



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina: Programação Orientada a Objetos (EAD-05.032)
AD1 – 1º semestre de 2023.

IMPORTANTE

(Leia atentamente antes de iniciar a avaliação)

A resposta (programa) deve ser entregue pela plataforma com código fonte (extensão “.java”), o qual pode estar zipado (.zip) ou não. Respostas entregues fora do formato especificado, por exemplo, em arquivos com extensão “.pdf”, “.doc” ou outros, não serão corrigidas.

Serão aceitos apenas soluções escritas na linguagem Java. Programas com erro de interpretação não serão corrigidos. Evite problemas de correção ao utilizar recursos avançados de Java. Use o Java na versão 8 da linguagem.

Não serão aceitas soluções usando métodos ou estruturas que não foram apresentados nas aulas preparatórias para as avaliações AD1 e AP1.

Quando o enunciado de uma questão inclui especificação de formato de entrada e saída, tal especificação deve ser seguida à risca pelo programa entregue. Atender ao enunciado faz parte da avaliação e da composição da nota final.

Os exemplos fornecidos nos enunciados das questões correspondem a casos específicos apontados para fins de ilustração e não correspondem ao universo completo de entradas possíveis especificado no enunciado. Os programas entregues devem ser elaborados considerando qualquer caso que siga a especificação e não apenas os exemplos dados. Essa é a prática adotada tanto na elaboração das listas exercícios desta disciplina quanto no mercado de trabalho.

Faça uso de boas práticas de programação, em especial, na escolha de identificadores de variáveis, subprogramas e comentários no código.

As respostas deverão ser entregues via atividade específica na Plataforma antes da data final de entrega estabelecida no calendário de entrega de ADs. Não serão aceitas entregas tardias ou substituição de respostas após término do prazo.

As ADs são um mecanismo de avaliação individual. As soluções podem ser buscadas por grupos de alunos, mas a redação final de cada prova tem que ser individual. Respostas plagiadas não serão corrigidas.

Questão Única

Nesta questão, você deverá implementar classes Java que atendam ao cenário do problema descrito abaixo. Os quesitos avaliados serão a aplicação adequada de (i) abstração, (ii) encapsulamento, (iii) herança e (iv) polimorfismo. Indique no código fonte, na forma de comentário, onde cada um desses princípios foi empregado na solução entregue.

Descrição do Cenário:

A venda de computadores e componentes de informática pode ser feita por lojas considerando que os itens são vendidos de forma avulsa ou agregada. Por exemplo, é possível comprar placas de vídeo, processadores e pentes de memória individualmente ou na forma de um computador montado.

Seu trabalho nesta questão é modelar e implementar este cenário de vendas, onde todos os itens disponíveis têm um código sequencial que os identifica de forma inequívoca. Qualquer item, independente de ser um item avulso ou um item montado, deve ser capaz de informar qual o seu preço. O preço de um item de compra é calculado da seguinte forma:

- Todos os itens avulsos possuem preço próprio.
- Um item montado tem o seu preço calculado pela soma dos preços de suas partes.

Nesta questão, projeto classes para os seguintes itens de compra:

- Processadores: são itens avulsos que se diferenciam uns dos outros apenas pela descrição (string) e pelo preço (double).
- Pentec de Memória: são itens avulsos que se diferenciam uns dos outros pela descrição (string), capacidade em gigabyte (integer) e preço (double).
- Monitores de Vídeo: são itens avulsos que se diferenciam uns dos outros pela descrição (string), resolução em polegadas (double) e preço (double).
- Teclado: são itens avulsos que se diferenciam uns dos outros pela descrição (string) e preço (double).
- Gabinetes: são itens que o usuário pode montar pela inclusão de processadores e pentes de memória.
- Computadores: são itens que o usuário pode montar pela inclusão de gabinetes, monitores e teclados.

Você pode utilizar o programa a seguir para testar sua implementação. Porém, conforme informado na primeira página deste documento “Os exemplos fornecidos nos enunciados das questões correspondem a casos específicos apontados para fins de ilustração e não correspondem ao universo completo de entradas possíveis especificado no enunciado. Os programas entregues devem ser elaborados considerando qualquer caso que siga a especificação e não apenas os exemplos dados.”.

```

class Teste {
    public static void main(String[] args) {
        /* Criar diversos itens avulsos */
        Processador cpu1 = new Processador("Intel CORE i5", 790.0);
        Processador cpu2 = new Processador("Intel CORE i7", 1599.99);
        Processador cpu3 = new Processador("AMD Ryzen 5", 489,90);
        Placa mem1 = new Memoria("Kingstone", 8, 109.0);
        Placa mem2 = new Memoria("Corsair", 32, 449.99);
        Placa mem3 = new Memoria("Corsair", 32, 449.99);
        Monitor mon1 = new Monitor("Monitor LED", 17.0, 379.0);
        Monitor mon2 = new Monitor("Monitor LED", 19.5, 590.0);
        Teclado tec1 = new Teclado("Teclado USB", 30.0);
        Teclado tec2 = new Teclado("Teclado Bluetooth", 99.99);
        /* Criar gabinetes */
        Gabinete gab1 = new Gabinete("Super Dragon");
        Gabinete gab2 = new Gabinete("Power Gamer");
        /* Cria os computadores */
        Computador com1 = new Computador("Super Dragon Completo");
        Computador com2 = new Computador("Power Gamer Completo");
        /* Monta gabinetes */
        gab1.inclui(cpu1);
        gab1.inclui(mem1);
        gab1.inclui(mem2);
        gab2.inclui(cpu2);
        gab2.inclui(mem3);
        /* Monta computadores */
        com1.inclui(gab1);
        com1.inclui(mon1);
        com1.inclui(tec1);
        com2.inclui(gab2);
        com2.inclui(mon2);
        com2.inclui(tec2);
        /* Mostra os computadores */
        System.out.println(com1);
        System.out.println(com2);
    }
}

```