

Programação Orientada a Objetos com Java e WEB

Lista de Exercícios I - codificação de classes, objetos e métodos

Prof. Dr. Antonio Marcos Selmini (selmini@fiap.com.br)

Esses exercícios têm como objetivo praticar os conceitos de orientação a objetos apresentados durante as aulas. A resolução dos exercícios será divulgada no repositório da sala (<https://github.com/fiap-poo-XXXX>). Você deverá substituir XXXX pela sua sala de aula.

1. Codifique uma classe chamada **Bicicleta** que tenha os seguintes atributos: **cor** (string), **número de marchas** (inteiro) e **valor** (real). A sua classe deverá ter também um método para retornar todos os dados de um objeto **Bicicleta**.

Codificar um programa orientado a objetos em Java para instanciar a classe **Bicicleta**. Imprima todos os dados do objeto no vídeo. Os dados da bicicleta deverão ser informados pelo usuário do sistema via terminal.

2. Considere o programa abaixo codificado na linguagem de programação Java. Após a sua execução quais os valores armazenados nos atributos de cada objeto? **Este exercício é para você tentar descobrir o valor de cada variável sem executar o programa, ou seja, fazer o teste de mesa (ou debugar sem usar uma IDE como o Eclipse, VSCode ou IntelliJ).**

```
public class Exercicio {
    int x, y, z;

    public static void main(String[] args) {
        Exercicio e1 = new Exercicio();
        Exercicio e2 = new Exercicio();
        e1.x = e1.z + e2.x + e2.y;
        e2.y = 2*e1.x + e2.z;
        e1.z = e2.z + e1.y;
        e2.z = e2.z + 2;
    }
}
```

3. Desenvolver uma aplicação que permita gerar retângulos. A aplicação deve permitir que o usuário informe as características do retângulo a ser criado. A aplicação deve permitir o cálculo da área e do perímetro de um retângulo.

Teste a sua aplicação gerando pelo menos um retângulo. Forneça seus dados e imprima sua área e seu perímetro no vídeo.

4. Desenvolva uma aplicação orientada a objetos que permita representar uma temperatura em graus celsius. A sua aplicação deverá permitir a conversão da temperatura para as escalas fahrenheit e kelvin. As expressões para conversão de temperatura a partir da escala celsius é dada a seguir.

$$f = \frac{9}{5}c + 32$$

$$k = c + 273$$

onde f representa a temperatura na escala fahrenheit, c em celsius e k em kelvin. Teste a sua aplicação gerando pelo menos um objeto temperatura fornecendo seu valor e imprimindo as conversões no vídeo.

5. Desenvolva uma aplicação orientada a objetos que permita representar uma distância em metros. A sua aplicação deverá permitir a conversão da distância de metros para milhas e de metros para pés. Lembre-se que 1 m = 0,000621371 milhas e 1 m = 3,28084 pés.
6. Desenvolva uma classe chamada **Hora** contendo os atributos: **horas**, **minutos** e **segundos**. O método construtor deverá chamar métodos para validar os valores dos atributos, ou seja, o valor das horas deve estar entre 0 e 23, o valor dos minutos e dos segundos entre 0 e 59. Quando o valor não for válido deverá ser atribuído o valor 0. A classe também deverá ter um método para retornar a hora no seguinte formato: **hh:mm:ss**. Em seguida, escreva um programa Java (uma classe contendo o método **main()**) para testar a classe **Hora**.
7. Analise o programa abaixo codificado na linguagem de programação Java. A partir da análise realizada na classe **Main**, escreva a classe **Bola** com todas as suas informações.

```
public class Main {

    public static void main(String[] args) {

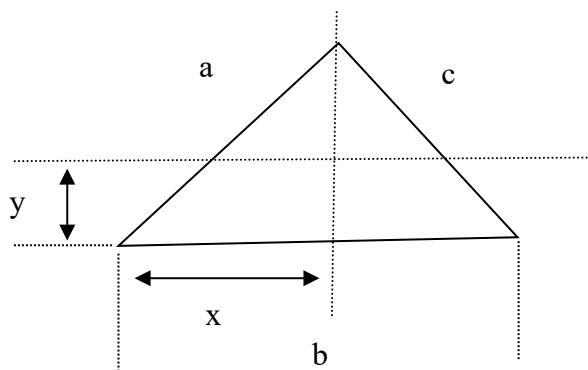
        Bola bola1 = new Bola("azul", 2.75);
        Bola bola2 = new Bola("branca", 3.5);
        Bola bola3 = new Bola("vermelha", 3.25);

        Bola aux = bola1.maiorBola(bola2, bola3);
        System.out.println("Dados da bola com o maior raio -> "+aux.retornarDados());
    }
}
```

8. Escreva uma classe para representar uma equação do segundo grau. A classe deverá ter o método para calcular e retornar as raízes da equação. Escreva um programa em Java para testar a sua classe.

Observação: para calcular raiz quadrada é utilizado o método estático *Math.sqrt()* e, para calcular potenciação o método estático *Math.pow(a, b)*. A classe *Math* está no pacote *java.lang* que não precisa ser importado.

