

# 在庫管理プログラム

VERC

C zaikokannri.c ×

C zaikokannri.c > ...

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 #define MAX_ITEMS 100
6 #define NAME_LENGTH 50
7 #define INPUT_BUFFER_SIZE 256
8
9 // 商品情報を格納する構造体
10 typedef struct {
11     char name[NAME_LENGTH]; // 商品名
12     int quantity;           // 在庫数量
13     float price;            // 単価
14 } InventoryItem;
15
16 InventoryItem inventory[MAX_ITEMS];
17 int item_count = 0;
18
19 // --- 安定した入力処理のためのヘルパー関数 ---
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) return NULL;
27
28     // 改行文字 '\n' を探す
29     newline_pos = strchr(buffer, '\n');
30     if (newline_pos) {
31         *newline_pos = '\0'; // 見つかったらNULL終端に置き換える
32     } else {
33         // 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)
34         int c;
35         while ((c = getchar()) != '\n' && c != EOF);
36     }
37     return buffer;
38 }
39
40 // 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
41 int get_menu_choice() {
42     char buffer[INPUT_BUFFER_SIZE];
43
44     if (read_line(buffer, sizeof(buffer)) == NULL) {
45         return -1; // 終了シグナル
46     }
47
48     // 文字列を整数に変換して返す
```

```
49     return atoi(buffer);
50 }
51
52 // 3. read_lineを使って浮動小数点数の入力を受け取り、floatに変換する関数
53 float get_float_input() {
54     char buffer[INPUT_BUFFER_SIZE];
55     if (read_line(buffer, sizeof(buffer)) == NULL) return -1.0f;
56
57     // 文字列を浮動小数点数に変換して返す
58     return atof(buffer);
59 }
60 // --- メインロジック ---
61
62 void add_item() {
63     float input_price;
64     int input_quantity;
65
66     if (item_count >= MAX_ITEMS) {
67         printf("登録件数が上限に達しました。\\n");
68         return;
69     }
70
71     // 1. 商品名 (文字列) の入力 - fgetsで安全
72     printf("商品名を入力してください: ");
73     if (read_line(inventory[item_count].name, NAME_LENGTH) == NULL) return;
74     if (inventory[item_count].name[0] == '\\0') {
75         printf("商品名が入力されなかったため、登録を中止します。\\n");
76         return;
77     }
78
79     // 2. 数量 (整数) の入力 - fgets/atoiで安全
80     printf("数量を入力してください: ");
81     input_quantity = get_menu_choice(); // 整数なのでget_menu_choiceを流用
82     if (input_quantity <= 0) {
83         printf("数量が無効です。登録を中止します。\\n");
84         return;
85     }
86     inventory[item_count].quantity = input_quantity;
87
88     // 3. 価格 (浮動小数点数) の入力 - get_float_inputで安全
89     printf("単価を入力してください: ");
90     input_price = get_float_input();
91     if (input_price <= 0.0f) {
92         printf("単価が無効です。登録を中止します。\\n");
93         return;
94     }
95     inventory[item_count].price = input_price;
```

```
96
97     item_count++;
98     printf("商品「%s」を登録しました。\\n", inventory[item_count - 1].name);
99 }
100
101 void display_inventory() {
102     float total_value = 0.0f;
103
104     if (item_count == 0) {
105         printf("在庫に登録されている商品はありません。\\n");
106         return;
107     }
108
109     printf("\\n--- 在庫一覧 ---\\n");
110     printf("| No | 商品名 | 数量 | 単価 | 在庫額 |\\n");
111     printf("-----\\n");
112
113     for (int i = 0; i < item_count; i++) {
114         float item_value = inventory[i].quantity * inventory[i].price;
115         total_value += item_value;
116
117         printf("| %-2d | %-16s | %-4d | %-6.2f | %-6.2f |\\n",
118             i + 1,
119             inventory[i].name,
120             inventory[i].quantity,
121             inventory[i].price,
122             item_value);
123     }
124     printf("-----\\n");
125     printf("在庫合計金額: %.2f円\\n", total_value);
126     printf("-----\\n");
127 }
128
129 int main() {
130     int choice;
131
132     printf("--- 簡易在庫管理プログラム ---\\n");
133
134     while (1) {
135         printf("\\n--- メニュー ---\\n");
136         printf("1. 商品の新規登録\\n");
137         printf("2. 在庫一覧表示\\n");
138         printf("3. 終了\\n");
139         printf("-----\\n");
140         printf("選択肢を入力してください: ");
141
142         choice = get_menu_choice();
143         if (choice == 1) add_item();
144         if (choice == 2) display_inventory();
145         if (choice == 3) break;
146     }
```

# 在庫管理プログラム VERC

在庫を管理できるプログラムです。商品名、数量、単価を入力する必要があります。在庫合計金額の表示及び、在庫一覧表示が可能です。改良すればより多くの商品在庫を管理できるようになるでしょう。

必要に応じて追加すべき情報を加えた改良を施すことでより洗練された使いやすい代物になっていきます。

```
143 // ★★★ すべての入力を安全な get_menu_choice() で処理 ★★★
144 choice = get_menu_choice();
145
146 if (choice == -1) {
147     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");
148     break;
149 }
150
151 switch (choice) {
152     case 1:
153         add_item();
154         break;
155     case 2:
156         display_inventory();
157         break;
158     case 3:
159         printf("プログラムを終了します。\\n");
160         return 0;
161
162     default:
163         printf("無効な選択です。1から3の番号を入力してください。\\n");
164         break;
165 }
166
167 return 0;
168 }
```