Do zero ao C

Aprendendo Lógica de Programação de uma Forma Divertida

Guilherme Rosa de Moraes

Prefácio **8**

O que são algoritmos **11**

Vamos à pratica 13

Compilado ou interpretado? **16**

Compilado 17
Interpretado 18

Um pouco sobre a linguagem **20**

Comandos de entrada e saída **22**

Scanf 22
Printf 24

Escrevendo nosso primeiro código **27**

Variável **32**

Declaração de variáveis 33
Vamos praticar 34
Conclusão 36

Constante **38**

Tipo de dado **40**

Exercícios de fixação 42

Marcadores **44**

Caractere \n 49
Conclusão 51

Operadores aritméticos **52**

Vamos praticar 62
Exercícios de fixação 64

Operadores relacionais 66

Vamos praticar 71
Exercícios de fixação 74

Operadores lógicos 75

Vamos praticar 79
Conclusão 80
Exercício de fixação 81

Estruturas condicionais 83

Vamos praticar 88
Exercício de fixação 91

Estrutura de Caso - SWITCH CASE 93

Vamos praticar 94
Exercício de fixação 97

Estrutura de repetição 98

While 98
Vamos à prática 99
Do-While 101
Vamos praticar 102
For 104
Vamos praticar 107
Conclusão 109
Exercício de Fixação 110

Comando BREAK 113

Exercício de Fixação 116

Comando CONTINUE 119

Vamos praticar 119

Exercício de Fixação 123

O que são Vetores? 125

Vamos praticar(*) 128
Exercício de Fixação 132

O que são matrizes? 134

Exercício de Fixação 142

Dividir para conquistar 144

Passagem de parâmetros por valor e referência 150
Passagem por valor 151
Exercício de Fixação 158

Strings 161

Biblioteca String.h 167
Exercício de Fixação 175

Biblioteca Math.h 177

Conclusão 194
Exercícios de Fixação 196

Aponte para um lugar 198

Chegamos ao final da nossa aventura 203

Referências 205

Dedico este livro primeiramente a Deus por me dar forças para não desistir e aos meus professores e amigos por me ensinarem e aprenderem junto comigo

Prefácio

Programar parece ser algo complexo, ou melhor, um bicho de sete cabeças que nunca sai do nosso caminho. Assim como você, caro leitor, tive muita dificuldade em aprender programação pois não via utilidade em estudar muita teoria e não entender o verdadeiro sentido de programar.

Contudo, ao participar de maratonas de programação, dentro da universidade, me apaixonei por codar, aceitar novos desafios e principalmente solucionar problemas que pareciam impossíveis de serem resolvidos com apenas um lápis e uma folha de papel.

A partir dessa perspectiva, nosso objetivo é mostrar o quão divertido é programar e quantas coisas podemos tirar do papel após entender esse mundo divertido da programação.

Para que o seu aprendizado seja divertido e produtivo decidimos juntar a forma lúdica de ensino, a partir de exemplos do dia a dia, com conceitos da própria linguagem que servirão de base para a sua evolução no mundo da programação.

Dessa forma partiremos dos conceitos preliminares onde entenderemos a importância da programação em nossas atividades

até chegarmos ao ponto ideal de conhecimento para prosseguirmos com as tecnologias fornecidas no mercado.

Um ponto muito importante em nosso processo de aprendizagem é seguir o passo-a-passo indicado pelo livro, caso esteja iniciando no mundo da programação, pois muitas vezes pular etapas faz com que surjam muitas dúvidas em algoritmos complexos na qual lhe fará desistir.

Dessa forma busque entender cada ponto apresentado para que no final tudo que aprendeu tenha dado uma base sólida para que consiga prosseguir no novo que o mundo que a programação nos proporciona.

Visando uma maior praticidade, cada capítulo do livro estará dividido em três pilares: conceitual, situação problema e exercícios de fixação. Contudo, capítulos com foco introdutório não serão fornecidos exercícios.

Na parte conceitual será detalhado de maneira didática o conceito do assunto que aprenderemos, já na situação problema utilizaremos situações do nosso dia a dia para demonstrar onde esse conceito se aplica facilitando ainda mais o nosso entendimento.

Como é através da prática que aprendemos, no último pilar, serão fornecidos exercícios para que você possa verificar se de fato o

conceito foi entendido e assim poderá prosseguir para a próxima etapa.

