Projeto PPP

Introdução:

O projeto que foi proposto da cadeira de princípios de programação solicitou-nos para desenvolver uma aplicação para auxiliar um funcionário de uma oficina. A aplicação serve para guardar a informação do cliente, as horas e também que tipo de programa quer, no caso de manutenção ou lavagem. Foi imposto regras e condições para realizar o trabalho.

Desenvolvimento:

O meu trabalho está dividido em três partes: No PPP.main onde se trata da consola e também da interface da aplicação onde é usado o switch para tal. Dentro do switch é chamado várias funções onde é tratado no PPP.C. No PPP.h onde é mais focado nas estruturas e listas ligadas.

Começando na raiz do trabalho é criado no PPP.h a estrutura com o nome “Reserva”, onde esta presente todos os dados que precisamos acerca do cliente. Para o tal foi criado uma outra estrutura com o nome “noReserva” onde contem a estrutura Reserva e uma seta com o nome “\*prox” que serve para apontar para uma outra lista, tendo listas ligadas. Contudo, foi criado outra estrutura chamada tipoReserva que contem dentro dela duas estruturas noReserva com o nome inicio e fim, como o nome indica o inicio apontar para o inicio da lista noReserva e o fim para apontar para o fim da lista noReserva.

Começando pelo ficheiro PPP.main, são criadas duas listas com o nome tipoRes e tipoPreRes com o inicio e o fim, igualando-as á função criaTipo, onde se reserva um bloco de memória, apontando o inicio e i fim para NULL, ou seja para nada visto que ainda não foi criado nenhuma lista noReserva. Na parte do switch é onde esta presente a interface da consola/aplicação, onde tem as opções para o utilizador escolher. No caso de se escolher a opção 1 com o nome Adicionar Reserva, são criadas duas noReserva com o nome \*novoNo e \*aux, igualando-as à função cria, onde dentro dessa função é criado um novoNo em mémoria e é inicializado as suas propriedades. No caso do \*aux é criado no caso a reserva novoNo esteja sobreposta noutra reserva, assim passa para pré-reserva. É criado um inteiro chamado reserva que igualamos ao adiciona, passando para a função o novoNo que acabamos de criar para por a reserva, o inicio da da lista das reservas e o aux caso haja sobreposição. Dentro da função do adiciona pedimos ao utilizador para inserir os dados do cliente para realizar a reserva. Após o utilizador inserir os dados como no caso a data, é verificado se data que introduziu esta dentro das normas, como no caso das horas onde é passada para essas funções os dados do cliente onde se verifica se há algum erro na data ou na hora, caso exista dá-se return -1 e liberta-se espaço na memória. Voltando ao main o valido fica igual a -1 e dá-se break. Caso os dados tenham passado nos testes de verificar a data e as horas o programa pede que tipo de serviço o cliente quer, passando para minutos e horas finais. Caso os dados do cliente estiverem todos certos vemos se existe alguma sobreposição usando a função verifica sobreposição, onde passamos a nova reserva que foi criada e o início da lista das reservas para a percorre-la toda. Dentro da função convertemos as horas em minutos para ser mais fácil verificar a sobreposições. É usado um while para percorrer a lista das reservas até ao fim. Caso as datas não coincidirem avançasse para a outra reserva que esta a seguir, se datas coincidirem convertemos em minutos a reserva que esta na lista e comparamos com as duas reservas.Se existir sobreposição da return 0 e igualamos o aux que passamos na função adiciona e libertamos a memória da reservas que passou para pré-reserva caso não tenha ocorrido nenhuma sobreposição da return 1 e insere-se na lista das reservas. Se houver return 1 o valido fica igual a 1e entra-se na função insere com a lista da reserva e a nova reserva que criamos para inserimo-la na lista das reservas. Se na lista das reservas não existir nenhuma reserva , ou seja, tipoRes->inicio==NULL, o inicio fica a apontar para a nossa reserva, como também o fim. Se não for o caso criamos um noReserva aux que é igual ao inicio da lista das reservas e um noReserva \*anterior. Criou-se p while para percorrer a