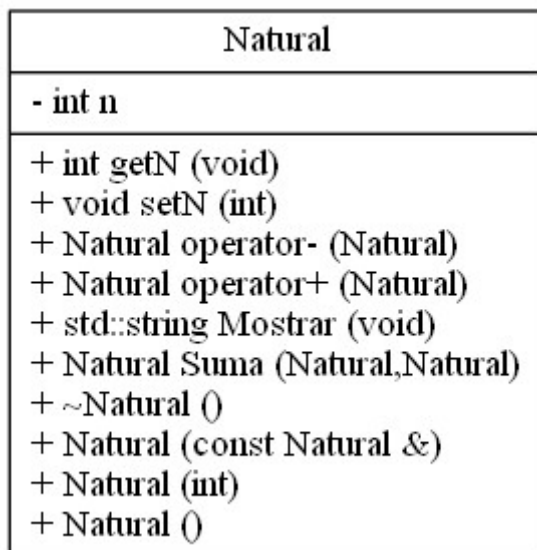


## Ejercitación C++: objetos

- 1) (Ejemplo) Desarrollar un programa en lenguaje C++ con una clase para operar con números naturales respetando sus propiedades:
  - La clase permitirá crear objetos que almacenen sólo números naturales (positivos y cero)
  - Se deberán poder realizar las operaciones suma, resta, multiplicación y división cuyo resultado será siempre un numero natural

Diagrama de la clase:



Se pide:

- a. Definición de la clase Natural (interface):
    - Atributo privado entero
    - Métodos detallados a continuación
  - b. Implementación de los métodos siguientes:
    - i. Constructor ordinario
    - ii. Constructor por defecto
    - iii. Constructor de copia
    - iv. Sobrecarga de los operadores +, -, \* y /
    - v. Método set que sólo asigne un valor al atributo privado si es un numero natural, cero en caso contrario
    - vi. Método get del atributo privado
  - b. Implementar un programa main para demostrar el funcionamiento correcto de la clase
- 
- 2) Desarrollar un programa en lenguaje C++ con una clase Rectángulo con las siguientes características:
    - a. Atributos privados enteros base y altura
    - b. Métodos:
      - i. Constructor ordinario y por defecto

- ii. Métodos set y get de sus atributos (recordar que sus atributos no podrán ser negativos)
- iii. Método imprimir que muestre en pantalla un rectángulo formado por un carácter recibido como argumento
- iv. Métodos para obtener el área y el perímetro del rectángulo
- v. Método para determinar si el rectángulo es un cuadrado

Diagrama de la clase:



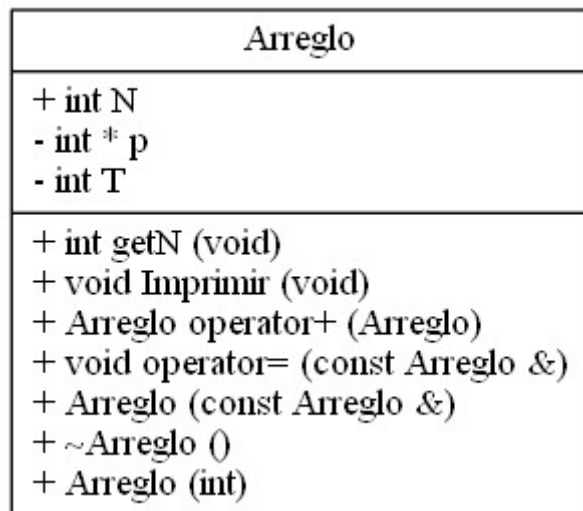
- 3) Desarrollar un programa en lenguaje C++ que implemente 3 funciones para elevar al cuadrado un numero entero (se puede aplicar sobrecarga de funciones):
- a. Pasando el parámetro por valor y devolviendo el resultado
  - b. Pasando el parámetro por referencia mediante puntero, guardando el resultado en la posición de memoria recibida
  - c. Ídem anterior mediante referencia o alias

Desde el programa main, verificar el funcionamiento de las 3 funciones mostrando en pantalla desde main el valor original y su cuadrado para cada una.

- 4) Desarrollar un programa en lenguaje C++ con una clase Arreglo que permita almacenar un arreglo de números enteros con los siguientes elementos:
- a. Atributos: tamaño del arreglo, puntero de inicio y cantidad de arreglos instanciados (static)
  - b. Métodos:
    - i. Constructor ordinario y por defecto que recibe el número de elementos del arreglo, si no recibe nada utiliza 10. Reserva memoria dinámica mediante operador *new*
    - ii. Destructor (uso de *delete*)
    - iii. Sobrecarga operador copia, si el arreglo de destino es de diferente tamaño lo va a redimensionar
    - iv. Sobrecarga operador suma que sume elemento a elemento dos arreglos
    - v. Método imprimir arreglo con sobrecarga operador

- vi. Sobrecarga de los operadores de inserción y extracción de flujo para mostrar en pantalla el arreglo y cargarlo respectivamente

Diagrama de la clase:



5) Dada la clase Complejo:

```
class Complejo {
    double real, imag;
public:
    Complejo(double r=0, double i=0){real = r; imag = i;}
    Complejo(const Complejo &c){real = c.real; imag= c.imag;}
    void Set(double r=0, double i=0){real = r; imag = i;}
    double getR(void){return real;}
    double getI(void){return imag;}}
```

- a. Implementar una funcion Suma que permita sumar dos números complejos como:
    - i. Funcion global
    - ii. Funcion amiga a la clase
    - iii. Método de la clase
  - b. Sobrecarga de los operadores +, -, \*, /
  - c. Sobrecarga del operador de asignación (=)
  - d. Sobrecarga de los operadores de inserción y extracción de flujo (<< y >>)
  - e. Sobrecarga de + para sumar un escalar a un complejo
  - f. Método de la clase `double abs(void)`; que devuelve el módulo de un numero complejo ( $abs(z) = \sqrt{real^2 + imag^2}$ )
- 2) Declarar y cargar un vector de N números complejos mediante:
- a. Vector de tamaño fijo
  - b. Reserva de memoria dinámica mediante operadores new y delete
  - c. Obtener la sumatoria de los valores absolutos de los elementos de cada vector:  $\sum_{i=0}^{N-1} abs(vector(i))$
- 6) Dado el programa principal del archivo **Clase\_contador.cpp**, implemente la clase contador para que el programa funcione correctamente. El mismo genera una cierta

cantidad de números aleatorios y mediante dos objetos de la clase Contador registra una cuenta de la cantidad de números pares e impares generados.