

**Data de Entrega: 09.12 (sexta-feira) – NÃO SERÃO ACEITOS TRABALHOS ENVIADOS APOS ESTA DATA.**

**O que deve ser entregue:**

1. Deve ser enviado por email um arquivo zipado contendo:
  - a. (5,0 pontos) um documento PDF descrevendo os algoritmos implementados, os testes realizados e o gráfico comparativo das execuções. Este documento deve possuir uma conclusão discutindo os resultados obtidos;
  - b. (5,0 pontos) os códigos fontes dos programas implementados.

Observe que cada parte do trabalho possui uma pontuação específica.

**Regras para entrega do trabalho prático:**

Este trabalho deve ser realizado em grupo de no máximo 3 alunos. Cada grupo deve escolher um representante que será responsável por enviar o trabalho em um arquivo zipado, cujo nome deve respeitar o seguinte formato: CK069\_<matricula do aluno>\_T<numero do trabalho pratico>, por exemplo, CK069\_123456\_T1 é o arquivo contendo o trabalho pratico 1 do estudante com matricula 123456. O arquivo compactado deve ser enviado por email do professor jose.macedo@lia.ufc.br até meia-noite do prazo estipulado acima. O subject da mensagem deve ser "[ck060] T1: <sua matricula>", exemplo "[ck069] T1: 123456". Caso o aluno não respeite as regras apresentadas anteriormente, poderá sofrer redução na nota do trabalho.

**Como será avaliado o trabalho:**

A avaliação deste trabalho levará em conta a correção e qualidade das questões respondidas. Além da implementação realizada.

## Trabalho Pratico 7 – Algoritmos de Troca de Pagina

Escreva um programa que implemente os algoritmos de troca de paginas: FIFO, LRU, LFU, MFU e Optimal (algoritmo ótimo), apresentados no capítulo 9. Os algoritmos implementados deve estender a classe abstrata apresentada abaixo. Cada classe deve implementar o método **insert**, o qual deve ser usado a cada momento que uma pagina for referenciada.

```
public abstract class ReplacementAlgorithm
{
    protected int pageFaultCount;

    protected int pageFrameCount;

    public ReplacementAlgorithm (int pageFrameCount)
    {
        if (pageFrameCount < 0)
            throw new IllegalArgumentException();

        this.pageFrameCount = pageFrameCount;
        pageFaultCount = 0;
    }

    public int getPageFaultCount()
    {
        return pageFaultCount();
    }

    public abstract void insert (int pageNumber);
}
```

Duas classes são dadas, disponíveis na pagina do curso, as quais permitirão testar seus algoritmos. As classes são:

1. `PageGenerator.java` – classe que gera string com referencias as paginas com numero de paginas variando de 0 a 9. O tamanho da string é passado para o contrutor da classe `PageGenerator`. Apos a construção do objeto `PageGenerator`, o metod `getReferenceString()` retorna a string de referencias como um array de inteiros.
2. `Test.java` – classe usada para testar seus algoritmos implementados. Você deve invocar esta classe da seguinte forma:

```
Java Test <tamanho string de referencias> <quantidade de frames>
```

Apos implementar os algoritmos, teste-os com os seguintes valores de frames dada uma string de referencias de tamanho 1000: 1, 10, 50, 100, 500, 750, 900, 1000.

Gere um gráfico comparando o numero de Page Faults gerados por cada um dos algoritmos implementados. Discuta os resultados obtidos descrevendo textualmente o que foi possível concluir da execução dos experimentos.