lista, onde dentro desse while criou-se um if se caso a nova reserva que criamos é mais antiga do que a reserva que estamos a analisar entra dentro do if. Dentro desse if obtemos o id da reserva e dentro do if existe outro if caso se a nossa reserva é mais antiga do que a mais antiga da lista das reservas, caso não seja adicionamos no meio entre duas reservas usamos o anterior que fica para apontar para a nossa nova reserva que fica a apontar para a reserva a seguir e assim sucessivamente. Caso não tenha entrado no primeiro if criamos outro if em que a nossa ultima reserva que criamos ser a mais recente , ou seja obtemos o id da nossa nova reserva no obtem maior id, onde nessa função obtemos o id para o cliente, seguidamente a reserva que estava em ultimo da lista das reservas fica a apontar para a nossa nova reserva que criamos que é mais recente e o fim da lista das reserva fica a apontar para a nossa nova reserva. Caso não tenha entrado em nenhum destes ifs passamos para a próxima reserva. Contudo este processamento funciona também para as Pré-reservas em que no insere passamos o tipoPreRes o aux para a função. Passando para o próxima a opção, temos a cancelar reserva na opção 2,que também serva para a opção 8 de realizar reserva, onde pedimos ao utilizador para inserir o do que quer cancelar onde vai para a função eliminar, onde passa a lista das reservas e o id.Dentro dessa função temos o caso se o inicio da lista das reservas for NULL , ou seja não existir nada na lista dizemos que não existe nada. Caso isso não aconteça criamos um aux que é o inicio da lista das reservas. Fazemos um if se caso o id que nos queremos esteja no inicio da lista, pomos que o novo inicio é o prox (tipoRes->inicio=tipoRes->inicio->prox), se houvesse só uma reserva tínhamos que criar uma nova lista das reservas. Se não entrou em nenhum desses ifs criamos o anterior e percorremos a lista até encontramos o id correspondente. Caso o id correspondente esta no fim da lista o fim da lista fica a apontar para o anterior caso não esteja o anterior fica a apontar para o prox da reserva que eliminamos (anterior->prox=aux->prox). Depois de eliminar vamos ver se existe alguma Pré-reserva que dê para passar para reserva. Usando a função verificaPreReservas onde passamos para essa função a lista das Reservas e Pré-reservas. Começamos por criar um aux que igualamos ao inicio das Pre-Reservas. Percorremos uma a uma das Pré-Reservas para verificar se existe qualquer sobreposição com todas as reservas. Criamos o verifica para receber o return da verificasobreposição, e caso o verifica for igual a 0 significa que podemos passar uma pre-reserva para reserva, eliminamos a pre-reserva com o eliminar e depois passamos para o insere para inserimos nas reservas. Caso o verificar não for igual a zero avançamos para outra Pre-reserva. No case 3 que é eliminar Pré-Reserva usamos o mesmo método mas envés ser reservas são Pré-reservas, onde voltamos a usar a função eliminar mas não a verificarPreReserva, porque não precisamos de passar Pré-Reservas para Reservas. No case 4 e 5 servem para imprimir as reservas e Pré-reservas, usando a função imprimir ordenado em que passamos o inicio das listas e utlizamos um while para percorrer a lista e onde temos a função imprime que imprime os dados dos clientes. No case 6 temos a opção para listar a reserva do cliente como também serve para o case 7 que é listar a Pré-Reserva do cliente, onde usamos a função imprime o cliente, onde pedimos o nome do cliente e percorremos a lista até esse nome que inserimos corresponder ao nome que esta na lista das reservas guardado. Depois de obter a reserva chamamos a função imprime e imprimimos os dados do cliente. No case 8 temos realizar tarefa,onde voltamos a pedir o id do cliente, vai a função eliminar, que depois vai a função verificaPrereservas para ver se existe alguma Pré-reserva que dê para inserir nas Reservas. O Case 9 tem o gravar ficheiro que serve para carregar o ficheiro das reservas e pré-reservas, abre-se o ficheiro e percorremos a lista das reservas/Pré-reservas e gravamos no ficheiro.No case 10 carregamos os ficheiros Reserva e Prereserva.