

**ATIVIDADE 2 - ENG SOFT - PROGRAMAÇÃO DE SISTEMAS II - 2018D3****Período:** 22/10/2018 08:00 a 06/11/2018 23:59 (Horário de Brasília)**Status:** ABERTO**Nota máxima:** 0,50**Gabarito:** Gabarito será liberado no dia 07/11/2018 00:00 (Horário de Brasília)**Nota obtida:****1ª QUESTÃO**

O controle do fluxo de execução de uma aplicação é muito importante, pois, por meio destes desvios, a aplicação se torna mais completa e capaz de lidar com situações esperadas ou não, podendo depender da interação de um usuário ou não (MANZANO, 2011). Tal controle de fluxo cria alternativas em tempo de execução, tornando a aplicação capaz de lidar com situações previstas como escolhas de valores para um atributo ou parâmetro de um método, ou imprevistas como o tratamento de erros.

MANZANO, J. A. N. G. **Java 7: programação de computadores: guia completo de introdução, orientação e desenvolvimento.** São Paulo: Érica, 2011.

Assinale a alternativa que contenha apenas instruções de controle de fluxo no java.

**ALTERNATIVAS**

- ☐ if, else, switch.
- ☐ if, elseif, while.
- ☐ case, for, gountil.
- ☐ switch, for. continue.
- ☐ foreach, while, gountil.

**2ª QUESTÃO**

Os métodos de classe são frequentemente utilizados em classes utilitárias. Um exemplo de classe utilitária é a classe java.util.Math em que é possível chamar métodos como: Math.random(), Math.min(1,2), Math.max(1,2), entre outros. Marque abaixo a assinatura de método que se refere a um método de classe em Java:

**ALTERNATIVAS**

- ☐ void validarCondicao(boolean condicao, String mensagem)
- ☐ public void validarCondicao (booean condicao, String mensagem)
- ☐ protected void validarCondicao (boolean condicao, String mensagem)
- ☐ private final void validarCondicao (boolean condicao, String mensagem)
- ☒ public static void validarCondicao (boolean condicao, String mensagem)

**3ª QUESTÃO**

Pequenos erros de programação podem consumir muito tempo até serem resolvidos. Felizmente, alguns desses erros são indicados pelo compilador quando se utiliza uma IDE. Veja um trecho de código Java que apresenta um problema:

```
10: do {  
11:   int i = 0;  
12:   System.out.println("Iteração " + i);  
13:   i++;  
14: } while ( i < 10 );
```

Marque a opção que indica corretamente o problema que há nesse código.

### ALTERNATIVAS

- ☐ O código compila, mas o laço repete infinitamente porque na linha 11 a variável i recebe 0 em todas as iterações.
- ☐ O código não compila, porque não se pode concatenar um tipo int a uma String, como feito na linha 12.
- ☐ O código não compila porque a instrução i++ da linha 13 não é válida em Java.
- ☐ O código não compila, porque a instrução do ... while não pode ter chaves nas linhas 10 e 14.
- ☒ O código não compila, porque a variável i da linha 11 foi definida dentro de um bloco e está fora de escopo na linha 14.

### 4ª QUESTÃO

Analisando o seguinte trecho código.

```
public class Pai {  
    public void metodo(Integer valor) {  
        System.out.println(valor);  
    }  
}  
  
public class Filho1 extends Pai {  
    public void metodo(Integer valor){  
        System.out.println(valor + 1);  
    }  
}  
  
public class Filho2 extends Pai {  
    public void metodo(Integer valor){  
        System.out.println(valor + 2);  
    }  
}  
  
public class Exercicio {  
    public static void main(String  
  
args) {  
        Pai pai = new Filho1();  
        pai.metodo(2);  
        // ao executar o metodo passando o valor 2 por parametro a saída no console será igual a 3  
    }  
}
```

Assinale a alternativa que contenha o conceito de orientação a objetos que justifique a saída no console do resultado da conta feita na classe Filho1.java.

### ALTERNATIVAS

- ☐ abstração
- ☐ hierarquia
- ☒ polimorfismo
- ☐ subordinação
- ☐ encapsulamento

### 5ª QUESTÃO

O conceito de herança é um dos pilares da orientação a objetos, que consistem em uma forte ligação entre classes que podem compartilhar características e comportamentos, mantendo uma relação de dependência.

TOKUMOTO, R. C. Programação de Sistemas II. Maringá-Pr.: Unicesumar, 2017.

Analise todas as afirmações a respeito da implementação de herança com Java.

