DOCUMENTAÇÃO

O projeto Controle de despesas pessoais foi desenvolvido para ser simples e prático, é conveniente para os usuários controlarem suas despesas e garantir que os dados sejam persistidos durante a execução do programa. O projeto contém um sistema desenvolvido em linguagem C para auxiliar na gestão financeira pessoal, permitindo aos usuários cadastrar visualizar, editar e excluir despesas mensais, bem como realizar análises como gastos máximos e mínimos, despesas totais, filtragem por categoria e simulação de renda financeira. A interface é completamente baseada em terAminal (console) e usa operações de cursor (gotoxy).

Nº	**Funcionalidade**	**Descrição**
1	Adicionar despesa	Permite inserir uma nova despesa com categoria e valor
2	Listar despesas	Exibe todas as despesas cadastradas
3	Mostrar total gasto	Calcula e exibe a soma total das despesas
4	Excluir uma despesa	Remove uma despesa informando o índice
5	Editar uma despesa	Permite alterar a categoria e o valor de uma despesa
6	Filtrar por categoria	Exibe apenas despesas que pertencem a uma determinada categoria
7	Maior e menor despesa	Exibe o maior e o menor valor entre todas as despesas
8	Limpar todas as despesas	Apaga todos os dados registrados
9	Simular rendimento	Calcula rendimento com juros compostos sobre um valor investido

- **Bibliotecas Utilizadas:**

stdio.h

Funções de entrada e saída padrão (ex: printf, scanf).

stdlib.h

Funções utilitárias (ex: system, exit).

string.h

Funções para manipulação de strings (ex: strcpy, strcmp).

locale.h

Permite a configuração de localização (ex: setlocale para usar acentos).

math.h

Funções matemáticas (ex: pow para potenciação).

windows.h

Usado para funções específicas do Windows como gotoxy e system . A inclusão direta não está no código, mas é necessária para que SetConsoleCursorPosition e GetStdHandle funcionem.

- **Constantes e Estruturas:**

MAX

Define o número máximo de despesas que podem ser armazenadas (100)

TAM_CATEGORIA

Define o tamanho máximo da string para o nome da categoria (50 caracteres)

Despesa

Uma struct que representa uma única despesa, contendo sua categoria (uma string) e seu valor (um float)

-Variáveis:

despesas [MAX]

Um array do tipo Despesa que armazena todas as despesas cadastradas

totalDespesas:

Um contador que mantém o número atual de despesas no array. É fundamental para o controle do programa.

- **FUNÇÕES:**

main()

Configura o console, define a linguagem como português e chama o menu principal.

menu()

Exibe o menu visual com gotoxy(), lê a opção do usuário e chama a função correspondente.

adicionarDespesa()

Cadastra uma nova despesa pedindo a categoria e valor. Valida o valor e salva no arquivo.

listarDespesas()

Percorre o vetor despesas[] e imprime cada uma na tela.

mostrarTotalGasto()

Soma todos os valores de despesas[] e imprime o resultado total.

excluirDespesa()

Remove uma despesa deslocando os elementos do vetor para preencher o espaço.

editarDespesa()

Permite alterar a categoria e o valor de uma despesa já cadastrada.

filtrarPorCategoria()

Compara a categoria digitada com as armazenadas e mostra as que combinam.

maiorMenorDespesa()

Compara todos os valores para encontrar o maior e o menor.

limparDespesas()

Zera a variável totalDespesas, apagando todos os registros da memória.

simularRendimento()

Recebe um valor inicial, taxa de juros e tempo e simula juros compostos:

Fórmula: R = P * (1 + i)^t

salvarDespesas() / carregarDespesas()

Lê e escreve os dados do vetor despesas[] no arquivo DESPESAS.txt, garantindo persistência.

gotoxy(x, y)

Move o cursor para a posição (x, y) no console (apenas no Windows com windows.h).

desenharJanela()

Cria uma interface visual no console usando =, * e gotoxy().

limparBuffer()

