

Análise de código - Simulador de loja utilizando ponteiros

Modularização:

No quesito de modularização, o código apresenta quase todas as funções pedidas explicitamente na descrição do programa (incluir, alterar, consultar e excluir), com a adição de uma função para listar todos os produtos. Porém, não possui a função de venda que foi pedida, não sendo possível realizar uma compra.

Ao analisar a função “main”, podemos perceber a entrada e saída de dados que é realizada dentro dela, algo que não deve ser feito. Logo, era necessário criar uma outra função para o I/O de dados, que por mais que não estivesse especificada nos requisitos, é algo implícito da programação.

Clean code:

Como já comentei, o código segue o paradigma da modularização, ou seja, nesse aspecto do clean code está bem formulado.

Em relação à nomenclatura de variáveis e funções, está algo confuso e desorganizado, apesar de possuírem nomes significativos. A maioria das variáveis não estão seguindo a forma camel case, com uma ou duas exceções. As funções estão em camel case, mas começam com letra maiúscula, algo que não seria a norma.

O programa utiliza uma variável diretiva, mas seu nome não descreve totalmente sua funcionalidade, sendo apenas “MAX” e não “MAX_PRODUTOS”, ou algo do tipo. Fora isso, na definição da variável “nomeproduto”, o tamanho máximo da string está sendo definido com um número e não utiliza a diretiva já existente, ou qualquer outra.

Por último, em relação aos comentários, o código os apresenta, mas são muito poucos e com poucas instruções, não deixando claro as funcionalidades para alguém se deparando com o código pela primeira vez.

Requisitos técnicos:

Sobre os demais requisitos técnicos (explícitos e implícitos), o programa utiliza “struct” como foi instruído, e faz a utilização de ponteiros para manipular a maioria dos dados.

Porém, é importante ressaltar que o uso dos ponteiros no código está bagunçado, principalmente na hora de passar dados para as funções, não sendo criada uma variável como ponteiro, mas sim passando o endereço de uma variável normal. No fim isso acaba não afetando o código, mas definir as variáveis colaboraria para o clean code.

Além disso, a variável “quantidade” nas funções de listar e consultar produtos não está sendo passada com ponteiros, assim ocupando mais espaço e não seguindo os parâmetros requeridos.

Incluir produto:

O programa inclui os produtos de maneira correta, mas permite a inserção de valores que não fazem sentido, como preço ou quantidade negativa. Além disso, o programa permite a entrada de letras e outros caracteres que não são números nesses mesmos valores, assim trazendo resultados inesperados e indesejados.

O programa também conta com um ID automático, que é útil inicialmente, mas após a exclusão de qualquer produto que não seja o último a ter sido cadastrado, esse sistema quebra e na hora da inclusão registra um ID duplicado.

Um ponto bom, é que o programa permite a inserção de uma leva de produtos de uma vez, tirando a necessidade de inserir um produto por vez e ter que retornar ao menu e selecionar a opção de inserir novamente caso queira cadastrar outro, mas mantendo a possibilidade de cadastrar apenas um produto agora e os demais depois.

Por último, o programa não pede confirmação de se você deseja mesmo cadastrar o produto, algo que evitaria o cadastro de um produto com erros no nome, preço ou quantidade. Isso não é um problema tão grande devido à funcionalidade de alterar, mas ainda seria algo útil.

Alterar produto:

A função de alterar está funcionando corretamente, fazendo a busca do produto a ser alterado por ID e avisando caso o ID não exista, mas também apresenta os mesmos erros citados na função de inclusão, podendo definir um preço ou valor negativo e não apresentando uma mensagem de erro ao digitar caracteres onde o input deveria ser um número, levando a resultados indesejados.

Essa função também não possui uma opção de confirmar a alteração que deseja fazer e, caso existam dois produtos com o mesmo ID, algo possível devido ao erro na função de inclusão, a alteração é sempre feita no mais antigo, sendo impossível alterar o mais novo.

Excluir produto:

A função de excluir apresenta o mesmo erro generalizado de todas as funções, onde não pede uma confirmação de que deseja mesmo tomar essa ação. Porém nesse caso é mais grave, pois a exclusão de um produto indevidamente pode acarretar os erros citados na hora da inclusão e, apesar de ser possível apenas incluí-lo novamente no caso de uma exclusão errônea, a total perda de um produto é mais grave que um erro de digitação que possa ter passado temporariamente.

Fora isso, a exclusão é feita corretamente pelo ID e avisa caso o ID informado não exista no estoque. No caso de dois produtos com o mesmo ID, a exclusão realizada é a do mais antigo, assim como na função de alterar.

Consultar produtos:

A consulta do produto, como a alteração e exclusão, é feita corretamente pelo ID e informa caso o ID informado seja inexistente. Também consulta apenas o produto mais antigo no caso de ID duplicado.

Algo importante de notar, é que ao utilizar a função de consulta o programa encerra. Isso ocorre por causa de um erro no menu de opções, reconhecendo a opção que encerra o programa como a mesma que consulta. Assim, logo após consultar ou não encontrar o ID informado o programa encerra abruptamente.

Listar produtos:

Apesar de não ser uma das funções essenciais, era algo que implicitamente deveria ter para auxiliar na user experience (UX). Ela lista todos os produtos corretamente, por ordem de ID, mas não prevê a lista vazia, apresentando apenas a lista sem nenhum produto e não uma mensagem de erro.

Menu:

O menu apresenta todas as opções disponíveis no programa de maneira correta e, apesar de esperar um número como input, não quebra totalmente com a entrada de um caractere, apenas age como a última opção escolhida, ou apresenta uma mensagem de erro se for a primeira entrada.

A opção “sair” não funciona, continuando o programa sem reportar nenhum feedback, e a opção de consultar encerra o programa logo após ser executada, como já foi citado anteriormente.

Conclusão:

No geral, o programa está funcionando de maneira aceitável, realiza quase todas as funções que foram pedidas, com exceção da compra, e apresenta os módulos necessários para isso. Possui alguns bugs e erros em diversas partes, com os mais notórios sendo:

- Programa encerra após uma consulta
- ID duplicado na hora de inclusão
- Entrada de caracteres onde números são esperados
- Entrada de valores negativos no preço ou quantidade do produto

No que diz respeito a visibilidade e integridade do código em si, é necessário revisar:

- Nomenclatura de variáveis
- Uso de diretivas
- I/O dentro da “main”
- Comentários mais instrutivos.

Disciplina: Programação Estruturada e Modular Prof. Carlos Veríssimo

Nome: Luis Felliipe Franco Arakaki 2ºADS - Tarde

28/11/2024 - São Paulo