



UNIDADE CURRICULAR: Programação Orientada a Objetos

CÓDIGO: 21093

DOCENTE: Jorge Morais e Leonel Morgado

A preencher pelo estudante

NOME: Pedro Pereira Santos

N.º DE ESTUDANTE: 2000809

CURSO: Licenciatura em Engenharia Informática

DATA DE ENTREGA: 27/11/2022

TRABALHO / RESOLUÇÃO:

Para a criação deste programa utilizei os conceitos aprendidos nos temas dados na UC até ao momento, com ajuda da world wide web para compreender melhor como se usam e funcionam certas funções das bibliotecas standart de C++.

Para começar, fui criando as classes necessárias para o programa, Atividade, para incorporar os atributos de cada classe, segundo o briefing, como o Nome (Título), descrição, etc, e Lista, para criar uma lista que guardasse todas as Atividades adicionadas/existentes, segundo os requisitos pedidos.

Decidi fazer uma interface, com menu, para acesso às diferentes opções oferecidas pelo programa com escolha por input de um número que equivale à opção que se quer tomar. Estava com problemas quanto ao input recebido ficar com bug por inserir uma letra ou mais do que um caractere, neste caso algarismo, criando assim uma pequena função para filtrar o input, recebendo-o em “char” e usando o código ASCII de 0 (48) para verificar se o valor inserido é de um número ou não. Depois para as diferentes opções, utilizei o comando switch com as várias opções possíveis a tomar, fazendo outras pequenas decisões, para focar no requerido.

Para a classe Atividade, utilizei dois construtores, um para criar a atividade em si com os seus atributos e outro para criar uma Atividade “vazia”, para servir como recipiente para atribuir outra Atividade. Os atributos da classe são os pedidos no briefing.

Para os métodos, implementei os básicos, para obter os valores ou definir o valor de cada atributo. Implementei também um método para mostrar a atividade requerida e 3 métodos privados, para ajudar a criação do objeto, para verificar se os critérios numéricos estão dentro dos valores permitidos (0 a 100%, sendo estes os valores para facilitar a sua inserção), sendo, com outro método, convertidos para o “standart” requerido (entre 0 e 1, inclusive) e por fim um método para verificar se os critérios de url começam como foi requerido.

Por fim, o método de sobrecarga do operador + para juntar 2 atividades diferentes.

Para a sobrecarga de operador, decidi juntar os nomes, descrições e urls das duas atividades, fazendo depois o requerido de manter o valor mais elevado de cada atributo.

Para a classe Lista, decidi utilizar um vetor para fazer a listagem, uma vez que é uma estrutura de dados que valora a velocidade e de fácil utilização e acesso, com auxílio de uma variável iteradora.

Para os métodos, criei tais para a ordenação de lista, mostrar a lista, adicionar e combinar atividades, tendo criado mais 2 métodos para auxílio, um que verifica se a lista está vazia e outro, privado, para auxiliar na ordenação da lista.

O método de ordenação, que decidi em forma ascendente do valor do critério escolhido para tal, envolve a comparação de duas atividades, e, se o valor da atividade “atual” for inferior à atividade a compara, trocam de posição com a superior a ir para “cima” na lista, ou, se o valor for igual, ordena por forma alfabética em relação ao Nome da Atividade (Título), até avaliar a lista toda, comparando assim todas as atividades, sem saltar nenhuma.

O método para adicionar uma nova atividade, decidi primeiro criar uma atividade, e, caso os critérios estejam dentro do requerido e esta ainda não exista na lista, adiciona-a, caso contrário, envia um erro e destrói a atividade criada.

O método de combinar atividades verifica se as atividades pedidas a combinar existem na lista e se estas são as mesmas. Caso existam e sejam diferentes, adiciona-as, com o operador + em sobrecarga e “empurra” a atividade para o fim da lista.