

IMD - UFRN – Linguagem de Programação I



2019.2 – Avaliação Prática 1

Leia as seguintes instruções ANTES DE INICIAR A AVALIAÇÃO

- Utilize estritamente recursos da linguagem C++.
- Durante a compilação do seu código fonte, você deverá habilitar a exibição de mensagens de aviso (*warnings*).
- Aplique boas práticas de programação. Codifique o programa de maneira legível (com indentação de código fonte, nomes consistentes, etc.). **Modularize** e comente seu código.
- Garanta que seu programa 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.
- A resolução da questão deve conter um arquivo makefile responsável pela compilação e ligação.
- Quando não especificado, a modelagem das classes é livre, ou seja, inclua os atributos que você considerar importantes para resolver a questão.

Questão

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

a) Contenha uma classe que representa um automóvel, registrando sua marca, preço, número do chassi e data de fabricação;

b) Contenha também uma classe que representa uma concessionária, registrando seu nome, CNPJ e número de automóveis no estoque;

c) O programa deverá permitir criar uma concessionária;

d) O programa deverá permitir que se adicione carros (um por vez) a uma concessionária e não deve permitir adicionar um carro que já tenha sido anteriormente adicionado, sendo neste caso exibida uma mensagem de erro. Os carros de uma concessionária poderão ser mantidos em um *array*, *vector*¹ ou *list*², ou ainda alguma estrutura de dados que você considere conveniente.

e) O programa deverá permitir listar os dados de todos os carros de uma concessionária, sobrecarregando-se o operador de inserção em stream (<<);

f) O programa deverá permitir que seja dado um aumento de X% ao preço de todos os carros de uma determinada concessionária;

g) O programa deverá permitir listar os dados de todos os carros de uma concessionária que foram produzidos há menos de 90 dias considerando a data corrente;

h) O programa deverá permitir listar a média de carros por concessionária. O cálculo deve ser realizado a partir de atributos estáticos das próprias classes (Concessionaria e Automovel). (Dica: para isso, seu programa deverá criar, como exemplo, algumas concessionárias contendo alguns automóveis).

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

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

Autoria e política de colaboração

Esta atividade é em dupla. A colaboração é permitida apenas entre os membros da dupla. Não são permitidos a comunicação ou o compartilhamento de códigos entre as duplas.

Entrega

Cada membro da dupla 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 09 de Setembro 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 destaquem as funcionalidades (a)-(h)**; a descrição das limitações (caso existam) do programa e quaisquer dificuldades encontradas.

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.

Critério	Nota
(A) Readme	[0, 10]
(B) Legibilidade, organização e documentação de código fonte	[0, 10]
(C) Modularização	[0, 10]
(D) Classes: atributos e métodos; construtores e destrutores; níveis de acesso e métodos de acesso	[0, 10]
(E) Sobrecarga de funções e operadores	[0, 10]
(F) Uso consistente de alocação de memória	[0, 10]
(G) Uso adequado de makefile	[0, 10]
(H) Corretude, saída em conformidade com a especificação	[0, 10]
(I) Compilação e warnings	[0, 1]
Nota final: $[(A+B+C+D+E+F+G+2*H)*I] / 9$	