



Laboratório 2

Classes, Objetos, Métodos Construtores, Método Destrutor e Sobrecarga de Operadores

Objetivo

O objetivo deste exercício é colocar em prática conceitos iniciais do paradigma de Programação Orientada a Objetos (POO) na linguagem de programação C++, em particular a implementação de classes, objetos, métodos construtores, método destrutor e sobrecarga de operadores.

Questão 01

Implemente um programa em C++ que atenda aos seguintes critérios:

- a) Contenha uma classe que representa um funcionário, registrando seu nome (pesquise o uso da classe `string`¹), salário e data de admissão;
- b) Contenha também uma classe que representa uma empresa, registrando seu nome, CNPJ e lista de funcionários;
- c) O programa deverá permitir criar uma empresa;
- d) O programa deverá permitir que se adicione funcionários (um por vez) a uma empresa e não deve permitir adicionar um funcionário que já tenha sido anteriormente adicionado, sendo neste caso exibida uma mensagem de erro. Os funcionários de uma empresa poderão ser mantidos em um `array`, `vector`² ou `list`³ (pesquisar sobre a STL do C++), ou ainda alguma estrutura de dados que você considere conveniente. Dica: utilize a sobrecarga de operadores.
- e) O programa deverá permitir listar os dados de todos os funcionários de uma empresa, sobrecarregando-se o operador de inserção em stream (`<<`);
- f) O programa deverá permitir que seja dado um aumento de X% a todos os funcionários de uma determinada empresa;
- g) O programa deverá permitir listar os dados de todos os funcionários de uma empresa em período de experiência, ou seja, contratados há menos de 90 dias considerando a data corrente;
- h) O programa deverá permitir listar a média de funcionários por empresa. O cálculo deve ser realizado a partir de atributos estáticos das próprias classes (Empresa e Funcionário).

Dica 1: Observe que as funcionalidades exigidas nesta questão podem (e devem) ser resolvidas com a alteração das classes (por exemplo, criando novos métodos) e não manipulando de forma mirabolante os objetos no programa principal.

Dica 2: A modelagem das classes é livre, ou seja, inclua os atributos que você considerar importantes para resolver a questão.

¹<http://www.cplusplus.com/reference/string/string/>

²<http://www.cplusplus.com/reference/vector/vector/>

³<http://www.cplusplus.com/reference/list/list/>

Autoria e política de colaboração

Esta atividade é em dupla. O trabalho em cooperação entre estudantes da turma é estimulado, sendo aceitável a discussão de ideias e estratégias. Contudo, tal interação não deve ser entendida como permissão para utilização de (parte de) código fonte de outros colegas, o que pode caracterizar situação de plágio. Trabalhos copiados em todo ou em parte de outros colegas ou da Internet serão rejeitados.

Entrega

Você deverá submeter um único arquivo compactado no formato .zip contendo, de forma organizada, todos os códigos fonte resultantes da implementação deste laboratório, sem erros de compilação e devidamente testados e documentados, **até as 23h59 do dia 01 de Abril de 2019** através da opção *Tarefas* na Turma Virtual do SIGAA. Seu arquivo compactado deverá incluir também um arquivo texto README contendo: a identificação completa da dupla do laboratório; a descrição de como compilar e rodar o programa, **incluindo um roteiro de entradas e comandos que destaque as funcionalidade (a)-(h)**; a descrição das limitações (caso existam) do programa, e; quaisquer dificuldades encontradas.

Orientações gerais

- 1) Utilize estritamente recursos da linguagem C++.
- 2) Durante a compilação do seu código fonte, você deverá habilitar a exibição de mensagens de aviso (*warnings*), pois eles podem dar indícios de que o programa potencialmente possui problemas em sua implementação que podem se manifestar durante a sua execução.
- 3) Aplique boas práticas de programação. Codifique o programa de maneira legível (com indentação de código fonte, nomes consistentes, etc.). **Modularize** e comente seu código.
- 4) Busque desenvolver o seu programa com qualidade, garantindo que ele funcione de forma correta e eficiente. Pense também nas possíveis entradas que poderão ser utilizadas para testar

apropriadamente o seu programa e trate adequadamente possíveis entradas consideradas inválidas.

Avaliação

O trabalho será avaliado sob os seguintes critérios: (i) utilização correta e abrangente dos conteúdos vistos anteriormente e nas aulas presenciais da disciplina; (ii) a corretude da execução do programa implementado, que deve apresentar saída em conformidade com a especificação e as entradas de dados fornecidas, e; (iii) a aplicação correta de boas práticas de programação, incluindo legibilidade, organização e documentação de código fonte. A presença de mensagens de aviso (*warnings*) ou de erros de compilação e/ou de execução, a modularização inapropriada e a ausência de documentação são faltas que serão penalizadas.