I - Apenas classes abstratas podem possuir classes filhas.

II - Atributos e métodos "protected" podem ser acessados pelas classes filhas.

III - Atributos e métodos "final" podem ser acessados pelas classes filhas.

IV - Classes "final" não podem possuir subclasses.

É correto o que se afirma em:

### ALTERNATIVAS

- ☐ I e II, apenas.
- ☐ II e III, apenas.
- ☐ I e IV, apenas.
- ☒ II e IV, apenas.
- ☐ I, II, III e IV.

### 6ª QUESTÃO

Java possui palavras reservadas que são modificadoras de acesso, *public*, *private*, *protected* e sem modificador (*default*). Dentre eles, dois modificadores restringem o acesso à atributos e métodos para acesso em classes filhas que não estejam na mesma pasta.

TOKUMOTO, R. C. Programação de Sistemas II. Maringá-Pr.: Unicesumar, 2017.

Assinale a alternativa que contenha esses dois modificadores.

### ALTERNATIVAS

- ☐ public e private
- ☐ public e protected
- ☒ private e protected
- ☐ private e default (sem modificador)
- ☐ protected e default (sem modificador)

### 7ª QUESTÃO

Um dos problemas que podem ocorrer em códigos Java é na comparação de Strings. Em Java, String é uma classe e não um tipo primitivo, bastante utilizada no desenvolvimento, em que cadeias de caracteres podem ser armazenadas nesse tipo de dado.

TOKUMOTO, R. C. **Programação de Sistemas I**. Maringá-Pr.: Unicesumar, 2017.

Sabendo disso, suponha que tenhamos duas variáveis do tipo String, cujos nomes são valor1 e valor2 e que ambas estejam preenchidas.

Marque abaixo expressão que faz corretamente a comparação de valor1 e valor2, caractere por caractere.

### ALTERNATIVAS

- ☐ valor1 = valor2
- ☐ valor1 == valor2
- ☐ valor1 := valor2
- ☒ valor1.equals(valor2)
- ☐ String.equals(valor1, valor2)

### 8ª QUESTÃO

Para controlar o fluxo de execução de um programa Java, pode-se utilizar as instruções if e else. Veja abaixo um trecho de código que utiliza essas instruções:

```
10: int pontos = 1000;
11: if (pontos = 0)
12:     System.out.println("Você agora tem 0 pontos.");
13: else if( pontos > 0 && pontos <= 1000) {
14:     System.out.println("Você está no nível básico");
15: } else if( pontos > 1000 && pontos <=10000){
16:     System.out.println("Você está no nível intermediário.");
17: } else System.out.println("Parabéns, você chegou ao nível avançado.");
```

Considerando o trecho de código mostrado, marque a opção correta:

### ALTERNATIVAS

- ☐ O código compila e mostra a mensagem "Você está no nível básico".
- ☐ O código compila e mostra a mensagem "Você está no nível intermediário ".
- ☐ O código apresenta erro de compilação na linha 11, porque o uso de chaves é obrigatório na instrução if.
- ☐ O código apresenta erro de compilação na linha 17, porque o uso de chaves é obrigatório na instrução else.
- ☒ O código apresenta erro de compilação na linha 11, porque a instrução if deve receber uma expressão booleana e está recebendo um inteiro.

### 9ª QUESTÃO

Dado o seguinte código:

```
public class Exercicio {
    public static void main(String
        args) {
        for (int i = 0; i < 10; i++) {
            if (i == 2) {
                _____
            }
            System.out.println(i);
        }
    }
}
```

Assinale a alternativa que completa corretamente o código de forma que o laço de repetição pare assim que a condição do *if* for verdadeira.

### ALTERNATIVAS

- ☐ finish him;
- ☐ continue;
- ☒ break;
- ☐ stop;
- ☐ end;

### 10ª QUESTÃO

Objetos do mundo real compartilham duas características: todos eles têm estado e comportamento. Identificar o estado e o comportamento para objetos do mundo real é uma ótima forma de começar a pensar em termos de programação orientada a objetos.

WHAT IS AN OBJECT? Disponível em: <<https://docs.oracle.com/javase/tutorial/java/concepts/object.html>> Acesso em 18 set. 2018.

Na programação orientada a objetos, estado e comportamento de um objeto são representados, respectivamente, através de:

### ALTERNATIVAS

- ☐ classes e métodos
- ☐ métodos e classes
- ☒ atributos e métodos
- ☐ métodos e atributos
- ☐ classes e bibliotecas