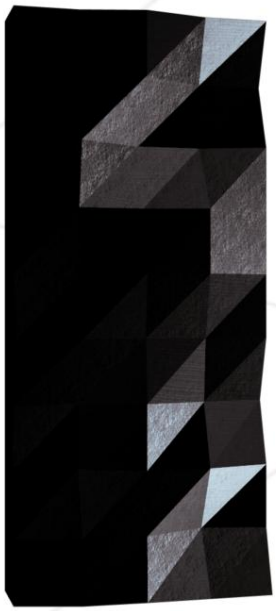

C++ - Módulo 07

Modelos C++

Resumo:
Este documento contém os exercícios do Módulo 07 dos módulos C++.

Versão: 7

Conteúdo

Capítulo I

Introdução

C++ é uma linguagem de programação de propósito geral criada por Bjarne Stroustrup como uma extensão da linguagem de programação C, ou "C com Classes" (fonte: Wikipedia).

O objetivo destes módulos é apresentar a **Programação Orientada a Objetos.**
Este será o ponto de partida da sua jornada C++. Muitas linguagens são recomendadas para aprender POO. Decidimos escolher C++, pois é derivado do seu velho amigo C.
Como esta é uma linguagem complexa e para manter as coisas simples, seu código estará em conformidade com o padrão C++98.

Sabemos que o C++ moderno é muito diferente em muitos aspectos. Então, se você quer se tornar um desenvolvedor C++ proficiente, cabe a você ir além depois do 42 Common Core!

Capítulo II

Regras gerais

Compilando

- Compile seu código com c++ e os sinalizadores -Wall -Wextra -Werror
- Seu código ainda deve compilar se você adicionar o sinalizador -std=c++98

Convenções de formatação e nomenclatura

- Os diretórios dos exercícios serão nomeados desta forma: ex00, ex01, ... , exn
- Nomeie seus arquivos, classes, funções, funções de membro e atributos conforme necessário em as diretrizes.
- Escreva nomes de classes no formato **UpperCamelCase .** Arquivos contendo código de classe serão sempre será nomeado de acordo com o nome da classe. Por exemplo: ClassName.hpp/ClassName.h, ClassName.cpp ou ClassName.tpp. Então, se você tiver um arquivo de cabeçalho contendo a definição de uma classe "BrickWall" representando uma parede de tijolos, seu nome será BrickWall.hpp.
- A menos que especificado de outra forma, todas as mensagens de saída devem ser encerradas por uma nova linha caractere e exibido na saída padrão.
- *Adeus Norminette!* Nenhum estilo de codificação é imposto nos módulos C++. Você pode seguir seu favorito. Mas tenha em mente que um código que seus avaliadores pares não conseguem entender é um código que eles não conseguem classificar. Faça o seu melhor para escrever um código limpo e legível.

Permitido/Proibido

Você não está mais codificando em C. Hora de C++! Portanto:

- Você tem permissão para usar quase tudo da biblioteca padrão. Assim, em vez de ficar preso ao que você já sabe, seria inteligente usar o máximo possível as versões C++-ish das funções C com as quais você está acostumado.
- No entanto, você não pode usar nenhuma outra biblioteca externa. Isso significa que as bibliotecas C++11 (e formas derivadas) e Boost são proibidas. As seguintes funções também são proibidas: *printf(), *alloc() e free(). Se você usá-las, sua nota será 0 e pronto.

- Observe que, a menos que explicitamente indicado de outra forma, o uso do namespace <ns_name> e palavras-chave de amigos são proibidas. Caso contrário, sua nota será -42.
- **Você tem permissão para usar o STL somente no Módulo 08 e 09.** Isso significa: nenhum **Container** (vetor/lista/mapa/e assim por diante) e nenhum **Algoritmo** (qualquer coisa que exija incluir o cabeçalho <algoritmo>) até então. Caso contrário, sua nota será -42.

Alguns requisitos de design

- Vazamento de memória ocorre em C++ também. Quando você aloca memória (usando o novo palavra-chave), você deve evitar **vazamentos de memória.**
- Do Módulo 02 ao Módulo 09, suas aulas devem ser elaboradas na **Língua Ortodoxa Forma canônica, exceto quando explicitamente indicado de outra forma.**
- Qualquer implementação de função colocada em um arquivo de cabeçalho (exceto para modelos de função) significa 0 para o exercício.
- Você deve ser capaz de usar cada um dos seus cabeçalhos independentemente dos outros. Assim, eles devem incluir todas as dependências de que precisam. No entanto, você deve evitar o problema de inclusão dupla adicionando **guardas de inclusão.** Caso contrário, sua nota será 0.

Leia-me

- Você pode adicionar alguns arquivos adicionais se precisar (por exemplo, para dividir seu código). Como essas atribuições não são verificadas por um programa, sinta-se à vontade para fazê-lo, desde que entregue os arquivos obrigatórios.
- Às vezes, as diretrizes de um exercício parecem curtas, mas os exemplos podem mostrar requisitos que não estão explicitamente escritos nas instruções.
- Leia cada módulo completamente antes de começar! Sério, faça.
- Por Odin, por Thor! Use seu cérebro!!!

