



Laboratório 10

Tratamento de Exceções

Objetivo

O objetivo deste exercício é colocar em prática conceitos de tratamento de exceções em linguagem de programação C++.

Orientações gerais

Você deverá observar as seguintes orientações na implementação deste exercício:

- 1) Apesar da completa compatibilidade entre as linguagens de programação C e C++, seu código fonte **não** deverá conter recursos da linguagem C nem ser resultante de mescla entre as duas linguagens, o que é uma má prática de programação. Dessa forma, deverão ser utilizados **estritamente** recursos da linguagem C++.
- 2) Durante a compilação do seu código fonte, você deverá habilitar a exibição de mensagens de aviso (*warnings*), pois elas 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.) e documente-o adequadamente na forma de comentários. A título de sugestão, anote o seu código fonte para dar suporte à geração automática de documentação utilizando a ferramenta Doxygen (<http://www.doxygen.org/>). Consulte o documento extra disponibilizado na Turma Virtual do SIGAA com algumas instruções acerca do padrão de documentação e uso do Doxygen.
- 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.
- 5) Lembre-se de aplicar boas práticas de modularização, em termos da implementação de diferentes funções e separação entre arquivos cabeçalho (.h) e corpo (.cpp).
- 6) Garanta o uso consistente de alocação dinâmica de memória. Para auxiliá-lo nesta tarefa, você pode utilizar o Valgrind (<http://valgrind.org/>) para verificar se existem problemas de gerenciamento de memória.

Autoria e política de colaboração

Este trabalho deverá ser realizado **individualmente**. O trabalho em cooperação entre estudantes da turma é estimulado, sendo admissível a discussão de ideias e estratégias. Contudo, tal interação não deve ser entendida como permissão para utilização de (parte de) código fonte de colegas, o que pode caracterizar situação de plágio. Trabalhos copiados em todo ou em parte de outros colegas ou da Internet serão sumariamente rejeitados e receberão nota zero.

Apesar de o trabalho ser feito individualmente, você deverá utilizar o sistema de controle de versões Git no desenvolvimento. Ao final, todos os arquivos de código fonte do repositório Git local deverão estar unificados em um repositório remoto Git hospedado em algum serviço da Internet, a exemplo do GitHub, Bitbucket, Gitlab ou outro de sua preferência. A fim de garantir a boa manutenção de seu repositório, configure corretamente o arquivo `.gitignore` em seu repositório Git.

Entrega

Você deverá submeter um único arquivo compactado no formato `.zip` contendo todos os códigos fonte resultantes da implementação deste exercício, sem erros de compilação e devidamente testados e documentados, **até as 23:59 do dia 25 de junho de 2017** através da opção *Tarefas* na Turma Virtual do SIGAA. Você deverá ainda informar, no campo *Comentários* do formulário de submissão da tarefa, o endereço do repositório Git utilizado.

Avaliação

O trabalho será avaliado sob os seguintes critérios: (i) utilização correta dos conteúdos vistos anteriormente e nas aulas presenciais da disciplina para o tratamento de exceções; (ii) a corretude da execução do programa implementado, que deve apresentar saída em conformidade com a especificação e as entradas de dados fornecidas, (iii) a aplicação correta de boas práticas de programação, incluindo legibilidade, organização e documentação de código fonte, e; (iv) **relatório da atividade – A AUSÊNCIA DO RELATÓRIO IMPLICARÁ NA ATRIBUIÇÃO DE NOTA ZERO A ESTA ATIVIDADE – Cabe lembrar que não basta enviar um arquivo qualquer!** 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. Este trabalho contabilizará nota de até 2,0 pontos na 3ª Unidade da disciplina.

Questão Única

Implemente melhorias à sua biblioteca, desenvolvida no Laboratório 9, utilizando tratamento de exceções. Uma vez que o objetivo principal do Laboratório 9 era colocar em prática conceitos de bibliotecas e namespace na linguagem de programação C++, o tratamento de erros foi relegado. Neste exercício você deverá identificar situações, nas operações de seus algoritmos e TADs, que possam levar a erros (leitura de valores, valores passados como parâmetro, alocação de memória, leitura de arquivos, etc) e aplicar o correto tratamento para a interceptação ou lançamento do erro, caso ocorra.

Para garantir que a sua implementação funciona adequadamente, implemente testes para cada situação de erro apontada.

Você deverá apresentar ainda um relatório detalhado, descrevendo cada situação de erro considerada e apontando o tratamento aplicado como solução. Este é um item obrigatório e a sua ausência implicará em nota ZERO nesta atividade.