

### Aula 3

#### Especial True or False / Compila ou não?

a) ( **F** ) package br.com.senac; import java.util.\*; public class AreaDoQuadrado() { public static void main(String[] args) {int lado1=0; int lado2=0; Scanner sc = new Scanner(System.in); System.out.print("Lado 1 -> "); lado1=sc.nextInt(); System.out.print("Lado 2 -> "); lado2=sc.nextInt(); int resultado = lado1\*lado2; System.out.println("Resultado -> " + resultado; sc.close(); } }

**Sublinhamos o erro, ou seja, uma confusão entre nome de classe e chamada de método. Tirando isso, o código deve compilar sem problemas.**

b) ( **F** ) package br.com.senac; import java.util.\*; public class MediaSimples { public static void main(String[] args) { Scanner sc=new Scanner(System.in); System.out.print("Nota 1 -> "); double nota1=sc.nextInt(); System.out.print("Nota 2 -> "); double nota2=sc.nextInt(); ; int media = (nota1\*nota2)/2; System.out.println("Media -> " + media; sc.close(); } }

**A parte destacada revela o erro deste código. Chamamos o método do objeto sc na forma sc.nextInt() para guardar um valor inteiro em um double. Vai dar erro. Uma solução seria alterar o método para sc.nextDouble, já que, pelas regras do negócio, uma nota pode ter um valor fracionado. O passo seguinte é ajustar a variável média para o tipo double. Pronto. Código compilará.**

c) ( **V** ) package br.com.senac; import java.util.\*; public class FatiaDePizza { public static void main(String[] args) { int media=6, grande=9, gigante=12; Scanner sc = new Scanner(System.in); System.out.print("Nro de pizzas -> "); int qtdadePizza=sc.nextInt(); System.out.print ("Total de fatias -> " + (qtdadePizza\*media)+ " fatias de pizza media ou " + (qtdade\*grande)+ " fatias de pizza grande ou " + (qtdade\*gigante)+ " fatias de pizza gigante "); sc.close(); } }.

d) ( **F** ) package br.com.senac; import java.util.\*; public class ZeladorCortaGrama { public static void main (String[] args) { Scanner sc = new Scanner(System.in); System.out.print ("Vai chover? Sim ou Não?"); String escolha =sc.next(); String true="Sim"; String false="Não"; if (escolha.equals(true)) {System.out.println("Corta a grama");} if (escolha.equals(false)) {System.out.println("Não corta a grama");} else {System.out.print("Opção inválida");} sc.close(); } }

**A pegadinha aqui está no nome da variável. O Java não permite o uso de palavras reservadas para nomear atributos, métodos, classes entre outros. Se, ao contrário, guardássemos true e false como strings dentro das variáveis (String Sim= "true" e String Nao="false") não haveria erro de compilação porque o Java vai interpretar essas palavras como texto, sem problemas. Assim, cuidado na hora de criar as variáveis.**

e) ( **F** ) package br.com.senac; import java.util.\*; public class Compra { public static void main (String[] args) {System.out.printf("Produto escolhido %s", escolheProduto());} public static String escolheProduto() { Scanner x = new Scanner(System.in); System.out.print("Nro do produto-> "); int nro = x.nextInt(); if (nro ==0) {return "Meia";} if (nro==1){return "Sapato";} x.close(); } }.

***Aqui o compilador vai identificar problema para fechar o método de entrada do objeto x, pois o return interrompe o fluxo antes da invocação do método. Assim, se deslocarmos o trecho de código selecionado para a linha acima, antecedendo o if(nro==0), liberamos o objeto antes dos return e o compilador ficará feliz, sem identificar erros.***

### **For discussion**

a) Explique o erro de compilação (se é que existe) dos códigos informados acima e justifique qual correção seria mais indicada.

***Esta correção já foi realizada junto a cada código.***