Departamento de Engenharia Informática

Faculdade de Ciências e Tecnologia

Universidade de Coimbra

2009/10

Princípios de Programação Procedimental

CBA

Caixa Bancário Automático

Grupo:		Turma: TP5
Igor Cruz	Nº	2009111924
João Magalhães	Nº	2009112796

Arquitectura do Programa

Para armazenar os dados utilizados no programa foram utilizadas 3 estruturas. São elas:

- **struct_user** – terá como função armazenar o número de conta, PIN, quantia de dinheiro e nível de utilizador. O número de conta é o identificador numérico individual de cada conta, o PIN será utilizado como senha para aceder à conta ou fazer movimentos, a quantia indica a quantidade de dinheiro disponível na conta. E o nível de utilizador é o número que diferenciará as contas em: funcionário (1), cliente(0) ou cartão bloqueado(-1).

- **Dinheiro** - será a estrutura responsável pela gestão do dinheiro na caixa automática. Estão definidas as notas de 50€, 20€, 10€ e 5€.

- **List_node** – estrutura que armazena os nós da lista. Cada nó é constituído por um nó next (seguinte), previous (anterior) e um que aponta para a informação desse nó. As informações de cada conta são armazenadas nos nós.

Descrição das funções do programa:

- Função
  "int main()"

  Utilizando a função *cria_lista_utilizadores()* associa a uma variável global *users* que será a lista duplamente ligada que posteriormente servirá para armazenar as contas dos funcionários e clientes.

  Através da função *carrega_info_ficheiro()* tem acesso ao ficheiro binário *database.bin* que contém a base de dados dos utilizadores da caixa bancário.

  Chama a função *login()* que devolverá um dos seguintes números: -1, 1, 2 ou 3.

  Consoante o número devolvido pela função *login()* pode então suceder:
  (-1) cartão bloqueado, regista a operação através da função *salva_operacao()* e volta a chamar *login()*;
  (1) chama a função *mostra_menu_cliente()*;
  (2) chama a função *mostra_menu_admin()*;
  (3) tentativa errada, regista a operação através da função *salva_operacao()* e retira um tentativa de modo a bloquear o utilizador excedidas as 3 permitidas.

- Função
  “int login (Lista_utilizadores users)”

  Função chamada dentro da *main(),* pede ao utilizador o número de conta e PIN, e devolve (-1) caso a conta esteja bloqueada, (1) caso se trate de um cliente, (2) caso se trate de um funcionário ou (3) se o PIN introduzido for incorrecto.

  Faz uso da função *procura_utilizadores()* para encontrar o utilizador na base de dados e assim verificar os seus dados.

- Função
  “void mostra_menu_utilizador(void)”

  Esta função será responsável por imprimir o menu de cliente e chamar as funções correspondentes à acção que o utilizador pretenda efectuar.

  Um cliente tem acesso ás seguintes opções: *Levantamentos, Transferências, Depósitos, Alteração PIN* e *Abandonar*.

- Função
  “void mostra_menu_admin(void)”

  Esta função será responsável por imprimir o menu de funcionário e chamar as funções correspondentes à acção que o utilizador pretenda efectuar.

  Um funcionário tem acesso ás seguintes opções: *Levantamentos, Transferências, Depósitos, Alteração PIN, Criar Conta, Eliminar Conta, Listar Clientes, Listar Operações, Alterar o número de notas, Imprimir Conta, Bloquear Conta, Desbloquear Contar, Abandonar e Terminar Execução.*

- Função
  “int menu_levantamentos()”

  Função chamada quando o utilizador escolhe a opção *Levantamentos* no menu.

  Começa por verificar através do nível de utilizador se se trata de um funcionário ou de um cliente.

  Caso se trate de um funcionário, pedirá o número da conta onde se fará o levantamento e a quantia a levantar. Por outro lado, se o tipo de conta for a de um cliente apenas pedirá o montante uma vez que assumirá que a conta de onde levantar o dinheiro será a do utilizador em questão.

  Faz uso da função *procura_utilizadores()* para encontrar a conta destinatária na base de dados.

  Uma vez recebida a quantia a levantar e a número de conta de onde levantar, a função utiliza essas informações como parâmetros para as funções *levanta_quantia()* que irá proceder à transferência e *salva_operacao()* que irá registar o sucedido.

- Função
  “void levanta_quantia(Lista_utilizadores destino,double quantia)”

  Função chamada em *menu_levantamentos()*.

  Receberá o número de conta de onde levantar o dinheiro bem como a quantia a levantar por parâmetros.

