//Program PrBancoDJJLT;

Type //armazena os dados de cada cliente e funcionarios

TCliente = record

num\_conta : integer;

nome : string[25];

cpf : string[11];

data\_nascimento : string[10];

contato : string[14];

senha : string[8];

saldo : real;

end;

TFuncionario = record

login : string[10];

nome : string[25];

cpf : string[11];

senha : string[8];

end;

TListaFuncionario = array[1..100] of TFuncionario; // armazenamento quantidade de usuários

TListaCliente = array[1..100] of TCliente;

procedure inserirLinha;

begin

writeln('--------------------------------');

end;

procedure finalizarMenu;

begin

writeln('[ENTER] PARA FINALIZAR.');

readln();

end;

function setNumConta(i: integer): integer;

var

x : integer;

begin

x := 100100 + i;

writeln('NUMERO DA CONTA: ', x);

setNumConta := x; //retorno da funcao

end;

function getNumConta: integer;

var

x : integer;

begin

write('CONTA: ');

readln(x);

getNumConta := x; //retorno da funcao

end;

function getNome: string;

var

x : string;

begin

write('NOME: ');

readln(x);

getNome := x; //retorno da funcao

end;

function getCPF: string;

var

x : string;

begin

write('CPF: ');

readln(x);

getCPF := x; //retorno da funcao

end;

function getDataNascimento: string;

var

x : string;

begin

write('DATA DE NASCIMENTO: ');

readln(x);

getDataNascimento := x; //retorno da funcao

end;

function getContato: string;

var

x : string;

begin

write('CONTATO: ');

readln(x);

getContato := x; //retorno da funcao

end;

function getLogin: string;

var

x : string;

begin

write('LOGIN: ');

readln(x);

getLogin := x; //retorno da funcao

end;

function getSenha: string;

var

x : string;

begin

write('SENHA: ');

readln(x);

getSenha := x; //retorno da funcao

end;

function getValor: real;

var

x : real;

begin

write('VALOR: ');

readln(x);

getValor := x; //retorno da funcao

end;

function getIndexCPF(cliente: TListaCliente; qtde\_cliente: integer): integer;

var

i, indexCPF : integer;

cpf : string;

begin

clrscr;

writeln('BUSCAR CLIENTE POR CPF');

cpf := getCPF; // informando o CPF que estou procurando.

clrscr;

indexCPF := 0;

for i := 1 to qtde\_cliente do

begin

if (cliente[i].cpf = cpf) then

begin

indexCPF := i;

break;

end;

end;

getIndexCPF := indexCPF; //retorno da funcao

end;

function getIndexConta(cliente: TListaCliente; qtde\_cliente: integer): integer;

var

i, indexConta : integer;

conta : integer;

begin

clrscr;

writeln('CLIENTE');

inserirLinha;

conta := getNumConta; // informando o numero de conta que estou procurando.

indexConta := 0;

for i := 1 to qtde\_cliente do

begin

if (cliente[i].num\_conta = conta) then

begin

indexConta := i;

break;

end;

end;

getIndexConta := indexConta; //retorno da funcao

end;

function getIndexLogin(funcionario: TListaFuncionario; qtde\_funcionario: integer): integer;

var

i, indexLogin : integer;

login : string;

begin

clrscr;

writeln('FUNCIONARIO');

inserirLinha;

login := getLogin; // informando o numero de conta que estou procurando.

indexlogin := 0;

for i := 1 to qtde\_funcionario do

begin

if (funcionario[i].login = login) then

begin

indexLogin := i;

break;

end;

end;

getIndexLogin := indexLogin; //retorno da funcao

end;

function autenticacaoSenhaCliente(cliente : TlistaCliente; qtde\_cliente : integer) : integer;

var

i : integer;

senha : string;

begin

textbackground(blue);

clrscr;

i := getIndexConta(cliente, qtde\_cliente);

senha := getSenha;

if (i = 0) or (cliente[i].senha <> senha) then

begin

inserirLinha;

writeln('SENHA OU CONTA INCORRETA');

finalizarMenu;

autenticacaoSenhaCliente := 0;

end

else

begin

inserirLinha;

writeln('ACESSO PERMITIDO');

finalizarMenu;

autenticacaoSenhaCliente := i;

end;

end;

function autenticacaoSenhaFuncionario(funcionario : TlistaFuncionario; qtde\_funcionario : integer) : boolean;

