(../) ÿ **Rouille**

ESCALA PARA PROJETO CPP MÓDULO 07

(HTTPS://PROJECTS.INTRA.42.FR/PROJECTS/CPP- MODULE-07)

Você deve avaliar 1 aluno nesta equipe

Introdução

Por favor, cumpra as seguintes regras:

- Permaneça educado, cortês, respeitoso e construtivo durante todo o processo de avaliação. O bem-estar da comunidade depende disso.

- Identificar junto ao aluno ou grupo cujo trabalho é avaliado as possíveis disfunções no seu projeto. Aproveite o tempo para discutir e debater os problemas que possam ter sido identificados.

- Você deve considerar que pode haver algumas diferenças na forma como seus pares entenderam as instruções do projeto e o escopo de suas funcionalidades. Sempre mantenha a mente aberta e avalie-os da forma mais honesta possível. A pedagogia só é útil e somente se a avaliação pelos pares for feita com seriedade.

Diretrizes

- Avaliar somente o trabalho que foi entregue no repositório Git do aluno ou grupo avaliado.

- Verifique novamente se o repositório Git pertence ao(s) aluno(s). Certifique-se de que o projeto é o esperado. Além disso, verifique se 'git clone' é usado em uma pasta vazia.

- Verifique cuidadosamente se nenhum aliase malicioso foi usado para enganá-lo e fazer com que você avalie algo que não seja o conteúdo do repositório oficial.

- Para evitar surpresas e se for o caso, revisem juntos quaisquer scripts utilizados para facilitar a classificação (scripts para testes ou automação).

- Caso não tenha concluído o trabalho que vai avaliar, deverá ler a matéria na íntegra antes de iniciar o processo de avaliação.

- Use os sinalizadores disponíveis para relatar um repositório vazio, um programa que não funciona, um erro de norma, trapaça e assim por diante.
Nestes casos, o processo de avaliação termina e a nota final é 0,

ou -42 em caso de trapaça. Porém, exceto em caso de trapaça, os alunos são fortemente incentivados a revisarem juntos o trabalho entregue, a fim de identificar eventuais erros que não devam ser repetidos no futuro.

- Você nunca deverá editar nenhum arquivo, exceto o arquivo de configuração, se ele existir. Se você quiser editar um arquivo, reserve um tempo para explicar os motivos ao aluno avaliado e certifique-se de que ambos concordam com isso.

- Você também deve verificar a ausência de vazamentos de memória. Qualquer memória alocada no heap deve ser liberada adequadamente antes do final da execução.
Você tem permissão para usar qualquer uma das diferentes ferramentas disponíveis no computador, como leaks, valgrind ou e_fence. Em caso de vazamento de memória, marque o sinalizador apropriado.

Anexos

ex00.cpp (https://github.com/rphlr/42-Subjects/)

ex01.cpp (https://github.com/rphlr/42-Subjects/)

assunto.pdf (https://github.com/rphlr/42-Subjects/)

main.cpp (https://github.com/rphlr/42-Subjects/)

Testes preliminares

Se houver suspeita de trapaça, a avaliação termina aqui. Use o sinalizador "Cheat" para denunciá-lo. Tome esta decisão com calma, sabedoria e, por favor, use este botão com cautela.

Pré-requisitos

O código deve ser compilado em c++ e as flags -Wall -Wextra -Werror Não se esqueça que este projeto deve seguir o padrão C++98. Portanto, funções ou contêineres C++ 11 (e posteriores) NÃO são esperados.

Qualquer um destes significa que você não deve avaliar o exercício em questão:

- Uma função é implementada em um arquivo de cabeçalho (exceto funções de modelo).
- Um Makefile compila sem os sinalizadores necessários e/ou outro compilador que não seja c++.

Qualquer um destes significa que você deve sinalizar o projeto com "Proibido Função":

- Uso de uma função "C" (*alloc, *printf, free).
- Uso de função não permitida nas diretrizes do exercício.
- Uso de "using namespace <ns_name>" ou da palavra-chave "friend".
- Uso de uma biblioteca externa ou recursos de versões diferentes do C++98.

Sim Não

Exercício 00: Comece com algumas funções

Este exercício trata de escrever 3 modelos de funções simples: swap(), min() e max().

Tipos simples

Consulte o assunto para obter a saída esperada com tipos simples, como int.

Sim Não

Tipos complexos

As funções também funcionam com tipos complexos? (verifique com o arquivo ex00.cpp em anexo)

Sim Não

Exercício 01: Iter

Este exercício trata de escrever uma função genérica para iterar por arrays.

Funciona?

Teste o código ex01.cpp (em anexos) com o iter do aluno avaliado.
Se tudo correr bem, deverá exibir:

```
0
1
2
3
4
42
42
42
42
42
```

Sim Não

Exercício 02: Matriz

Este exercício trata de escrever um modelo de classe que se comporte como um array. Se a alocação interna do array real não vier do uso de new[], não avalie este exercício. Peça ao aluno avaliado para provar que o modelo de aula funciona com matrizes de tipos simples e complexos antes de avaliar o exercício.

Construtores

É possível criar um array vazio e um array de tamanho específico?

Sim Não

Acesso

Os elementos devem estar acessíveis para leitura e escrita através do operador[]
(ou lendo apenas se a instância for const). Acesso a um elemento que é
fora do intervalo deve lançar uma std::exception.

Sim Não

Avaliações

Não se esqueça de verificar a bandeira correspondente à defesa

OK Projeto excelente

Trabalho vazio Trabalho incompleto Compilação inválida Trair Colidir

Situação preocupante Vazamentos l Função proibida Não é possível apoiar/explicar o código

Dê uma estrela a este repositório.

(https://github.com/rphlr/42-Evals)