

	<b>Trabalho Prático de POO I</b>		
	<b>Professor:</b> Suelen Mapa de Paula		
	<b>Data Entrega:</b> 16/11/2025	<b>Valor:</b> 20,0	<b>Média:</b> 12,0
<b>O Trabalho pode ser realizado em grupo de até 3 alunos</b>			

**Objetivo:** praticar conceitos básicos de programação orientada a objetos utilizando a linguagem de programação Java.

### **Descrição:**

Seu José tem dois netinhos muito levados que estão passando as férias em sua casa. Para mantê-los entretidos enquanto trabalha em sua mercearia, ele contratou você para desenvolver um jogo chamado “Caça ao Tesouro”. O jogo consiste em permitir que dois jogadores escondam seus tesouros em um tabuleiro e tentem descobrir onde estão os tesouros do oponente. Cada jogador possui seu próprio tabuleiro e realiza jogadas alternadas para tentar localizar os tesouros adversários. Vence o jogo aquele que encontrar todos os tesouros do oponente primeiro ou se exceder o número de jogadas, quem fizer mais pontos.

### **O jogo deve incluir :**

- Um tabuleiro de tamanho 10x10 para cada jogador.
- Cada jogador terá 8 tesouros (sendo que eles podem ser amarelos - 4,0 pts; verde - 6,0 pts e vermelho - 10,0 pts)
- Os jogadores devem posicionar seus tesouros no tabuleiro, na seguinte quantidade: 3 verdes, 3 amarelos e 2 vermelhos.
- Os jogadores se alternam para tentar adivinhar a posição dos tesouros do oponente.
- O jogo termina quando todos os tesouros de um jogador forem encontrados ou se atingir 20 partidas (10 partidas para cada jogador).
- Cada jogador acumula pontos conforme encontra tesouros (4, 6 ou 10).
- Caso o jogo termine antes de alguém encontrar todos os tesouros (por limite de jogadas), vence quem tiver maior pontuação.

### **Validações obrigatórias:**

- O programa deve impedir posicionamentos repetidos de tesouros.
- O jogador não pode jogar duas vezes na mesma posição.
- O jogo deve exibir mensagens de erro claras.

### **Requisitos:**

Para resolver o exercício, você deverá obrigatoriamente implementar o jogo aplicando os conceitos de orientação objeto vistos em aula até o momento (classe, atributos , métodos, objetos, construtores, sobrecarga, encapsulamento - visibilidade e os métodos

getters e setters. Sendo assim, seu exercício deve :

1. Modelar e Implementar de maneira coerente as classes que descrevem o problema, além das classes jogador e jogo. Sendo que:
  - Classe Jogador: representa um jogador no jogo. Cada jogador tem um tabuleiro e pode posicionar os tesouros e atacar o oponente.
  - Classe Jogo: gerencia o fluxo do jogo, incluindo a configuração inicial, os turnos dos jogadores e as condições de vitória.
2. Entre as classes deve haver pelo menos uma relação de composição ou agregação, além das associações;
3. Use encapsulamento com coerência real: atributos privados e acesso apenas via getters e setters que façam sentido para o sistema.
4. Além de construtores, crie também métodos sobrecarregados úteis
  - `posicionarTesouro(int linha, int coluna)`
  - `posicionarTesouro(int linha, int coluna, String cor)`
5. Pelo menos em uma das classes é esperado que seja utilizado alguma Collection (Vector ou ArrayList). Uma dica seria utilizá-la para armazenar as coordenadas de tentativas ou os objetos “Tesouro” encontrados.
6. Além das classes, atributos e métodos especificados, você pode criar outros para implementar sua lógica. Mas lembre-se de manter a coerência com os requisitos especificados para cada classe e de aplicar as regras do encapsulamento.

#### O que deve ser entregue:

1. Código fonte do programa em **Java** (bem comentado).
2. Diagrama UML com as classes, seus componentes e suas relações (associação, agregação ou composição)
3. Documentação do trabalho. Entre outras coisas, a documentação deve conter:
  - a) Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa. Se for necessário, utilize figuras e diagramas para tornar sua explicação clara e didática.
  - b) Implementação: descrição sobre a implementação do programa. Deve ser detalhada a estrutura das classes modeladas, os principais atributos e o funcionamento dos principais métodos utilizados, enfatizado a abstração feita e detalhes de especificação que porventura estejam omissos no enunciado.
  - c) Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
  - d) Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso.
  - e) Formato: mandatoriamente em PDF

4. Elaboração de um vídeo com duração máxima de 3 minutos, apresentando o sistema desenvolvido, as classes abstraídas para representar o problema e a demonstração do correto funcionamento da solução. Todos os integrantes do grupo devem participar da gravação, seja na explicação, na demonstração prática ou em ambas.

**Como deve ser feita a entrega:**

A entrega DEVE ser feita através da tarefa cadastrada na plataforma moodle, na forma de um único arquivo zipado, contendo o código, os arquivos e a documentação.

**Comentários Gerais:**

- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar;
- Clareza, indentação e comentários no programa também vão valer pontos;
- O trabalho por ser feito em trios (grupo de TRÊS alunos);
- Trabalhos copiados (e FONTE) terão nota ZERO;
- Trabalhos entregue em atraso serão aceitos, todavia a nota atribuída ao trabalho será zero.