var

i : integer;

senha : string;

begin

textbackground(green);

clrscr;

i := getIndexLogin(funcionario, qtde\_funcionario);

senha := getSenha;

if (i = 0) or (funcionario[i].senha <> senha) then

begin

inserirLinha;

writeln('SENHA OU LOGIN INCORRETO');

finalizarMenu;

autenticacaoSenhaFuncionario := false;

end

else

begin

inserirLinha;

writeln('ACESSO PERMITIDO');

finalizarMenu;

autenticacaoSenhaFuncionario := true;

end;

end;

procedure getCliente(var cliente : TListaCliente; var i: integer); //i = qtde\_Cliente

begin

clrscr;

inc(i);

writeln('CADASTRO CONTA CLIENTE');

inserirLinha;

cliente[i].num\_conta := setNumConta(i);

cliente[i].nome := getNome;

cliente[i].cpf := getCPF;

cliente[i].data\_nascimento := getDataNascimento;

cliente[i].contato := getContato;

cliente[i].senha := getSenha;

inserirLinha;

writeln('CLIENTE CADASTRADO COM SUCESSO');

finalizarMenu;

end;

procedure getFuncionario(var funcionario : TListaFuncionario; var i: integer); //i = qtde\_Funcionario

begin

clrscr;

inc(i);

writeln('CADASTRO FUNCIONARIO');

inserirLinha;

funcionario[i].login := getLogin;

funcionario[i].nome := getNome;

funcionario[i].cpf := getCPF;

funcionario[i].senha := getSenha;

inserirLinha;

writeln('FUNCIONARIO CADASTRADO COM SUCESSO');

finalizarMenu;

end;

procedure setFuncionarioAdmin(var funcionario : TListaFuncionario; var i: integer);

begin

inc(i);

funcionario[i].login := 'admin';

funcionario[i].nome := 'admin';

funcionario[i].cpf := 'admin';

funcionario[i].senha := 'admin';

end;

procedure listarClientes(cliente: TListaCliente; qtde\_cliente : integer);

var

i : integer;

begin

for i := 1 to qtde\_cliente do

begin

writeln('CONTA: ', cliente[i].num\_conta, ' - NOME: ', cliente[i].nome);

end;

inserirLinha;

finalizarMenu;

end;

procedure consultarCliente(cliente: TListaCliente; qtde\_cliente: integer);

var

index : integer;

begin

clrscr;

index := getIndexConta(cliente, qtde\_cliente); // identificando atraves da conta \*\* criar metodo para consultar Cliente atraves do CPF

if (index <> 0) then

begin

writeln('DADOS CLIENTE');

inserirLinha;

writeln('CONTA: ', cliente[index].num\_conta);

writeln('NOME: ', cliente[index].nome);

writeln('CPF: ', cliente[index].cpf);

writeln('DATA NASCIMENTO: ', cliente[index].data\_nascimento);

writeln('CONTATO: ', cliente[index].contato);

writeln('SALDO: R$ ', cliente[index].saldo);

end

else

writeln('CONTA NAO ENCONTRADA');

inserirLinha;

finalizarMenu;

end;

procedure consultarSaldo(cliente : TListaCliente; i : integer);

begin

writeln('SALDO: R$ ', cliente[i].saldo :2:2);

finalizarMenu;

end;

procedure efetuarDeposito(var cliente : TListaCliente; i : integer);

var

deposito : real;

begin

clrscr;

writeln('DEPOSITO');

inserirLinha;

deposito := getValor;

cliente[i].saldo := cliente[i].saldo + deposito;

writeln('TRANSAÇÃO REALIZADA COM SUCESSO');

finalizarMenu;

end;

procedure efetuarSaque(var cliente : TListaCliente; i : integer);

var

saque, saldo : real;

begin

clrscr;

writeln('SAQUE');

inserirLinha;

saque := getValor;

saldo := cliente[i].saldo - saque;

if (saldo < 0) then

writeln('SALDO INSUFICIENTE')

else

begin

cliente[i].saldo := saldo;

writeln('TRANSAÇÃO REALIZADA COM SUCESSO');

end;

finalizarMenu;

end;

procedure efetuarTransferencia(var cliente : TListaCliente; i : integer; qtde\_cliente: integer);

var

j : integer;

transf, saldo : real;