Evita erros de leitura ao limpar o buffer do teclado após scanf()

- **FUNÇÕES E FUNCIONALIDADES:**

adicionarDespesa()

Pede ao usuário a categoria e o valor. Adiciona a nova despesa ao final do array despesas e incrementa totalDespesas. Valida se o valor é positivo

listarDespesas()

Percorre o array despesas de 0 até totalDespesas - 1 e imprime cada item formatado na tela.

mostrarTotalGasto()

Inicializa uma variável soma com 0. Percorre o array de despesas, somando o valor de cada uma, e exibe o resultado final.

excluirDespesa()

Pede o índice da despesa a ser removida. Move todos os elementos posteriores uma posição para trás no array, sobrescrevendo o item a ser excluído, e decrementa totalDespesas.

editarDespesa()

Pede o índice da despesa a ser editada. Em seguida, solicita a nova categoria e o novo valor, atualizando os dados diretamente no array.

filtrarPorCategoria()

Solicita uma categoria ao usuário. Percorre todo o array de despesas e imprime apenas aquelas cuja categoria corresponde (usando strcmp) à informada.

maiorMenorDespesa()

Inicializa as variáveis maior e menor com o valor da primeira despesa. Em seguida, percorre o restante do array, atualizando maior e menor sempre que encontra um valor maior ou menor, respectivamente.

limparDespesas()

Reseta o contador totalDespesas para 0. Isso efetivamente apaga todas as despesas da memória. A função salvarDespesas() é chamada para sincronizar o arquivo vazio.

simularRendimento()

Pede ao usuário um valor inicial, uma taxa de juros anual e um período em anos. Calcula o montante final usando a fórmula de juros compostos: $M=C\times(1+i)t$.

- **FUNÇÕES DE PERSISTÊNCIAS:**

salvarDespesas()

Abre o arquivo DESPESAS.txt no modo de escrita ("w"), que apaga o conteúdo anterior. Percorre o array despesas e escreve cada categoria e valor no arquivo, separados por um espaço e com uma nova linha ao final.

carregarDespesas()

Abre DESPESAS.txt no modo de leitura ("r"). Lê o arquivo linha por linha, extraindo a categoria e o valor e preenchendo o array despesas até o final do arquivo (EOF). Incrementa totalDespesas para cada despesa carregada.

- **FUNÇÕES AUXILIARES:**

desenharJanela()

Orquestra a criação da interface visual, chamando as funções linhaHorizontal, linhaVertical e textoJanela para desenhar as bordas e o cabeçalho.

gotoxy(int x, int y)

Move o cursor do console para uma coordenada específica (coluna x, linha y). É uma função essencial para construir a interface.

linhaHorizontal(...)

Desenha uma linha horizontal no console usando um caractere específico.

linhaVertical(...)

Desenha duas linhas verticais (esquerda e direita) no console.

textoJanela()

Posiciona e imprime os textos do cabeçalho da janela.

limparBuffer()

Limpa o buffer de entrada do teclado. É crucial para evitar que caracteres indesejados (\n) de uma leitura com scanf interfiram na próxima.

Fluxo de Execução e Persistência de Dados

1. **Inicialização:** O programa inicia e a função main chama menu().

2. **Carregamento:** menu() chama carregarDespesas(), que lê o arquivo DESPESAS.txt e popula o array despesas com os dados salvos anteriormente.

3. **Interação:** O usuário interage com o menu, e as funções manipulam os dados exclusivamente no array em memória (despesas e totalDespesas).

4. **Salvamento:** Funções que modificam os dados (como adicionarDespesa, excluirDespesa, editarDespesa e limparDespesas) chamam salvarDespesas() ao final de sua execução.

5. **Sincronização:** salvarDespesas() reescreve o arquivo DESPESAS.txt com o conteúdo atualizado do array em memória. Isso garante que, se o programa for fechado, os dados estarão salvos para a próxima vez.

6. **Encerramento:** Quando o usuário escolhe a opção "Sair", o loop do menu termina e o programa é encerrado.