



COLÉGIO DE SÃO JOSÉ RAMALHÃO

Teste 3 de Programação Orientada a Objetos

Duração: 50 min

11º B

Nome:

Data: 31 Janeiro 2019

Leia atentamente todas as perguntas antes de começar a responder. Responda dentro das caixas.

PARTE I

1. Um dos conceitos fundamentais da programação orientada a objectos é o encapsulamento. O que entende por encapsulamento? Dê um exemplo.

Encapsulamento é um conceito de programação cujo objectivo é esconder a forma como os dados de um programa são tratados internamente. Um exemplo seria declarar um método para alterar uma variável, em vez de a alterar directamente.

2. Qual a diferença entre uma “Classe” e um “Objecto”? Dê um exemplo.

Uma classe define apenas a estrutura de um objecto. Um objecto é uma concretização ou instância de uma classe que contém os seus próprios dados.

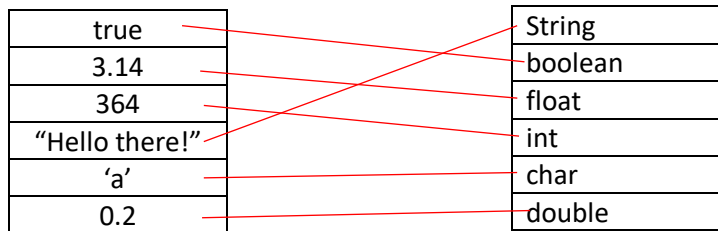
3. Qual é a diferença entre uma variável declarada “private” e uma variável declarada “public”?

Uma variável “private” só pode ser alterada pela própria classe onde foi declarada. Uma variável pública pode ser alterada directamente por outra classe.

PARTE II

1. Faça a associação entre os tipos de dados com os dados na seguinte tabela:

true	String
3.14	boolean
364	float
"Hello there!"	int
'a'	char
0.2	double



2. Escreva um método que recebe um número e verifica se esse número é ou não par. O método deve ter o seguinte comportamento:
 - a. `par(8) = true`
 - b. `par(747) = false`
 - c. `par(0) = true`

```
public boolean par(int n) {  
    return n%2==0;  
}
```

3. Escreva um método que recebe o valor de uma nota de uma avaliação e retorne a avaliação dessa nota. As notas assumem qualquer valor real entre 0 e 20 valores. Caso o método receba um valor inválido, deve avisar o utilizador. O método deve ter o seguinte comportamento:
 - a. `classificacao(5.2) = "Reprovado"`
 - b. `classificacao(18.3) = "Aprovado"`
 - c. `classificacao(-12) = "Nota inválida!"`
 - d. `classificacao(100) = "Nota inválida!"`

```
public String classificacao(double nota) {  
    if(nota<0 || nota>20)  
        return "Nota inválida!";  
    else {  
        if(nota<9.5)  
            return "Reprovado";  
        else  
            return "Aprovado";  
    }  
}
```

4. Considere o seguinte algoritmo:

```
public int algoritmo(int n) {  
    int resultado = 0;  
    for(int i = 0; i<=n; i++) {  
        resultado = resultado + i;  
    }  
    return resultado;  
}
```

Qual o resultado esperado das seguintes expressões?

Expressão	Resultado
algoritmo(3)	6
algoritmo(-2)	0
algoritmo(2.3)	ERRO
algoritmo(2+3)	15
algoritmo(0) + algoritmo(2)	3

PARTE III

1. Pretende-se criar uma classe que representa uma pessoa. Um pessoa deve ter um nome, idade e um número de identificação. Deve também ser capaz de ser apresentar, através de um método que devolva uma apresentação no formato de String. O programa deve comportar-se da seguinte forma:

Código:

```
public static void main(String[] args) {  
  
    Pessoa p1 = new Pessoa("Francisco", 29);  
    Pessoa p2 = new Pessoa("Pedro", 25);  
  
    System.out.println(p1.apresentar());  
    System.out.println(p2.apresentar());  
  
}
```

