

Introdução ao Go

Aula 15: Compreendendo a Importância das Variáveis

Nesta aula, o professor explicou a diferença fundamental entre **constantes** e **variáveis** e mostrou como obter entrada do usuário no terminal utilizando a função `Scan` da biblioteca `fmt`. Vamos detalhar os pontos principais.

Diferença entre Variáveis e Constantes

Embora tanto variáveis quanto constantes sejam "contêineres" que armazenam valores, elas possuem comportamentos distintos:

- **Constantes:**
 - Declaradas com a palavra-chave `const`.
 - Seu valor **não pode ser alterado** após a definição.
 - **Devem ter um valor inicial atribuído** no momento da declaração.
 - São úteis para valores fixos, como a taxa de inflação em um programa financeiro.
- **Variáveis:**
 - Declaradas com a palavra-chave `var` ou com a notação curta `:=`.
 - Podem ter seu valor alterado ao longo da execução do programa.
 - Podem ser declaradas sem um valor inicial (nesse caso, recebem o valor padrão para o tipo).

Exemplo:

```
go
```

```
const inflationRate = 2.5 // Constante: valor fixo
var investmentAmount float64 // Variável: valor pode mudar

investmentAmount = 1000 // Atribuição inicial
investmentAmount = 2000 // Valor alterado posteriormente
```

Obtendo Entrada do Usuário com `Scan`

Go permite que você obtenha valores do terminal de maneira simples, usando a função `Scan` do pacote `fmt`. Aqui está como funciona:

1. Função `Scan`:

- A função `Scan` é usada para **ler entrada do usuário** e preencher uma variável.
- Requer que você passe um **ponteiro** para a variável que será preenchida.

2. Uso de Ponteiros:

- Um **ponteiro** indica o local na memória onde o valor da variável está armazenado.
- Para passar um ponteiro em Go, você utiliza o símbolo `&` (ampersand) antes da variável.

Exemplo:

```
go

var investmentAmount float64

fmt.Print("Enter investment amount: ")
fmt.Scan(&investmentAmount) // Preenche a variável com o valor digitado pelo usuário
```

Neste exemplo:

- O programa exibe a mensagem "Enter investment amount: ".
- O usuário digita um valor (ex.: 2000) e pressiona **Enter**.
- A variável `investmentAmount` é preenchida com o valor digitado.

Removendo o Valor Inicial de uma Variável

Se a variável será preenchida por uma entrada do usuário, você pode declará-la sem atribuir um valor inicial:

```
go

var investmentAmount float64 // Sem valor inicial
fmt.Scan(&investmentAmount) // Valor será definido pelo usuário
```

Isso não seria possível com uma constante, pois constantes exigem um valor inicial.

Fluxo do Programa com Entrada do Usuário

1. Declare as variáveis necessárias (ex.: `investmentAmount`, `years`, `expectedReturnRate`).
2. Use `fmt.Scan` para obter os valores do usuário:
 - Pergunte pelo **valor do investimento**.
 - Pergunte pelo **número de anos**.
 - Pergunte pela **taxa de retorno esperada**.
3. Realize os cálculos com base nos valores inseridos.

Exemplo Completo:

```
go

package main

import (
    "fmt"
    "math"
)

func main() {
    var investmentAmount float64
    var years int
    var expectedReturnRate float64
    const inflationRate = 2.5

    // Obter valores do usuário
    fmt.Print("Enter investment amount: ")
    fmt.Scan(&investmentAmount)

    fmt.Print("Enter number of years: ")
    fmt.Scan(&years)

    fmt.Print("Enter expected return rate (%): ")
    fmt.Scan(&expectedReturnRate)
```

```

// Calcular valor futuro
futureValue := investmentAmount * math.Pow(1+(expectedReturnRate/100),
float64(years))

// Calcular valor ajustado pela inflação
futureRealValue := futureValue / math.Pow(1+(inflationRate/100), float64(years))

// Exibir resultados
fmt.Printf("Future Value: %.2f\n", futureValue)
fmt.Printf("Inflation Adjusted Value: %.2f\n", futureRealValue)
}

```

Como funciona:

1. O programa solicita que o usuário insira o valor do investimento, o número de anos e a taxa de retorno esperada.
2. Ele calcula o **valor futuro** e o **valor ajustado pela inflação**.
3. Exibe os dois valores no terminal.

Resumo da Aula

1. **Variáveis** podem ser reatribuídas e permitem que valores mudem durante a execução do programa.
2. **Constantes** possuem valores fixos e não podem ser alteradas após a definição.
3. A função `fmt.Scan` é usada para obter entrada do usuário, e é necessário passar um **ponteiro** para a variável.
4. Você pode declarar variáveis sem valores iniciais e preenchê-las posteriormente com entradas do usuário.
5. Esse conceito é fundamental para tornar programas interativos e úteis em diversos cenários.

Se precisar de mais exemplos ou explicações detalhadas, é só pedir!