

Testes com IF

Transcrição

Passaremos pelo nosso primeiro controle de fluxo, e testaremos a condicional `if` ! Para isso, criaremos uma classe denominada "TestaCondicional", com uma variável inteira `idade` , inicializada na mesma linha:

```
public class TestaCondicional {  
  
    public static void main(String[] args) {  
        System.out.println("testando condicionais");  
        int idade = 20;  
        if (idade >= 18) {  
            System.out.println("você tem mais de 18 anos");  
            System.out.println("seja bem vindo");  
        }  
    }  
}
```

Dica: pode-se usar "Ctrl + barra de espaço" após digitarmos "**sysout**" e apertarmos "Enter" para autocompletar o `System.out.println()`; também!

Feito isso, salvaremos, e com o lado direito do mouse acessaremos "Run As > Java Application". Obteremos o seguinte:

```
testando condicionais  
você tem mais de 18 anos  
seja bem vindo
```

Entre `if` e os parênteses que vêm a seguir, não é obrigatório ter espaço, mesmo que geralmente se use. No Java, o espaço, as teclas "TAB" e "Enter" não possuem papel fundamental. No Eclipse, ao acessarmos "Source > Format", o código é formatado de maneira correta.

Até aqui, nenhuma grande novidade. Neste caso, há duas instruções no bloco do `if` ... Existe algo proveniente do C no Java, em que as chaves não são necessárias, quando se quer apenas uma instrução na condicional. Isto é, se a linha referente ao texto "seja bem vindo" não existisse, poderíamos remover as chaves, deixando assim:

```
public class TestaCondicional {  
  
    public static void main(String[] args) {  
        System.out.println("testando condicionais");  
        int idade = 20;  
        if (idade >= 18)  
            System.out.println("você tem mais de 18 anos");  
        // System.out.println("seja bem vindo");  
    }  
}
```

O `System.out.println();` que não está comentado (não está com as duas barras antes) faz parte do caso em que o `if` é `true`, verdadeiro. Quando temos um `if` ou um `else` sem o uso das chaves, não é possível ter duas instruções, e sim apenas uma. Por isso, a boa prática implica em usarmos as chaves independentemente da quantidade de instruções existentes.

Isso facilita enxergarmos quem faz parte do quê, deixando menos margem para dúvidas e erros, mas isso vai da preferência de quem programa.

Para o `else`, alteraremos `idade` para que se receba `16`, e digitaremos:

```
public class TestaCondicional {

    public static void main(String[] args) {
        System.out.println("testando condicionais");
        int idade = 16;
        if (idade >= 18) {
            System.out.println("você tem mais de 18 anos");
            System.out.println("seja bem vindo");
        } else {
            System.out.println("infelizmente você não pode entrar");
        }
    }
}
```

Com isso, veremos a impressão de `infelizmente você não pode entrar` no Console. Para o caso da pessoa estar acompanhada, ela poder entrar, então acrescentaremos `int quantidadePessoas = 3;`, e um `if` após `else`. Em seguida, incluiremos outro `else` para o caso da pessoa ter menos de `18` e estar desacompanhada:

```
public static void main(String[] args) {
    System.out.println("testando condicionais");
    int idade = 16;
    int quantidadePessoas = 3;

    if (idade >= 18) {
        System.out.println("você tem mais de 18 anos");
        System.out.println("seja bem vindo");
    } else {
        if(quantidadePessoas >= 2) {
            System.out.println("você não tem 18, mas " + "pode entrar, pois está acompanhado");
        } else {
            System.out.println("infelizmente você não pode entrar");
        }
    }
}
```

Quando o código começa a se estender demais pela tela, dificultando a visualização integral, pode-se apertar "Enter", o que, no Eclipse, faz com que as *strings* sejam separadas por aspas e `+` automaticamente.

Salvando e rodando o código, obteremos:

você não tem `18`, mas pode entrar, pois está acompanhado

Outra dica: com duplo clique em qualquer uma das *views*, ela é maximizada. Fazemos o mesmo para minimizá-la. Isto pode facilitar nosso trabalho!