



Fundação CECIERJ - Vice

Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Programação Orientada a Objetos

AD1 1º semestre de 2021.

Nome:

Matrícula:

Pólo:

IMPORTANTE

A resposta (programa) deve ser entregue pela plataforma em um arquivo único 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.

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.

Suponha que fomos contratados para implementar uma rede social minimalista. Essa rede permite a criação de postagens e de curtidas e estas postagens. A classe para teste dessa aplicação tem o conteúdo abaixo:

```
public class AD1_2021_1 {
    public static void main(String[] args) {
        Amigo fulano = new Amigo("Fulano");
        Amigo ciclano = new Amigo("Ciclano");
        Amigo beltrano = new Amigo("Beltrano");

        Rede paraíso = new Rede();
        paraíso.adicionarAmigo(fulano);
        paraíso.adicionarAmigo(ciclano);
        paraíso.adicionarAmigo(beltrano);

        Post bomdia = new Post("Bom dia!");
        Post boatarde = new Post("Bom tarde!");
        Post boanoite = new Post("Bom noite!");

        fulano.postar(bomdia);
        fulano.postar(boatarde);
        ciclano.postar(boanoite);

        beltrano.curtir(bomdia);
        beltrano.curtir(boatarde);
        beltrano.curtir(boanoite);
        fulano.curtir(boanoite);

        System.out.println("Timeline:");
        System.out.println(paraíso.timeline());
        System.out.println("Post mais curtido de um usuário:");
        Post maiscurtido = ciclano.retornaPostMaisCurtido();
        System.out.println(maiscurtido);
        System.out.println("Quem curtiu: " +
maiscurtido.retornaNomesQueCurtiram());
    }
}
```

Crie todas as classes utilizadas no código acima, para que este rode sem alterações. Utilize conceitos de OO sempre que possível. Limite cada classe para tratar somente o que diz respeito a esta. Por exemplo, a classe Post só deve manipular informações referentes a posts. O resultado de sua execução, para este exemplo de entrada acima, deve ser como na saída abaixo:

```
Timeline:
Fulano:
Bom dia!: 1 curtidas
Bom tarde!: 1 curtidas
Ciclano:
Bom noite!: 2 curtidas
Beltrano:

Post mais curtido de um usuário:
```

Bom noite!: 2 curtidas

Quem curtiu: Beltrano Fulano