

Há nomes por todos os lados em um software. Nomeamos nossas variáveis, funções, parâmetros, classes, pacotes e diretórios. Como fazemos muito isso, é melhor fazer bem. A seguir estão algumas regras simples para a criação de bons nomes.

Use nomes que revelem seu propósito

Escolher bons nomes leva tempo, mas economiza mais, devemos cuidar dos nomes e trocá-los quando encontrar melhores. Todos que lerem o código, incluindo você mesmo, ficarão agradecidos.

Exemplo:

```
int d; // tempo decorrido em dias
```

```
int elapseTimeInDays;
```

O primeiro não revela nada, não indica ideia de tempo corrido, ou de dias. O segundo, especifica seu uso para mensuração e a unidade usada.

Evite informações erradas

Devemos evitar passar dicas falsas que podem confundir o sentido do código e o programador. Um simples exemplo seria criar uma variável de nome *“hp”*, pois ela pode significar mais de uma coisa.

Devemos também ter cuidado para os nomes de fato representarem o que são. Por exemplo, se tem um grupo de contas com nome *“accountList”*, mas ela não for uma *“list”* de verdade, seria melhor trocarmos o nome para apenas *“account”*.

Faça distinções significativas

Muitas vezes criamos problemas para nós mesmos. Por exemplo, você não pode usar o nome de uma variável então irá alterá-la de forma arbitrária, como *“class”* virar *“klass”*.

Palavras vagas e comuns também não são uma boa escolha. Digamos que tenha uma classe *“ProductData”*. Se houver outra chamada com nome de *“ProductInfo”*, você terá usado nomes distintos, mas que não revelam nada diferente, pois ambos são semelhantes. Você não saberia qual usar.

Evite o mapeamento mental

Não se deve deixar que os leitores tentem traduzir os nomes que você escolheu. Você criou uma variável de nome *“r”*, sabendo que era para se referir ao termo URL. Você sendo um programador inteligente, certamente irá se lembrar que fez isso.

Mas melhor que um programador inteligente, é um programador profissional que entende que clareza é fundamental, sendo assim, nomeie o URL como *“url”*. Pode facilitar sua vida.

Nomes de classes

Classes e objetos devem ter nomes com substantivos(s), como *“Customer, WikiPage, Account”*. O nome da classe deve indicar sua responsabilidade ou o que ela representa. Exemplos: *“UserProfile, OrderProcessor, Invoice”*.

Termos como *“Manager, Processor, Handler”* são comuns, mas evite usá-los se puder substituí-los por algo mais claro e específico. Por exemplo, *“OrderManager”* poderia ser *“OrderService”* ou algo mais específico para sua função.

Nomes de métodos

Use verbos para descrever a ação que o método realiza. Isso torna o código mais próximo de uma linguagem natural e fácil de ler. Exemplos: ***calculateTotal***, ***sendEmailNotification***, ***saveOrder***.

Para métodos que obtêm ou definem valores de atributos, use o prefixo padrão ***get*** ou ***set***, como ***getBalance*** e ***setBalance***. Para métodos booleanos, prefira ***is*** ou ***has***, como ***isEnabled*** ou ***hasPermission***.