

Introdução ao Go

Aula 14: Fazendo Sentido dos Valores Constantes ("Constants")

Nesta aula, o professor introduziu o conceito de **valores constantes** e como eles podem ser usados para considerar a inflação ao calcular o valor futuro de um investimento. Vamos entender como isso funciona.

O Que São Constantes em Go?

Em Go, uma **constante** é um valor que **não pode ser alterado** após a sua definição. Para declarar uma constante, utilizamos a palavra-chave `const` em vez de `var`. O valor atribuído a uma constante não pode ser modificado em tempo de execução, o que é útil para valores que são fixos e não devem mudar.

Exemplo de Declaração de Constante:

```
go
```

```
const inflationRate = 2.5
```

Neste exemplo, `inflationRate` é uma constante que armazena o valor de 2.5% ao ano. A constante é útil para valores como taxas de inflação, onde o valor não mudará durante a execução do programa.

Considerando a Inflação no Cálculo do Valor Futuro

Na aula, o professor usou um cálculo de **valor futuro** considerando a **inflação** para ilustrar o uso das constantes. Aqui está um resumo de como isso foi feito:

1. Fórmula de Ajuste pela Inflação:

O valor futuro real (ajustado pela inflação) é calculado pela fórmula:

```
makefile
```

```
futureRealValue = futureValue / (1 + inflationRate / 100) ^ years
```

- `futureValue` é o valor futuro do investimento (sem considerar a inflação).
- `inflationRate` é a taxa de inflação anual (em percentagem).
- `years` é o número de anos de investimento.

2. Passos do Cálculo:

- A constante `inflationRate` é definida com o valor 2.5 (representando uma inflação de 2.5% ao ano).
- A fórmula é ajustada utilizando a função `math.Pow()` para calcular a inflação ao longo dos anos.

3. Exemplo de Código para Ajuste pela Inflação:

O código de Go para calcular o valor ajustado pela inflação seria algo assim:

```
go

const inflationRate = 2.5

futureRealValue := futureValue / math.Pow(1 + inflationRate/100, float64(years))
```

Nesse código:

- `futureValue` é o valor calculado sem a inflação.
- `inflationRate / 100` converte a taxa de inflação em um número decimal.
- `math.Pow(base, exp)` calcula a potência de `(1 + inflação)^anos`, considerando o efeito acumulado da inflação.

4. Impressão dos Resultados:

O valor futuro ajustado pela inflação é impresso da seguinte forma:

```
go

fmt.Println("Future Real Value: ", futureRealValue)
```

Assim, o programa exibe tanto o valor futuro sem inflação quanto o valor ajustado pela inflação.

1. O usuário insere o valor de investimento, o número de anos e a taxa de retorno esperada.
 2. A constante de inflação é usada para ajustar o valor futuro, e o programa exibe o valor ajustado.
 3. Se a taxa de inflação for maior que a taxa de retorno, o valor ajustado será menor que o valor inicial.
-

Exemplo Final

Suponha que a taxa de inflação seja 6.5% e o retorno esperado seja 5%. Se o valor inicial do investimento for 1000 e o número de anos for 10, o valor ajustado pela inflação seria calculado e exibido. Se a inflação for maior que o retorno, o valor ajustado pela inflação será inferior ao valor inicial do investimento, mostrando a perda de poder de compra ao longo do tempo.

Resumo da Aula

- **Constantes em Go** são declaradas com a palavra-chave `const` e têm valores imutáveis durante a execução do programa.
 - As constantes são úteis para armazenar valores fixos, como taxas de inflação, que não mudam ao longo do tempo.
 - Para calcular valores ajustados pela inflação, você pode usar a função `math.Pow()` em Go para aplicar a fórmula de ajuste acumulado ao longo dos anos.
 - O valor ajustado pela inflação pode ser comparado com o valor futuro não ajustado para analisar o impacto da inflação sobre os investimentos.
-

Se precisar de mais detalhes sobre como implementar ou entender algum ponto, fique à vontade para perguntar!