9. Desenvolva uma classe chamada **Círculo** contendo apenas o **raio** como atributo. A classe deverá ter métodos para alterar o valor do raio, retornar o valor do raio, calcular a área e o perímetro. A área do círculo é dada por: $\pi * r^2$, onde r é o raio. O perímetro é calculado por: $2 * \pi * r$, onde r é o raio. Teste a classe, instanciando objetos e chamando os métodos. O valor de π pode ser obtido a partir da constante *Math.PI*, do pacote *java.lang*.
10. Desenvolva uma classe para representar um triângulo. Defina métodos para calcular a área *S*, o perímetro *P* e o **centróide** de coordenadas *x* e *y*. Construa a classe principal para testar a classe que representa o triângulo.



$$S = \sqrt{p(p-a)(p-b)(p-c)}$$

$$p = \frac{a+b+c}{2}$$

$$P = a + b + c$$

$$y = \frac{a}{3} \sqrt{1 - \lambda^2} \quad x = \frac{b + a\lambda}{3}$$

$$\lambda = \frac{a^2 + b^2 - c^2}{2ab}$$

Observação: para calcular raiz quadrada é utilizado o método estático `Math.sqrt()` e, para calcular potenciação o método estático `Math.pow(a, b)`. A classe `Math` está no pacote `java.lang` que não precisa ser importado.

11. Todas as professoras e professores recebem pelo número de aulas que ministram. O cálculo do salário está disciplinado na Convenção Coletiva de Trabalho. A remuneração mensal é formada por, no mínimo, três itens que devem se somados para se obter o salário bruto mensal do professor:

a) **Salário base**

Corresponde ao valor mensal das aulas ministradas sobre o qual serão calculados o descanso semanal remunerado e o adicional de hora-atividade. Para calcular, basta multiplicar o **número de aulas semanais** por 4,5 semanas e pelo **valor da hora-aula**. Se o professor(a) tiver o título de mestre, haverá um aumento de 8,5% no valor do salário base. Caso o professor(a) tenha o título de doutor, o aumento será de 12%, por outro lado, se o professor(a) não for mestre ou doutor, não terá nenhum aumento.

b) **Adicional de hora-atividade**

É o adicional destinado exclusivamente as atividades realizadas fora da escola, como preparação de aulas e correção de provas e trabalhos. As convenções coletivas da educação básica e do ensino superior prevê adicional de 5%, aplicado sobre o salário base.

c) **Descanso Semanal Remunerado**

Corresponde a 1/6 sobre a remuneração total, ou seja, deve ser calculado sobre a soma do salário-base, da hora-atividade, das horas extras e demais adicionais. A discriminação do DSR no holerite é obrigatória, exceto para professores mensalistas de educação infantil até a 4ª série do ensino fundamental.

Desenvolva um **projeto Java orientado a objetos** que gere três objetos representando os professores e, em seguida, imprima no vídeo o valor do salário bruto de cada professor. A entrada dos dados para cada professor deverá ser realizada via teclado.

12. Considere o programa abaixo codificado na linguagem de programação Java. Após a sua execução quais os valores armazenados nos atributos de cada objeto?

```

public class Exercicio {
    int x, y, z;

    public Exercicio(int x, int y, int z) {
        this.x = x;
        this.y = y;
        this.z = z;
    }

    public static void main(String[] args) {
        Exercicio e1 = new Exercicio(1, 2, 3);
        Exercicio e2 = new Exercicio(4, 1, 0);
        e1.x = e1.z + e2.x + e2.y;
        e2.y = 2*e1.x + e2.z;
        e1.z = e2.z + e1.y;
        e2.z = e2.z + 2;
    }
}

```

13. Analise o programa abaixo codificado na linguagem de programação Java. Após a execução do programa, quais os valores dos atributos de cada objeto?

```

public class Exercicio {
    int x, y, z;
    static int a;

    public Exercicio(int x, int y, int z) {
        this.x = x;
        this.y = y;
        this.z = z;
    }

    public static void main(String[] args) {
        Exercicio e1 = new Exercicio(1, 2, 3);
        Exercicio e2 = new Exercicio(4, 1, 0);
        e1.x = e1.z + e2.x + e2.y + Exercicio.a;
        e2.y = 2*e1.x - e2.z;
        Exercicio.a = e2.x;
        e1.z = e2.z + e1.y;
        e2.z = e2.z + 2;
        Exercicio.a = 2* Exercicio.a + e1.y;
    }
}

```

14. Considere as classes abaixo codificadas na linguagem de programação Java. Após a execução do programa, quais os valores nos atributos de cada objeto?

```
public class Exercicio {
    int a, b;

    public Exercicio() {
        this(0, 0);
    }

    public Exercicio(int a, int b) {
        this.a = a;
        this.b = b;
    }
}
```

```
public class Misterio {
    public Exercicio misterio(Exercicio ex) {
        ex.a *= 2;
        ex.b++;
        return ex;
    }
}
```

```
public class Teste {
    public static void main(String[] args) {
        Exercicio e1 = new Exercicio();
        Exercicio e2 = new Exercicio(5, 3);
        Misterio m = new Misterio();
        e1 = m.misterio(e1);
        e2 = m.misterio(e2);
    }
}
```

15. Analise o programa abaixo codificado na linguagem de programação Java e assinale a alternativa que corresponde a saída gerada pelo programa.

```
public class Exercicio {

    static int valor = 7;

    public static void main(String[] args) {
        new Exercicio().fazerAlgo(valor);
        System.out.println(" "+valor);
    }

    public void fazerAlgo(int valor) {
        valor++;
        for(int aux = 3; aux < 6; aux++);
        System.out.print(" "+valor);
    }
}
```

- a) 5 7
- b) 5 8
- c) 8 7
- d) 8 8
- e) Ocorre erro na compilação do programa.