

家計簿プログラム

VerC

```

C kakeibo.c  X  kakeibo.txt
C kakeibo.c
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 #define MAX_RECORDS 100
6 #define ITEM_LENGTH 50
7 #define INPUT_BUFFER_SIZE 256
8
9 // 家計簿の記録を格納する構造体
10 typedef struct {
11     char item[ITEM_LENGTH]; // 費目（例：食費、交通費）
12     int amount;             // 金額
13     char type;              // 収支のタイプ ('E':支出, 'I':収入)
14 } Record;
15
16 Record records[MAX_RECORDS];
17 int record_count = 0;
18
19 // --- ヘルパー関数 ---
20
21 // 1. 1行全体を安全に読み込む基本関数（バッファクリアも行う）
22 char* read_line(char *buffer, int size) {
23     char *result = fgets(buffer, size, stdin);
24     char *newline_pos;
25
26     if (result == NULL) {
27         return NULL; // EOFまたは読み込みエラー
28     }
29
30     // 改行文字 '\n' を探す
31     newline_pos = strchr(buffer, '\n');
32     if (newline_pos) {
33         *newline_pos = '\0'; // 見つかったらNULL終端に置き換える
34     } else {
35         // バッファ長を超えた入力を読み捨てる（タイムアウト防止の鍵）
36         int c;
37         while ((c = getchar()) != '\n' && c != EOF);
38     }
39     return buffer;
40 }
41
42 // 2. read_lineを使ってメニュー選択の入力を受け取り、数値に変換する関数
43 int get_menu_choice() {
44     char buffer[INPUT_BUFFER_SIZE];
45
46     // 安全に入力行を読み込む
47     if (read_line(buffer, sizeof(buffer)) == NULL) {
48         return -1; // 終了シグナル
49     }
50
51     // 文字列を整数に変換して返す
52     return atoi(buffer);
53 }
54
55 // --- メインロジック ---
56
57 void register_record(char type) {
58     char buffer[INPUT_BUFFER_SIZE];
59
60     if (record_count >= MAX_RECORDS) {
61         printf("登録件数が上限に達しました。\\n");
62         return;
63     }
64
65     // 費目（文字列）の入力 - fgetsで安全
66     printf("費目を入力してください: ");
67     if (read_line(records[record_count].item, ITEM_LENGTH) == NULL) return;
68     if (records[record_count].item[0] == '\\0') {
69         printf("費目が入力されなかったため、登録を中止します。\\n");
70         return;
71     }
72
73     // 金額（数値）の入力 - fgetsで読み込み、atoiで変換
74     printf("金額を入力してください: ");
75     if (read_line(buffer, sizeof(buffer)) == NULL) return;
76
77     records[record_count].amount = atoi(buffer);
78     if (records[record_count].amount <= 0) {
79         printf("無効な金額です。登録を中止します。\\n");
80         return;
81     }
82
83     // タイプとカウントを更新
84     records[record_count].type = type;
85     record_count++;
86
87     printf("%sとして%d円を登録しました。\\n",
88            (type == 'E' ? "支出" : "収入"),
89            records[record_count - 1].amount);
90 }
91
92 void display_summary() {
93     int total_income = 0;
94     int total_expense = 0;
95
96     if (record_count == 0) {
97         printf("登録されている記録はありません。\\n");
98         return;
99     }
100
101    printf("\\n--- 家計簿サマリー ---\\n");
102    for (int i = 0; i < record_count; i++) {
103        if (records[i].type == 'I') {
104            total_income += records[i].amount;
105        } else {
106            total_expense += records[i].amount;
107        }
108
109        printf("| %s | %-12s | %d円 |\\n",
110              (records[i].type == 'I' ? "収入" : "支出"),
111              records[i].item,
112              records[i].amount);
113    }
114    printf("-----\\n");
115    printf("総収入: %d円\\n", total_income);
116    printf("総支出: %d円\\n", total_expense);
117    printf("純利益: %d円\\n", total_income - total_expense);
118    printf("-----\\n");
119 }
120
121 int main() {
122     int choice;
123
124     printf("--- 簡易家計簿プログラム ---\\n");
125
126     while (1) {
127         printf("\\n--- メニュー ---\\n");
128         printf("1. 支出を登録\\n");
129         printf("2. 収入を登録\\n");
130         printf("3. サマリーディスプレイ\\n");
131         printf("4. 終了\\n");
132         printf("-----\\n");
133         printf("選択肢を入力してください: ");
134
135         // ★★★ すべての入力を安全な get_menu_choice() で処理 ★★★
136         choice = get_menu_choice();
137
138         if (choice == -1) {
139             printf("\\n入力の終端またはエラーによりプログラムを終了します。\\n");
140             break;
141         }
142     }
143 }

```

家計簿プログラム

家計簿を付けられるプログラムです。支出と収入を登録します。それぞれ費用名と費用額を入力します。純利益サマリーを見ることも出来ます。個人状態ですが、改良すると組織単位で使えるでしょう。

The screenshot shows a code editor with two tabs: 'kakeibo.c' and 'kakeibo.txt'. The 'kakeibo.c' tab displays C code for a budget application. The 'kakeibo.txt' tab shows a log file with entries for spending and income.

kakeibo.c

```
142
143     switch (choice) {
144         case 1:
145             register_record('E'); // 支出の登録
146             break;
147         case 2:
148             register_record('I'); // 収入の登録
149             break;
150         case 3:
151             display_summary();
152             break;
153         case 4:
154             printf("プログラムを終了します。\\n");
155             return 0;
156
157         default:
158             printf("無効な選択です。1から4の番号を入力してください。\\n");
159             break;
160     }
161
162     return 0;
163 }
```

kakeibo.txt

1	1
2	1500
3	食費
4	