



## Lista 05 – C++

**Para esta lista, entregar todos os exercícios em um arquivo zip. Indicar o ambiente de desenvolvimento.**

**1.** Crie uma classe para representar um polígono. Crie, também, uma classe para representar um quadrado e um triângulo, as quais herdam a classe do polígono. Utilize o conceito de polimorfismo em C++ para que seja possível calcular a área em qualquer objeto polígono.

**(ENTREGAR) 2.** Crie uma classe para gerenciar números complexos e que permita as seguintes operações: soma, subtração, multiplicação, e módulo.

Veja a definição das operações em <http://www.brasilecola.com/matematica/adicao-subtracao-multiplicacao-numero-complexo.htm>.

**(ENTREGAR) 3.** Altere o exercício 2 adicionando sobrecarga de operadores para as operações.

**(ENTREGAR) 4.** Crie uma classe para representar um rádio, uma para representar um relógio e uma para representar um rádio-relógio (utilize herança múltipla). O relógio deve possuir métodos para exibir as horas, definir as horas, e definir um alarme. O rádio deve possuir métodos para setar a estação, mudar entre AM/FM, e ligar e desligar. O rádio-relógio deve ter todas as funcionalidades de rádio e de relógio, além de permitir a definição de um alarme cujo despertador sintoniza o rádio em uma dada estação.

**(ENTREGAR) 5.** Em C++, há vários TADs já implementados, assim como no Java. Eles fazem parte da biblioteca STL (Standard Template Library), e são incluídos de acordo com a necessidade de utilização.

Pesquise o funcionamento da classe `std::vector` em <http://www.inf.pucrs.br/~pinho/PRGSWB/STL/stl.html> e faça um programa que:

1. Receba um valor inteiro  $n$ ;
2. Crie um vector (similar ao `ArrayList`) para armazenar  $n$  objetos do `Numero Complexo` do exercício 1;
3. Gere números complexos aleatórios;
4. Mostre a soma destes números complexos no final, usando o operador `+` para a soma e iterators para o `for`;
5. Mostre o resultado final na tela;
6. Livre a memória deletando todos objetos utilizados no programa, antes de encerrá-lo.