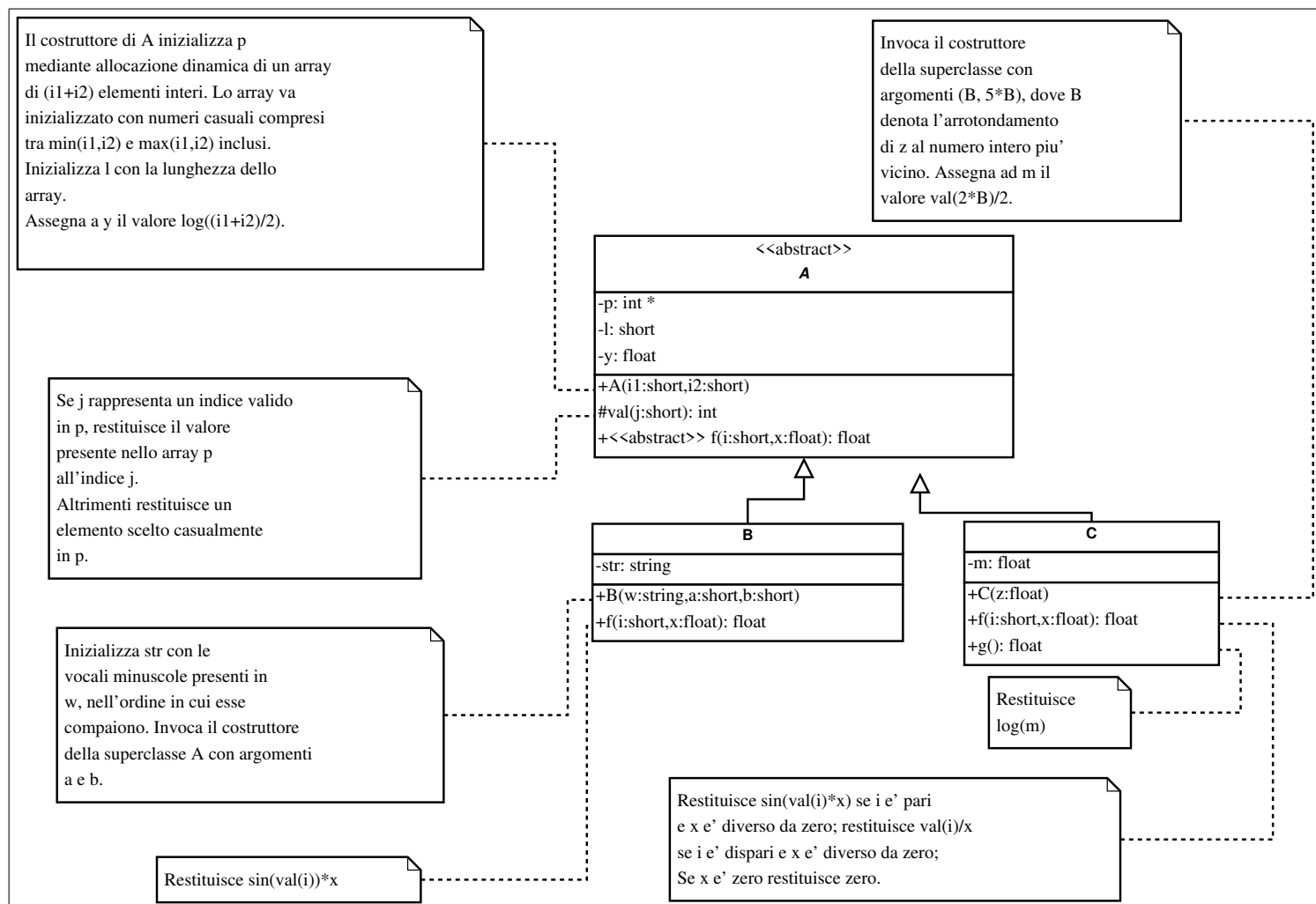


Università di Catania
Dipartimento di Matematica e Informatica
Corso di Studio in Informatica, A.A. 2017-2018
Prova di laboratorio di Programmazione I (9 CFU).
29 Giugno 2018

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. Il codice non indentato sarà considerato errato!!!



In un main, si generi una collezione di 30 oggetti utilizzando la sequenza di valori casuali riportata a pagina 2.

NB: Scaricare il frammento di codice da inserire nel main (e riportato a pagina 2) mediante il link:

<https://goo.gl/q3sRk9>.

Successivamente, relativamente a questa collezione:

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

```
22)Class 1B, p=[10 14 10 16 13 11 10 10 10 15 13 12 16 15 9 16 9 10 16 13 11 8 9 8 ]: l=24, y=2.48491{ str=oiuo}
23)Class 1B, p=[12 12 11 12 12 11 12 11 12 12 12 12 11 11 11 12 12 11 12 12 11 12 12 ]: l=23, y=2.44235{ str=e}
24)Class 1C, p=[4 4 4 1 2 2 ]: l=6, y=1.09861{ m = 2}
```

2. si calcolino tutti gli indici degli oggetti della collezione per cui $f(5, 0.5) < 0.5$.
3. si calcoli la media dei valori restituiti dal metodo `g()` per tutti gli oggetti di classe C della collezione;

Output di verifica:

- Windows: <https://goo.gl/Wkdzm5>
- Linux: <https://goo.gl/X4BpFZ>

```

srand(111222333);
A *vett [DIM];

string w;
short s1, s2;

for(int i=0; i<DIM; i++){
    if(rand()%2==0){ //B
        w = "";
        s1 = rand()%20 + 5;
        s2 = rand()%20 + 5;
        short len = rand()%50 + 10;
        for(int j=0; j<len; j++)
            w+=((rand()%2) ? (char) (rand()%26 + 'a') : (char) (rand()%10 + '0'));
        vett[i] = new B(w, s1, s2);
    }
    else{ // C
        int max = 10;
        float z = ((float) rand())/INT_MAX * (rand()%max) + 1.0;
        vett[i] = new C(z);
    }
}
}

```