  Levantará a quantia escolhida tendo a atenção de escolher se possível uma nota de cada valor. Uma vez escolhida uma nota de cada valor, passará a escolher notas de maior valor possível de maneira a minimizar o número de notas a entregar.

  O processo pode terminar com sucesso, notas entregues. Ou sem sucesso, falta de notas na caixa automática ou dinheiro em conta insuficiente.

- Função
  “int menu_transferencias()”

  Função chamada quando o utilizador escolhe a opção *Transferências* no menu.

  Pede o número de conta de onde retirar o dinheiro (apenas se o utilizador for funcionário), o número de conta do destinatário, a quantia a transferir e o PIN do utilizador em questão.

  Utiliza estas informações guardadas em variáveis para chamar a função *transfere()* e proceder à transferência.

- Função
  “int transfere(unsigned int conta_origem, unsigned int conta_dest, double quantia)”

  Recebe da função *menu_transferencias()* a conta de origem, a conta destinatária e a quantia a ser transferida como parâmetros.

  Efectua a transferência de dinheiro, fazendo uso da função *procura_utilizadores()* para encontrar as contas de origem e destino da transferência.

  O processo pode terminar com sucesso, faz a transferência. Ou sem sucesso, conta de origem inexistente ou sem dinheiro suficiente ou conta destinatária inexistente.

- Função
  “int menu_depositos()”

  Função chamada quando o utilizador escolhe a opção *Depósitos* no menu.

  Recebe as informações: conta do destinatário (apenas para funcionários), quantia a depositar e PIN do utilizador. Por fim, pede ao utilizador a quantidade de notas de cada valor que deseja depositar.

  Utiliza a função *procura_utilizadores()* para encontrar a conta onde depositar e da função *salva_operacoes()* para registar o sucedido.

  O processo pode terminar com sucesso, depósito concluindo. Ou sem sucesso, conta a depositar inexistente ou número de notas a depositar não coincidente com o valor desejado.

- Função
  “void menu_alterapin()”

  Função chamada quando o utilizador escolhe a opção *Alteração PIN* no menu.

  Pedirá a confirmação da acção ao utilizador e sendo esta afirmativa, o novo PIN. Caso contrário o PIN manter-se-á inalterado.

  Regista o sucedido através da função *salva_operacoes()*.

- Função
  “void menu_cria_conta()”

  Função chamada quando o funcionário escolhe a opção *Criar Conta* no menu.

  Será pedido o tipo de conta (funcionário ou cliente) e o PIN. O número de conta será automaticamente definido pelo programa. Sendo utilizado o número inteiro positivo mais pequeno possível que não esteja a ser utilizado.

  Utilizada a função *insere_lista_utilizadores()* para adicionar a conta, a função *procura_utilizadores()* para encontrar o local onde acrescentar a conta e *salva_operacoes()* para registar o acontecimento.

- Função
  “int menu_elimina_conta()”

  Função chamada quando o funcionário escolhe a opção *Eliminar Conta* no menu.

  Requer a inserção do número de conta a eliminar e a confirmação da acção.

  Faz uso da função *procura_utilizadores()* para encontrar a conta a eliminar, da função *levanta_quantia()* para retirar o dinheiro da conta antes de a eliminar e da função *salva_operacao()* para guardar o registo.

  O processo pode não ser efectuado quando uma das seguintes situações ocorre: O funcionário cancela a operação, a conta a apagar não existe na base de dados, existem menos de 3 contas na base de dados, o montante na conta é demasiado alto (superior a 200€) ou se a conta a apagar é a do funcionário que está a utilizar o programa.

- Função
  “void menu_lista_clientes()”

  Função chamada quando o funcionário escolhe a opção *Listar Clientes* no menu.

  Através de um sistema de ordenação e da função *imprime_clientes()* imprime todas as contas na base de dados por ordem decrescente da quantia de cada conta. No caso da quantia em duas ou mais contas serem iguais, então é apresentado por ordem crescente do número de conta.

  Tem a função *salva_operacao()* embutida para guardar o registo do sucedido.

- Função
  “void menu_lista_operacoes()”

  Função chamada quando o funcionário escolhe a opção *Listar Operações* no menu.

  Oferece duas opções ao funcionário, *Listar* ou *Apagar lista*.

  Caso o funcionário deseje ver a lista de operações efectuadas até ao momento, a função guardará o registo da operação com a função *salva_operacao()* e imprimirá todos os registos guardados através da função *imprime_operacoes()*.

