



C++ - Módulo 00

Namespaces, classes, funções de membro, fluxos stdio, listas de inicialização, estáticas, const e algumas outras básicas coisa

Resumo:

Este documento contém os exercícios do Módulo 00 dos módulos C++.

Versão: 9

Conteúdo

EU	Introdução	2
II	Regras gerais	3
III	Exercício 00: Megafone	6
4	Exercício 01: Minha incrível lista telefônica	7
V	Exercício 02: O emprego dos seus sonhos	9
VI	Submissão e avaliação por pares	11

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: [Wikipédia](#)).

O objetivo destes módulos é apresentá-lo 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 bem diferente em muitos aspectos. Então, se você quer se tornar um desenvolvedor C++ proficiente, cabe a você ir além depois do 42 Common Core!

Você descobrirá novos conceitos passo a passo. Os exercícios aumentarão progressivamente em complexidade.

Capítulo II

Regras gerais

Compilando

- Compile seu código com `g++` e as bandeiras `-Wall -Wextra -Werror`
- Seu código ainda deverá ser compilado se você adicionar o sinalizador `-std=c++98`

Convenções de formatação e nomenclatura

- Os diretórios de exercícios serão nomeados desta forma: `ex00`, `ex01`, ... , `ex`
- Nomeie seus arquivos, classes, funções, funções de membro e atributos conforme exigido nas diretrizes.
- Escreva os nomes das classes em **UpperCamelCase** formato. Arquivos contendo código de classe sempre serão nomeados de acordo com o nome da classe. Por exemplo:
`NomeDaClasse.hpp`/`NomeDaClasse.h`, `NomeDaClasse.cpp`, ou `NomeDaClasse.hpp`. Então, se você tiver um arquivo de cabeçalho contendo a definição de uma classe "BrickWall" que representa uma parede de tijolos, seu nome será `Parede de tijolos.hpp`.
- A menos que especificado de outra forma, todas as mensagens de saída devem ser encerradas por um caractere de nova linha e exibidas 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 C++11 (e formas derivadas) e **impulsionar bibliotecas** são proibidas. As seguintes funções também são proibidas: `*printf()`, `*alocação()` e `free()`. Se você usá-los, sua nota será 0 e pronto.

- Note que, salvo indicação explícita em contrário, ousando namespace <ns_name> e amigo palavras-chave são proibidas. Caso contrário, sua nota será -42.
- **Você tem permissão para usar o STL somente nos Módulos 08 e 09.** Isso significa: não **Recipientes** (vetor/lista/mapa/e assim por diante) e não **Algoritmos** (qualquer coisa que exija incluir o <algoritmo>header) até então. Caso contrário, sua nota será -42.

Alguns requisitos de design

- Vazamento de memória também ocorre em C++. 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 no **Forma canônica ortodoxa, exceto quando explicitamente declarado 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 **incluir guardas**. 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!!!



Em relação ao Makefile para projetos C++, as mesmas regras do C se aplicam (veja o capítulo Norm sobre o Makefile).




Você terá que implementar muitas classes. Isso pode parecer tedioso, a menos que você consiga criar scripts para 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 perderia muitas informações úteis! Não hesite em ler sobre conceitos teóricos.

Capítulo III

Exercício 00: Megafone

	Exercício: 00
Megafone	
Diretório de entrega:ex00/	
Arquivos para entregar:Makefile, megafone.cpp	
Funções proibidas:Nenhum	

Só para ter certeza de que todos estão acordados, escreva um programa que se comporte da seguinte maneira:


```
$>./megaphone "shhhhh... Acho que os alunos estão dormindo..."
SHHHHH... ACHO QUE OS ALUNOS ESTÃO DORMINDO...
$>./megaphone Droga " ! " "Desculpem, alunos, pensei que essa coisa estava errada." DROGA !
DESCULPEM, ALUNOS, PENSEI QUE ESSA COISA ESTAVA desligada.
$>./megafone
* RUÍDO DE FEEDBACK ALTO E INSUPORTÁVEL * $>
```



Resolva os exercícios no estilo C++.

Capítulo IV

Exercício 01: Minha incrível lista telefônica

	Exercício: 01
Minha incrível lista telefônica	
Diretório de entrega:ex01/	
Arquivos para entregar:Makefile, *.cpp, *.{h, hpp}	
Funções proibidas:Nenhum	

Bem-vindo aos anos 80 e sua tecnologia inacreditável! Escreva um programa que se comporte como um software de lista telefônica de baixa qualidade.

Você precisa implementar duas classes:

- **Lista telefônica**

- Possui uma variedade de contatos.
- Ele pode armazenar no máximo **8 contatos**. Se o usuário tentar adicionar um 9º contato, substitua o mais antigo pelo novo.
- Observe que a alocação dinâmica é proibida.

