

Arthur Antunes Santos Silva

Dilvonei Alves Lacerda Junior

Lucas Gonçalves Nojiri

INTRODUÇÃO

Foi proposto a execução de uma agenda para professores do DCOMP em linguagem c. Agenda que deve conter uma lista de compromissos, que seriam aulas, eventos, reuniões, compromisso particular e orientação. A agenda e os compromissos são tipos de dados que contem as informações de acordo as especificações dadas, que são datas, horários, nome do compromisso, identificador, id do professor, etc. E para receber e manipular tais informações de forma coerente é funções para realizar cada empecilho que possa haver, como por exemplo conflitos entre compromissos, cancelar uma reunião, inserir um compromisso na agenda.

IMPLEMENTAÇÃO

Inicialmente é declarado o tipo abstrato “Compromisso” contendo todas as informações necessárias para identificar o compromisso, o nome do compromisso, o tipo do compromisso (orientação, aula, compromisso particular, evento, reunião), um identificador único, um status que pode ser a “ser cumprido”, “adiado” ou “cancelado”, a data, o horário e a duração. Com o tipo compromisso criado é possível fazer todas as funções a seguir.

“showmenu”:

Permite ao usuário escolher qual das opções ele quer selecionar para iniciar o programa.

“criaAgenda”:

Esta função solicita que o usuário digite o nome do professor, seu ID e o ano que será criada a agenda.

“inicializa compromisso”:

Recebe como parâmetro dois compromissos e atribui um identificador numérico único para um, sendo um deles um compromisso criado anteriormente e retorna o identificador único do anterior mais um, no caso de não existir um compromisso anterior o valor um é atribuído ao identificador.

“altera_prioridade”:

Recebe como parâmetro um compromisso e a nova prioridade, faz a alteração no tipo de prioridade que recebe a novo tipo e a passa como parâmetro.

“retorna_prioridade”:

Recebe como parâmetro um compromisso e retorna o tipo deste, ou seja, retorna se determinado compromisso é orientação, aula, compromisso particular, evento ou reunião.

“Eadiavel”:

Recebe como parâmetro um compromisso e uma flag, sendo que o parâmetro flag deverá ser “TRUE” se for adiável e “FALSE” se não, o que define se um compromisso é adiável. É o seu tipo, sendo do tipo aula ou evento eles não podem ser adiados, os demais podem ser adiados, logo temos “FALSE” para flag se for do tipo aula ou evento, se não flag é “TRUE”.

“atribuiStatus”:

Recebe como parâmetro um compromisso e uma flag, parecida com a função anterior, e retorna o status de cancelado, adiado ou a ser cumprido.

“retornaStatus”:

Recebe um compromisso e retorna o status que está atribuído para o compromisso.

“temConflito”:

Verifica de acordo ao seu tipo se existe um conflito entre dois compromissos através de suas datas, em compromissos no mesmo dia são comparadas as horas adicionando a duração em horas. No caso de eventos compara-se a data do evento mais sua duração em dias e retorna se existe conflito ou não.

“imprimeCompromisso”:

Imprime todas as informações que estão associadas a um compromisso, todas essas funções ficam em um arquivo.h nomeado de compromissos.h que depois de implementado passamos para agenda.h.

No nosso tipo agenda foi declarado um tipo inteiro chamado id da agenda de determinado professor, um vetor de char que para receber o nome do professor que vai utilizar aquela agenda, o ano da agenda e um vetor do tipo compromissos, ou seja uma lista de compromissos, que vão ser todos os eventos que vão ser marcados na agenda pelo professor.

“imprimeCompromissosAdiados”:

A função recebe um parâmetro com tipo agenda, percorre a agenda e imprime todos os compromissos com status adiável (1).

“imprimeCompromissosCancelados”:

A função recebe um parâmetro com tipo agenda, percorre a agenda e imprime todos os compromissos com status Cancelados (2).

“imprimeCompromissosACumprir”:

A função recebe um parâmetro com tipo agenda, percorre a agenda e imprime todos os compromissos com status Acumprir (0).

“converte_duracao”:

Cria uma nova agenda com o nome e o identificador informados, e o ano da agenda, cria uma agenda vazia, sem compromissos.

“insereCompromisso”:

Recebe como parâmetro uma agenda e todas as informações de um compromisso e o insere na agenda e imprime uma mensagem de sucesso ou falha, em caso de sucesso também imprime o identificador.

“tipo_de_compromisso_string”:

Esta função é referente aos tipos de compromisso com o uso de switch, foi dado um numero para cada tipo de 1 – 4, respectivamente a orientação, aula, evento e reunião.

“converte_hora_dia_minuto”:

Calcula o horário dos compromissos, fazendo a conversão das horas do dia de acordo com os dias e minutos do horário registrado.

“buscarAgenda”:

Percorre todo o programa buscando uma agenda especifica, de acordo com os parâmetros da função como ID do professor.

“recuperaAgenda”:

Recebe como parâmetro uma agenda e uma data e retorna uma mensagem na tela com o nome do professor, ano, e o numero de compromissos na agenda que são após a data passada.

“removeCompromisso”:

Para remover os compromissos que já foram inseridos é utilizado a função “removeCompromisso” esta recebe como parâmetro uma agenda e o identificador de um compromisso, e remove o compromisso com identificador passado caso ele exista.

“imprimeAgenda”:

Para imprimir a agenda, é feito a chamada da função “imprimeAgenda”, ela imprime todos os compromissos agendados de um professor, ordenados cronologicamente pelo instante do início, caso exista empate, imprime-se o primeiro os compromissos com maior prioridade, e se ainda houver alguma empate respeita-se a ordem evento, reunião, aula, orientação e compromisso particular, foi utilizado o modo selection sort para ordenação do compromissos. .

“imprimeNadiamentos”:

Verifica os que estão status de adiados, soma mais um a variável número de adiados e a retorna no final, recebe uma agenda.

“retornaNcancelamentos”

Conta os que estão com status de cancelado e retorna à quantidade de cancelados, semelhante a função anterior, recebe uma agenda.

“retornaNcompromissos”:

Similar as duas funções anteriores, porem conta todos que tem status a ser cumprido, recebe como parâmetro uma agenda para fazer a contagem.

Temos também uma função para imprimir cada uma das quantidades obtidas nas três funções anteriores que são as respectivas funções, “imprimiCompromissosAdiados, imprimiCompromissosCancelados e imprimiCompromissosAsercumprido” que imprimem a quantidade adiados, cancelados e a ser cumpridos.

CONCLUSÃO

Durante o trabalho prático proposto tivemos alguns pontos a levantar entre eles a dificuldade em algumas funções, por exemplo a função resolver conflito a qual não conseguimos fazer-la. Além dessa função as funções recuperar “agenda” e tem “conflito” foram outras das quais tivemos dificuldade.