  Por outro lado, se o funcionário seleccionar a opção *Apagar lista* será chamada a função *limpa_operacoes()* para apagar todos os registos e posteriormente a função *salva_operacoes().*

- Função
  “void menu_carrega_notas()”

  Função chamada quando o funcionário escolhe a opção *Alterar número de notas* no menu.

  Requer a introdução do número de notas de cada valor (50€, 20€, 10€ e 5€) a acrescentar à Caixa Bancário Automática.

- Função
  “void menu_bloqueia()”

  Função chamada quando o funcionário escolhe a opção *Bloquear Conta* no menu.

  Pede ao funcionário o número da conta que pretende bloquear e altera o seu nível para menos um (-1).

  Utiliza a função *procura_utilizadores()* para encontrar a conta a bloquear e a função *salva_operacao()* para guardar o registo do sucedido.

- Função
  “void menu_desbloqueia()”

  Função chamada quando o funcionário escolhe a opção *Desbloquear Conta* no menu.

  Pede ao funcionário o número da conta que pretende desbloquear e altera o seu nível para (0).

  Utiliza a função *procura_utilizadores()* para encontrar a conta a bloquear e a função *salva_operacao()* para guardar o registo do sucedido.

- Função
  “void menu_imprimir_conta()”

  Função chamada quando o funcionário escolha a opção *Imprimir Conta* no menu.

  Pede o número da conta a imprimir e apresenta os dados da conta, número de conta, PIN, quantia, e tipo de conta.

  Utiliza a função *procura_utilizadores()* para encontrar a conta a imprimir.

  Caso a conta a imprimir não exista, o processo termina sem imprimir qualquer dado.

- Função
  “void menu_sair()”

  Função chamada quando o funcionário escolhe a opção *Terminar a Execução* no menu.

  Uma vez confirmado pelo funcionário que deseja terminar a execução, serve-se da função *salva_operacao()* para registar a acção, utiliza *salva_info_ficheiro()* para guardar as informações num ficheiro, apaga a lista de utilizadores através de *destroi_lista_utilizadores()* e termina o programa.

- Função
  “Lista_utilizadores destroi_lista_utilizadores (Lista_utilizadores lista)”

  Função chamada em *menu_sair().*

  Apaga a lista de utilizadores e liberta a memória usada.

- Função
  “int lista_vazia(Lista_utilizadores lista)”

  Função auxiliar ao processo da função *destroi_lista_utilizadores().*

- Função
  “int conta_contas(Lista_utilizadores lista)”

  Função utilizada para calcular o número total de contas na caixa bancário automática.

- Função
  “void salva_operacoes(int n,double quantia,Lista_utilizadores origem, Lista_utilizadores destino)”

  Função que recebe parâmetros em grande parte das funções principais e regista o sucedido no ficheiro *operacoes.txt* para posteriormente ser imprimido pela função *imprime_operacoes()*.

- Função
  “void imprime_operacoes()”

  Função chamada em *menu_lista_operacoes().*

  Abre o ficheiro *operacoes.txt* onde são guardados os registos de cada operação através da função *salva_operacoes()* e imprime-o.

- Função
  “void imprime_clientes()”

  Função chamada dentro de *menu_lista_clientes().*

  Imprime as informações de todas as contas de utilizadores previamente ordenadas por valor.

- Função
  “void limpa_operacoes()”

  Função chamada dentro de *menu_lista_operacoes()* quando o utilizador deseja apagar os registos até ao momento.

  Limita-se a abrir o ficheiro *operacoes.txt* onde são guardados os registos e fechá-lo. Apagando assim os registos.

- Função
  “void salva_info_ficheiro()”

  Função utilizada dentro de *menu_sair().*

  Guarda nos ficheiros binários *database.bin* e *notas.bin* a informação adquirida até ao momento.

- Função
  “Lista utilizadores cria_lista_utilizadores (void)”

  Função que criará uma lista duplamente ligada que posteriormente armazenará as contas dos funcionários e clientes.

- Função
  “void insere_lista_utilizadores(Lista_utilizadores users, unsigned int numero_conta, char b[10],double c, short d)”

  Através dos parâmetros recebidos da função *menu_cria_conta()* insere a nova conta no programa.

- Função
  “void procura_utilizadores(Lista_utilizadores users, int num_conta, Lista_utilizadores *ant, Lista_utilizadores *actual)

  Função utilizada para procurar nas listas ligadas um número de conta, devolvendo esse número caso exista.

- Função
  “void carrega_info_ficheiro()”

  Função chamada na *main().*

  Carrega os dados do ficheiro *database.bin* para as listas. Caso a base de dados esteja vazia pede a criação das duas primeiras contas obrigatórias para o programa funcionar.

Diagrama de Caminhos