

IMD - UFRN – Linguagem de Programação I

2019.2 – Avaliação Prática 2

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 e métodos que você considerar importantes para resolver a questão.

Questão

Esta atividade irá **expandir** o sistema desenvolvido na AP1 (seu programa deve manter as funcionalidades da AP1). Modifique o código desenvolvido na AP1 de acordo com os seguintes critérios:

- a) Uma Concessionária pode comercializar diversos tipos de veículos: automóveis, motos e caminhões. Todo Veículo possui os seguintes campos: marca, preço, número do chassi e data de fabricação. Apenas Automóveis possuem um atributo relativo ao tipo de motor (gasolina ou elétrico); apenas Motos possuem um atributo relativo ao modelo (clássico ou esportivo); apenas Caminhões possuem um atributo relativo ao tipo de carga (comum ou perigosa).
- b) O programa deverá permitir escrever em um arquivo de texto todo o estoque de uma concessionária, incluindo os atributos comuns e particulares de cada veículo.
- c) O programa deverá permitir criar uma concessionária à partir de um arquivo de texto no formato descrito no item b.
- d) A concessionária deve possuir um atributo relativo à sua Propriedade, que pode ser de uma pessoa física ou de uma pessoa jurídica. Uma Propriedade é caracterizada por dois nomes. Pessoas físicas possuem como primeiro e segundo nomes uma string, enquanto que pessoas jurídicas utilizam números para essas informações.
- e) A concessionária deve possuir um atributo para o registro de todos os chassis dos seus veículos. O programa deve permitir que se realize uma busca por número de chassi. Caso o veículo pertença à concessionária, o programa deve exibir na tela seus atributos.
- f) O programa deve possuir uma função para listar os proprietários de concessionárias, a frota total de cada tipo de veículo e o valor total dos veículos.
- g) Inclua uma funcionalidade inédita que utilize um ou mais conteúdos da Unidade 2.

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 28 de Outubro 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)-(g)**; 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. **Duplas não serão avaliadas individualmente - comunique imediatamente ao professor quaisquer ocorrências.**

| Critério | Nota |
|--|---------|
| (A) Readme, Legibilidade, organização e documentação de código fonte | [0, 10] |
| (B) Modularização | [0, 10] |
| (C) Classes: atributos e métodos; herança e classes abstratas | [0, 10] |
| (D) Manipulação de arquivos | [0, 10] |
| (E) Sobrecarga de funções e operadores; uso de templates | [0, 10] |
| (F) Uso consistente de alocação de memória e de TADs | [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: $[(A+B+C+D+E+F+G+2*H)*I] / 9$ | |