Resultado esperado:

```
Olá, sou o Francisco e tenho 29 anos!  
Olá, sou o Pedro e tenho 25 anos!
```

Defina a classe “Pessoa” para que o código acima devolva o resultado esperado.

```
class Pessoa {  
  
    private String nome;  
    private int idade;  
  
    public Pessoa(String nome, int idade) {  
        this.nome = nome;  
        this.idade = idade;  
    }  
  
    public String apresentar() {  
        return "Olá, sou o " + nome + " e tenho " + idade + " anos!";  
    }  
}
```

2. Leia com atenção o seguinte excerto de código.

```
class Animal {

    public String fazBarulho() {
        return "mmm";
    }

    public String senta() {
        return "Ignora...";
    }

}
```

```
class Gato extends Animal {

    public String fazBarulho()
    {
        return "Meow!";
    }

}
```

```
class Cao extends Animal {

    public String senta() {
        return "Sentado!";
    }

    public String fazBarulho() {
        return "Woof!";
    }

}
```


Qual é o resultado esperado para cada um dos seguintes conjuntos de instruções:

Nota: Caso o código dê qualquer tipo de erro, basta indicar “erro”.

Instruções	Resultado
Animal a1 = new Animal(); System.out.println(a1.fazBarulho());	mmm
Animal a2 = new Cao(); System.out.println(a2.senta());	Sentado!
Gato g1 = new Gato(); System.out.println(g1.fazBarulho());	Meow!
Cao c1 = new Animal(); System.out.println(c1.fazBarulho());	ERRO
Gato g2 = new Gato(); System.out.println(g2.senta());	Ignora...
Animal a3 = new Cao(); System.out.println(a3.senta());	Sentado!
Animal g3 = new Gato(); System.out.println(g3.festas());	ERRO

PARTE IV

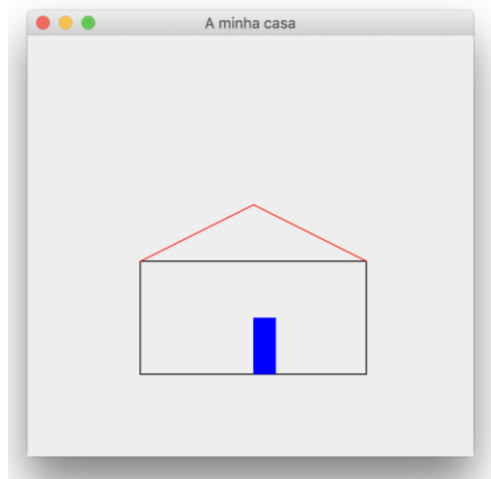
Considera a janela e listagem de código seguinte:

	<pre>1 package poo11B; 2 3 import java.awt.*; 4 5 import javax.swing.JFrame; 6 7 public class DesenhaCasa extends Canvas { 8 9 public void paint(Graphics g) 10 { 11 //define a cor 12 g.setColor(Color.black); 13 14 //desenha base 15 g.drawRect(100, 200, 200, 100); 16 17 //desenha porta 18 g.drawRect(200, 250, 20, 50); 19 20 //desenha telhado 21 g.drawLine(100, 200, 200, 150); 22 g.drawLine(x1, y1, x2, x2); 23 } 24 public static void main(String[] args) 25 { 26 JFrame janela = new JFrame("A minha casa"); 27 janela.setSize(800,600); 28 janela.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); 29 janela.add(new DesenhaCasa()); 30 janela.setVisible(true); 31 } 32 }</pre>
---	--

1. Qual o valor a preencher nas coordenadas x1, y1, x2, y2?

`g.drawLine(200, 150, 300, 200);`

2. Quais as alterações necessárias ao código para pintar a porta de azul e pintar o telhado de vermelho?



`g.setColor(Color.blue);`
`g.fillRect(200, 250, 20, 50);`

`g.setColor(Color.red);`
`//desenha telhado`