Para ajudá-lo no entendimento dos problemas, deixarei ao final deste capítulo um link de acesso do google drive com as soluções dos problemas existentes nessa jornada. Entretanto, peço que tente de todas as formas resolver os exercícios e somente em caso de dúvidas acesse o link com as soluções.

Agora que você já sabe como iremos trabalhar te convido para entrar comigo nessa aventura e assim aprender a programar de forma divertida e eficaz, pois é através da prática e da criatividade que nos motivamos a aprender cada vez mais.

Link dos Exercícios resolvidos:
https://drive.google.com/drive/folders/1fbx9mvnkb2NiAj_oFKoXfAKDOfNPcug0?usp=sharing

O que são algoritmos

Agora que conhecemos a metodologia de ensino e entramos em nosso primeiro capítulo vamos voltar bem lá no início da história com os homens das cavernas. Parece até estranho utilizarmos História para aprender programação, mas posso dizer que é uma forma bem prática de entendermos algoritmos.

Para a sua sobrevivência, o homem das cavernas tinha uma rotina, pois ele precisava se alimentar e se proteger dos eventos existentes na floresta, sendo esses um período chuvoso, a chegada da noite e até mesmo o ataque de animais perigosos.



Se analisarmos um ponto chave desse cenário, a alimentação, percebemos que este homem precisava executar determinadas tarefas para se alimentar, sendo elas desenvolver o seu ferramental de caça,

ir à caça, batalhar contra a sua presa, produzir fogo, cozinhar o seu alimento e assim se alimentar.

A partir dessa perspectiva e dos passos executados pelos homens das cavernas podemos entender o que um algoritmo nada mais é do que uma sequência de passos para a solução de um problema, sendo esses passos desenvolvidos em nossas mentes ou passados ao computador a partir de linhas de código para resolução de um problema.

Agora você pode estar se questionando e percebendo que a todo momento desenvolvemos algoritmos para realizar uma tarefa, seja para ir à escola, colocar uma lâmpada, preparar um almoço e até mesmo para escrever um livro. Muito interessante não é.

E para te deixar mais encantado com esse conceito, posso lhe dizer que todos nós somos professores de nossos computadores, pois os computadores foram desenvolvidos para executarem tarefas, sendo estas, até o exato momento na qual escrevo este livro, criadas pelo próprio ser humano.

Dessa forma, para que o computador realize uma simples soma entre dois números é preciso que o ensinamos que para somar dois números é preciso resgatar os dois números e adicionar o valor do segundo ao primeiro.

Agora que entendemos o que entendemos de forma conceitual o que são algoritmos te convido a resolver um problema que enfrentamos em nosso dia a dia, sendo ele um evento chave que te levará, indiretamente, ao entendimento prévio de novos conceitos que veremos futuramente.

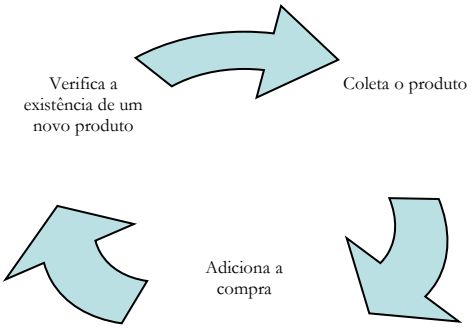
Vamos à pratica

Agora que entendemos o que são algoritmos, desenvolveremos um simples algoritmo que realiza o processo de venda em um supermercado pelo operador de caixa. Em primeiro momento temos que entender como funciona o fluxo de venda em um caixa de um supermercado. O primeiro passo a ser executado é coletar o primeiro produto da esteira do caixa.



Após coletar o primeiro produto devemos verificar se o mesmo tem o código de barras e colocar na leitora, que irá ler o código do produto, realizar os devidos cálculos, e assim adicionar o mesmo a lista de compras do cliente.

Agora que já coletamos o primeiro produto e o contabilizamos na compra, devemos verificar a existência de um novo produto. Caso ainda existam outros produtos executaremos o passo anterior até os produtos se encerrarem.



Ao verificar que não existe nenhum produto devemos fechar a venda, apresentar o valor da compra e esperar a forma de pagamento.