Begin

j := getIndexConta(cliente, qtde\_cliente);

transf := getValor;

saldo := cliente[i].saldo - transf;

if (saldo < 0) then

writeln('SALDO INSUFICIENTE')

else if (j <> 0) then

begin

cliente[i].saldo := saldo;

cliente[j].saldo := cliente[j].saldo + transf;

writeln('TRANSAÇÃO REALIZADA COM SUCESSO');

end

else

writeln('CONTA PARA TRANSFERENCIA INEXISTENTE');

finalizarMenu;

End;

procedure menuFuncionario(var cliente : TListaCliente; var funcionario : TlistaFuncionario;

var qtde\_cliente: integer; var qtde\_funcionario: integer);

var

op : integer;

begin

repeat

clrscr;

writeln('ACESSO FUNCIONARIO');

inserirLinha;

writeln('1 - CADASTRAR CLIENTE');

writeln('2 - CADASTRAR FUNCIONARIO');

writeln('3 - LISTAR CLIENTES');

writeln('4 - CONSULTAR CLIENTE');

writeln('5 - VOLTAR');

inserirLinha;

write('>>> ');

readln(op);

case (op) of

1: getCliente(cliente, qtde\_cliente);

2: getFuncionario(funcionario, qtde\_funcionario);

3: listarClientes(cliente, qtde\_cliente);

4: consultarCliente(cliente, qtde\_cliente);

5: break;

end;

until (false);

end;

procedure menuCliente(var cliente : TlistaCliente; i : integer; qtde\_cliente : integer);

var

op : integer;

begin

repeat

clrscr;

writeln('ACESSO CLIENTE');

inserirlinha;

writeln('1 - CONSULTAR SALDO');

writeln('2 - DEPOSITO');

writeln('3 - SAQUE');

writeln('4 - TRANSFERENCIA');

writeln('5 - VOLTAR');

inserirLinha;

write('>>> ');

readln(op);

case (op) of

1: consultarSaldo(cliente, i);

2: efetuarDeposito(cliente, i);

3: efetuarSaque(cliente, i);

4: efetuarTransferencia(cliente, i, qtde\_cliente);

5: break;

end;

until (false);

end;

procedure menuInicial;

var

op, qtde\_cliente, qtde\_funcionario: integer;

cliente : TListaCliente;

funcionario : TListaFuncionario;

i : integer;

begin

i := 0;

qtde\_cliente := 0;

qtde\_funcionario := 0;

setFuncionarioAdmin(funcionario, qtde\_funcionario);

repeat

textbackground(red);

clrscr;

writeln('ESCOLHA UMA OPCAO:');

inserirLinha;

writeln('1 - ACESSO FUNCIONARIO');

writeln('2 - ACESSO CLIENTE');

writeln('3 - FINALIZAR');

inserirLinha;

write('>>> ');

readln(op);

case (op) of

1: while (autenticacaoSenhaFuncionario(funcionario, qtde\_funcionario)) do

begin

menuFuncionario(cliente, funcionario, qtde\_cliente, qtde\_funcionario);

break;

end;

2: begin

i := autenticacaoSenhaCliente(cliente, qtde\_cliente);

if (i <> 0) then

begin

menuCliente(cliente, i, qtde\_cliente);

end;

end;

3: break;

end;

until (false);

end;

procedure inicializar; //procediemento menu iniciar

begin

clrscr;

gotoxy(45,2);

inserirLinha;

gotoxy(47,3);

writeln('BEM VINDO AO BANCO DJJLT S.A.');

gotoxy(49,4);

writeln('ONDE OS JUROS NAO EXISTEM');

gotoxy(45,5);

inserirLinha;

gotoxy(52,6);

writeln('[ENTER] PARA ENTRAR');

readln();

end;

procedure finalizar;

begin

clrscr;

gotoxy(47,2);

writeln('AGRADECEMOS SUA PREFERENCIA!');

gotoxy(45,3);

inserirLinha;

gotoxy(50,4);

writeln('TRABALHO REALIZADO POR:');

gotoxy(43,5);

writeln('DENNIS, JOAO, JEFFERSON, LUAN, THIAGO');

gotoxy(45,6);

inserirLinha;

gotoxy(48,7);

finalizarMenu;

end;

//programa principal

Begin

textbackground(red);

textcolor(white);

inicializar;

menuInicial;

finalizar;

End.