



navegação

[Página Inicial](#)

[Equipe](#)

[Atividades](#)

[Localização](#)

[Contatos](#)

[Área de Membros](#)

[Pastas das Disciplinas](#)

[EEL170 - Computação I](#)

[EEL270 - Computação II](#)

[EEL670 - Linguagens de Programação](#)

[EEL875 - Internet e Arquitetura TCP/IP](#)

[EEL878 - Redes de Computadores I](#)

[EEL879 - Redes de Computadores II](#)

[EEL480 - Sistemas Digitais](#)

[Tutoriais](#)

[Webmail](#)

[Novidades](#)

[Eventos](#)

acessar

Nome do Usuário

Senha

acessar

Esqueceu sua senha?

Aula Prática 13 - Roteiro

19/11/2019 - Roteiro referente à aula prática 13 - Etapa 1.3 do trabalho final.

Prazo: 26/11/2019 - 9:00

- Inclua, no arquivo *"mldlUmlTypes.h"*, a definição do tipo *mldlUmlConfigurationOptionsType*. Este tipo deverá ser uma estrutura contendo os campos correspondentes a todas as opções de configuração da biblioteca. Cada campo deverá ser do tipo apropriado (string, inteiro, etc.). Os campos desta estrutura deverão ser definidos durante o desenvolvimento da biblioteca, sempre que se mostrarem necessários.

- Inclua, no arquivo *"mldlUmlFunctions.h"*, o protótipo da função *MldlUmlGetConfigurationOptionsValues*.

```
mldlUmlErrorType  
MldlUmlGetConfigurationOptionsValues (char *, mldlUmlConfigurationOptionsType *);
```

- Inclua, no arquivo *"mldlUmlFunctions.c"*, a implementação da função *MldlUmlGetConfigurationOptionsValues*. Esta função deverá receber o nome do arquivo de configuração desejado e o endereço de uma variável local do tipo *mldlUmlConfigurationOptionsType*. Cada campo desta variável deverá receber inicialmente o seu valor padrão, ou seja, o valor que foi definido para a macro de configuração no arquivo *"mldlUmlConfig.h"* ou o valor definido através da opção *-D* no arquivo de dependências (lembrando que este último tem prioridade sobre o primeiro). A seguir o arquivo de configuração (*"mldl-uml.cfg"* ou o arquivo indicado) deverá ser lido. Cada opção de configuração lida deste arquivo deverá ser atualizada com este novo valor. Veja [Arquivo de Configuração](#).

- Escreva o arquivo *"mldlUml.c"* contendo o código do programa necessário para tratar todas as opções curtas e longas utilizadas pelas função primárias (ver [Funções Primárias](#) e [Opções e Argumentos de Linha de Comando](#)).

Este programa deverá ser implementando utilizando as funções *"getopt_long"* e *"getsubopt"* e deverá ser capaz de identificar qual a opção de linha de comando (no formato curto ou no formato longo) dentre as definidas em [Funções Primárias](#) e em [Opções e Argumentos de Linha de Comando](#) foi passada para o programa (função *getopt_long*). O programa deverá aceitar uma única opção por execução.

Além disso, o programa deverá recuperar, utilizando a função *"getsubopt"*, os valores dos argumentos recebidos no formato nome=valor. Estes valores deverão ser recebidos através de diferentes argumentos de linha de comando (diferentes argv's), ou seja, deverão estar separados por espaço (não será aceita a versão em que são separados por vírgula). Os valores recebidos deverão ser armazenados nas variáveis correspondentes, que deverão ser do tipo string (definidos como vetores de char), com exceção do idioma. Quando necessário, estas *strings* deverão ser convertidas para os tipos correspondentes.

Observações:

- É decisão de projeto aceitar mais de uma opção por execução.

- Crie uma variável (do tipo *string* = vetor de caracteres) para cada possível argumento do programa e inicialize essa variável com uma *string* de comprimento zero ("").

A exceção a esta sugestão fica por conta do idioma, que deverá ser definido como uma variável do tipo *xyzLanguageType* e deverá ser inicializado com o valor *"mldlUmlEnglish"* ou com o valor correspondente ao número de idiomas contemplados na implementação (significando idioma não definido). Neste último caso, a interface deverá ser construída utilizando o idioma padrão.

Os valores das variáveis correspondentes aos argumentos deverão ser modificados sempre que um novo valor for recebido via argumento de linha de comando, ou seja, sempre que for recebido através da função *getsubopt*.

Durante o processamento dos argumentos as variáveis correspondentes deverão ser atualizadas. Após o processamento de todos os argumentos, as variáveis não atualizadas estarão com o valor padrão, se existente, ou com o valor indicando que aquele argumento não foi definido pelo usuário. Se o argumento for obrigatório e não foi definido pelo usuário, a condição de erro correspondente deverá ser indicada. Lembre-se que este programa deverá usar as configurações corretas (arquivo de configuração, definições no arquivo de dependências e valores definidos no arquivo *"mldlUmlConfig.h"*, com o primeiro tendo a prioridade mais alta.

Após identificar a opção (curta ou longa) selecionada pelo usuário, e a consequente identificação e validação dos argumentos, coloque um comentário com o nome da função principal que deverá ser chamada para a opção desejada.

Lembre-se que os argumentos contendo os pares nome e valor podem ser recebidos em qualquer ordem. O que está entre colchetes é opcional. O que está em itálico é o nome do argumento, enquanto que o que está em negrito é um valor fixo. O que está entre os sinais de menor e maior deverá conter o valor desejado pelo usuário, como por exemplo, o nome completo do usuário.

- Escreva o arquivo *"mldlUmlRunNcursesInterface.h"* contendo o protótipo da função de mesmo nome (ver [Funções Primárias](#)).

- Escreva o arquivo *"mldlUmlRunNcursesInterface.c"* contendo a implementação da função de mesmo nome (ver [Funções Primárias](#)). Esta função deverá validar os argumentos recebidos e depois deverá criar todas as janelas necessárias para a chamada de todas as funções primárias definidas, ou seja, todas as janelas com menus, formulários e mensagens correspondentes às funções primárias.



| ◀ Novembro 2019 ▶ | | | | | | |
|-------------------|----|----|----|----|----|----|
| Do | Se | Te | Qu | Qu | Se | Sa |
| | | | | | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 |

| notícias |
|---|
| Horário 2017/2 31/07/2017 |
| Estágio Supervisionado - Importante 05/03/2012 |
| Mais notícias... |