Você terá que implementar muitas classes. Isso pode parecer tedioso,
a menos que você consiga criar um script no seu editor de texto favorito.

Você tem uma certa liberdade para completar os exercícios.
No entanto, siga as regras obrigatórias e não seja preguiçoso. Você iria
perca muitas informações úteis! Não hesite em ler sobre
conceitos teóricos.

Capítulo III

Exercício 00: Comece com algumas funções

	Exercício: 00
Comece com algumas funções	
Diretório de entrega: ex00/	
Arquivos para entrega: Makefile, main.cpp, qualquer que seja.{h, hpp}	
Funções proibidas: Nenhuma	

Implemente os seguintes modelos de função:

- swap: Troca os valores de dois argumentos fornecidos. Não retorna nada.
- min: Compara os dois valores passados em seus argumentos e retorna o menor um. Se os dois forem iguais, então ele retorna o segundo.
- max: Compara os dois valores passados em seus argumentos e retorna o maior. Se os dois forem iguais, ele retorna o segundo.

Essas funções podem ser chamadas com qualquer tipo de argumento. O único requisito é que os dois argumentos devem ter o mesmo tipo e devem suportar todos os operadores de comparação.

Os modelos devem ser definidos nos arquivos de cabeçalho.

Executando o seguinte código:

```
Inteiro principal( vazio ) {

    int a = 2;
    int b = 3;

    ::trocar( a, b );
    std::cout << "a = " << a << ", b = " << b << std::endl;
    std::cout << "min( a, b ) = " << ::min( a, b ) << std::endl;
    std::cout << "máx.(a, b) = " << ::máx.(a, b) << std::endl;

    std::string c = "cadeia1";
    std::string d = "cadeia2";

    ::trocar(c, d);
    std::cout << "c = " << c << ", d = " << d << std::endl;
    std::cout << "min( c, d ) = " << ::min( c, d ) << std::endl;
    std::cout << "max( c, d ) = " << ::max( c, d ) << std::endl;

    retornar 0;
}
```

Deve produzir:

```
a = 3, b = 2
min(a, b) = 2
máx(a, b) = 3
c = cadeia2, d = cadeia1
min(c, d) = cadeia1
max(c, d) = cadeia2
```

Capítulo IV

Exercício 01: Iter

	Exercício: 01
Iter	
Diretório de entrega: ex01/	
Arquivos para entrega: Makefile, main.cpp, iter.{h, hpp}	
Funções proibidas: Nenhuma	

Implemente um modelo de função iter que recebe 3 parâmetros e não retorna nada.

- O primeiro parâmetro é o endereço de um array.
- O segundo é o comprimento da matriz.
- A terceira é uma função que será chamada em cada elemento do array.

Entregue um arquivo main.cpp que contenha seus testes. Forneça código suficiente para gerar um executável de teste.

Seu modelo de função iter deve funcionar com qualquer tipo de array. O terceiro parâmetro pode ser um modelo de função instanciado.

Capítulo V

Exercício 02: Matriz

	Exercício: 02
Variedade	
Diretório de entrega: ex02/	
Arquivos para entrega: Makefile, main.cpp, Array.{h, hpp} e arquivo opcional: Array.tpp Funções proibidas: Nenhuma	

Desenvolva um modelo de classe **Array** que contenha elementos do tipo T e que implemente o seguinte comportamento e funções:

- Construção sem parâmetro: Cria um array vazio.
- Construção com um unsigned int n como parâmetro: Cria um array de n elementos inicializado por padrão.
  Dica: Tente compilar int * a = new int(); e então exiba *a.
- Construção por operador de cópia e atribuição. Em ambos os casos, modificando o o array original ou sua cópia após a cópia não deve afetar o outro array.
- Você DEVE usar o operador new[] para alocar memória. Alocação preventiva (alocar memória antecipadamente) é proibida. Seu programa nunca deve acessar memória não alocada.
- Os elementos podem ser acessados através do operador subscrito: [ ].
- Ao acessar um elemento com o operador [ ], se seu índice estiver fora dos limites, uma std::exception será lançada.
- Uma função membro size() que retorna o número de elementos no array. Esta função membro não aceita parâmetro e não deve modificar a instância atual.

Como de costume, certifique-se de que tudo funcione conforme o esperado e entregue um arquivo main.cpp que contenha seus testes.

Capítulo VI

Submissão e avaliação por pares

Entregue sua tarefa no seu repositório Git como de costume. Apenas o trabalho dentro do seu repositório será avaliado durante a defesa. Não hesite em verificar novamente os nomes das suas pastas e arquivos para garantir que estejam corretos.

16D85ACC441674FBA2DF65190663F43A243E8FA5424E49143B520D3DF8AF68036E47
114F20A16827E1B16612137E59ECD492E468BC6CD109F65388DC57A58E8942585C8
D193B96732206