

Introdução à Programação #15

Constantes

Em C, tal como em outras linguagens de programação, existem as **constantes**. Constantes são, à semelhança das variáveis, algo em que podemos armazenar dados mas que, ao contrário das variáveis, não podemos alterar os dados que lá estão gravados.

Existem diversas formas de declarar constantes em C. Iremos abordar duas que são as mais utilizadas: as constantes declaradas e as constantes definidas.

Constantes Definidas (*#define*)

Chamam-se Constantes Definidas àquelas que são declaradas no cabeçalho de um ficheiro. Estas são interpretadas pelo pré-processador que procederá à substituição da constante em todo o código pelo respetivo valor.

Sintaxe

```
1. #define identificador valor
```

Onde:

- **identificador** → o nome da constante que, convencionalmente, é escrito com letras maiúsculas;
- **valor** → o valor da constante.

Exemplo

```
1. #include <stdio.h>
2.
3. #define PI 3.14159
4.
5. int main (){
6.
7.     double r = 5.0;
8.     double circle;
9.
10.    circle = 2 * PI * r;
11.    printf("%f\n", circle);
12.    return 0;
13.
14. }
```

O que acontece aqui é que, quando o pré-compilador lê a definição da constante “PI”, substitui todos os “PI” no código pelo valor que lhe está atribuído, literalmente.

Constantes Declaradas (*const*)

As Constantes Declaradas, ao contrário das Constantes Definidas são, tal como o próprio nome indica, declaradas no código, em linguagem C.

A declaração destas constantes é extremamente semelhante à declaração das variáveis. Apenas temos que substituir escrever *const* antes do tipo de dados.

Sintaxe

```
1. const tipo nome = valor;
```

Onde:

- **tipo** → o tipo de dados que a constante vai conter;
- **nome** → o nome da constante;
- **valor** → o conteúdo da constante.

Exemplo

```
1. #include <stdio.h>
2. #include <math.h>
3.
4. int main() {
5.     const double goldenRatio = (1 + sqrt(5)) / 2;
6.
7.     goldenRatio = 9; //Erro. A constante não pode ser alterada.
8.
9.     double zero = (goldenRatio * goldenRatio) - goldenRatio - 1;
10.    printf("%f", zero);
11.
12.    return 0;
13. }
```

Existem vantagens ao utilizar esta forma de declarar constantes. Através das constantes declaradas podemos ter constantes locais ou globais porém as definidas são sempre globais.

Uma constante local é uma constante que está restrita a uma determinada função, por exemplo. Se declararmos uma constante dentro de uma função, esta só estará disponível para o código que está dentro dessa mesma função.

<http://pplware.sapo.pt/tutoriais/vamos-programar-introducao-a-programacao-15/>