- **Contato**

- Significa um contato da lista telefônica.

No seu código, a lista telefônica deve ser instanciada como uma instância do **Lista telefônica** classe. O mesmo vale para os contatos. Cada um deles deve ser instanciado como uma instância do **Contato** class. Você é livre para projetar as classes como quiser, mas tenha em mente que tudo o que sempre será usado dentro de uma classe é privado, e que tudo o que pode ser usado fora de uma classe é público.



Não se esqueça de assistir aos vídeos da intranet.

Na inicialização do programa, a lista telefônica está vazia e o usuário é solicitado a digitar um dos três comandos. O programa aceita apenas ADD, SEARCH e EXIT.

- **ADICIONAR:** salvar um novo contato
 - Se o usuário digitar esse comando, ele será solicitado a inserir as informações do novo contato um campo de cada vez. Depois que todos os campos forem preenchidos, adicione o contato à lista telefônica.
 - Os campos de contato são: primeiro nome, sobrenome, apelido, número de telefone e segredo mais obscuro. Um contato salvo não pode ter campos vazios.
- **PROCURAR:** exibir um contato específico
 - Exibir os contatos salvos como uma lista de **4 colunas**: índice, nome, sobrenome e apelido.
 - Cada coluna deve ser **10 caracteres** largo. Um caractere de pipe ('|') os separa. O texto deve ser alinhado à direita. Se o texto for maior que a coluna, ele deve ser truncado e o último caractere exibível deve ser substituído por um ponto ('.').
 - Em seguida, solicite novamente ao usuário o índice da entrada a ser exibida. Se o índice estiver fora do intervalo ou incorreto, defina um comportamento relevante. Caso contrário, exiba as informações de contato, um campo por linha.
- **SAÍDA**
 - O programa fecha e os contatos são perdidos para sempre!
- **Qualquer outra entrada é descartada.**

Uma vez que um comando foi executado corretamente, o programa espera por outro. Ele para quando o usuário insere EXIT.


Dê um nome relevante ao seu executável.



<http://www.cplusplus.com/reference/string/string/> e claro <http://www.cplusplus.com/reference/iostream/>

Capítulo V

Exercício 02: O emprego dos seus sonhos

	Exercício: 02
O emprego dos seus sonhos	
Diretório de entrega:ex02/	
Arquivos para entregar:Makefile, Conta.cpp, Conta.hpp, testes.cpp	
Funções proibidas:Nenhum	



Account.hpp, testes.cpp e o arquivo de log estão disponíveis para download na página de intranet do módulo.

Hoje é seu primeiro dia em *GlobalBanksters Unidos*. Depois de passar com sucesso nos testes de recrutamento (graças a alguns *Microsoft Office* truques que um amigo lhe mostrou), você se juntou à equipe de desenvolvimento. Você também sabe que o recrutador ficou surpreso com a rapidez com que você instalou *Leitor Adobe*. Esse pequeno extra fez toda a diferença e ajudou você a derrotar todos os seus oponentes (também conhecidos como os outros candidatos): você conseguiu!

De qualquer forma, seu gerente acabou de lhe dar um trabalho para fazer. Sua primeira tarefa é recriar um arquivo perdido. Algo deu errado e um arquivo de origem foi excluído por engano. Infelizmente, seus colegas não sabem o que Git é e usar chaves USB para compartilhar código. Neste ponto, faria sentido deixar este lugar agora mesmo. No entanto, você decide ficar. Desafio aceito!

Seus colegas desenvolvedores lhe dão um monte de arquivos. Compilando testes.cpp revela que o arquivo ausente é Conta.cpp. Sorte sua, o arquivo de cabeçalho Conta.hpp foi salvo. Há também um arquivo de log. Talvez você possa usá-lo para entender como o **Conta** classe foi implementada.

Você começa a recriar oConta.cpparquivo. Em apenas alguns minutos, você codifica algumas linhas de puro C++ incrível. Após algumas compilações com falha, seu programa passa nos testes. Sua saída corresponde perfeitamente àquela salva no arquivo de log (**exceto pelos carimbos de data e hora** que obviamente será diferente, já que os testes salvos no arquivo de log foram executados antes de você ser contratado).

Nossa, você é impressionante!



A ordem em que os destrutores são chamados pode diferir dependendo do seu compilador/sistema operacional. Então seus destrutores podem ser chamados em uma ordem reversa.



Você pode passar neste módulo sem fazer o exercício 02.

Capítulo VI

Submissão e avaliação por pares

Entregue sua tarefa em seu Git repositório como de costume. Apenas o trabalho dentro do seu repositório será avaliado durante a defesa. Não hesite em verificar novamente os nomes dos seus arquivos para garantir que estejam corretos.



????????????? XXXXXXXXXXXX = \$3\$f15bc138aca1e76ec6f4cfd0797ec037