



## Crie e uso laços do tipo do/while

Uma outra opção de laço `while` seria o `do .. while`, que é bem parecido com o `while`. A grande diferença é que a condição é testada após o corpo do `::loop::` ser executado pelo menos uma vez:

```
int i = 1;
do { //executa ao menos 1 vez
    System.out.println(i);
    i++;
} while (i < 10); // se der true, volta e executa novamente.
```

[COPIAR CÓDIGO](#)

A condição do `do .. while` só é verificada no final de cada iteração e não no começo, como no `while`. Repare que ao final do bloco `do .. while` existe um ponto e vírgula. Esse é um detalhe que passa despercebido muitas vezes, mas que resulta em erro de compilação se omitido:

```
int i = 1;
do {
    System.out.println(i);
    i++;
} while (i < 10) // não compila, faltou o ;
```

[COPIAR CÓDIGO](#)

Assim como no `while`, caso tenhamos apenas uma linha, as chaves podem ser omitidas. Caso exista mais de uma linha dentro do `do .. while` e não existam chaves, teremos um erro de compilação:

```
int i = 0;
//compila normal
do
    System.out.println(i++);
while(i<10);

//erro, mais de uma linha dentro do do .. while
do
    System.out.print("o valor é: "); //erro
    System.out.println(i++);
while(i<10);
```

[COPIAR CÓDIGO](#)