Após o pagamento podemos entregar a nota fiscal e atender o

próximo cliente. Lembrando que esse processo poderia ter mais passos, pois o cliente poderia não ter dinheiro suficiente e teríamos que cancelar a compra ou remover algum produto, poderíamos solicitar o CPF na nota fiscal entre outros passos.

Dessa forma desenvolvemos o nosso primeiro algoritmo a partir de um cenário vivenciado em cotidiano. Percebemos também que o nosso algoritmo pode variar de pessoa para pessoa onde podemos colocar ou não novos passos a serem executados.

Compilado ou interpretado?

Como podemos ver no capítulo anterior um algorítmo é uma sequência de passos para a resolução de um problema. Assim como a gente o computador possui uma forma de se comunicar que é a partir de números binários, os famosos zeros e uns.

A partir dessa perspectiva gerou-se uma grande dúvida, se somos professores de nossos próprios computadores mas utilizamos uma linguagem de comunicação diferente, uma linguagem de programação, como é que ele aprende e executa a tarefa que lhe foi passada.

Para que o nosso computador consiga entender o algorítmo que desenvolvemos através de uma linguagem de programação é necessária que seja feita uma conversão para linguagem de máquina, ou seja, linguagem binária, dessa forma o nosso computador consegue compreender e executar todos os passos necessários, para isso existe o processo de compilação e interpretação.

Compilado

Não precisamos entender como os compiladores fazem a tradução, basta sabermos como uma determinada linguagem de programação funciona (sintaxe e semântica) que o compilador traduzirá para uma linguagem que possa ser executada.

Os compiladores são tradutores que permitem com que um algoritmo seja transformado e assim ser compreendido e executado pelo computador. Dessa forma o compilador analisa o programa fonte, ou seja, o algoritmo que criamos, e traduz para uma linguagem de máquina) que será executado pelo computador.



Uma das principais vantagens de utilizar o processo de compilação em C é que ele permite transformar o código fonte em um arquivo executável que pode ser executado diretamente no sistema operacional, sem a necessidade de ter o compilador disponível em tempo de execução.

Isso significa que o código compilado pode ser executado em diferentes plataformas e sistemas operacionais sem a necessidade de fazer alterações no código-fonte.

Outra vantagem é que a compilação permite a detecção de erros no código antes da execução, pois o compilador verifica a sintaxe e a semântica do código, o que pode ajudar a identificar e corrigir erros de programação com mais facilidade e eficiência. Isso pode resultar em um código mais robusto e confiável.

Interpretado

Diferente do compilador, o interpretador traduz o nosso código linha por linha à medida em que ele vai sendo executado. Desta forma não há tradução de código, conversão de linguagem de máquina, mas sim interpretação do código fonte e de sua respectiva entrada, sendo um processo bem mais lento e que ocupará mais espaço em disco.



Uma das principais vantagens do processo de interpretação é que ele permite executar o código-fonte diretamente, sem a necessidade de compilar previamente. Isso significa que o código pode ser facilmente modificado e testado em tempo real, sem a necessidade de passar por todo o processo de compilação novamente.

A interpretação pode ser uma opção mais rápida para pequenos projetos ou projetos que não exijam alto desempenho, pois não é necessário passar pelo processo de compilação, o que pode economizar tempo e recursos de máquina, um ponto bem delicado quando trabalhamos com computadores que não possuem muitos recursos.

Um pouco sobre a linguagem

A linguagem C é uma linguagem de programação de computadores criada nos anos 70 por Dennis Ritchie, e é uma das linguagens mais utilizadas na atualidade. Ela é conhecida por ser uma linguagem de programação de baixo nível, o que significa que ela está mais próxima da linguagem de máquina do que outras linguagens mais modernas.

Considerada muito popular por sua rapidez e eficiência, permitindo assim aos programadores escreverem códigos que podem ser compilados e executados diretamente pelo hardware do computador. Ela é frequentemente usada para criar sistemas operacionais, drivers de dispositivos, softwares embutidos, jogos, entre outras aplicações.

Sua sintaxe é relativamente simples e consiste em uma série de instruções que são escritas em linhas separadas. A linguagem usa palavras-chave como "if", "else" e "while" para criar fluxos de controle, bem como operadores matemáticos, como "+" e "-", para realizar cálculos.