

Segundo Exercício-Programa

Instruções:

- I) O trabalho deve ser feito **individualmente**.
- II) As postagens das soluções devem ser feitas no Col até às **23:59** do dia **30/06/2012**.
- III) Se for detectado **plágio**, os alunos copiados e copiadores serão punidos com nota zero nessa atividade.
- IV) Trabalhos enviados por email serão completamente **ignorados**.

Neste exercício, deve ser implementado um aplicativo para realizar duas tarefas:

- Criptografar um trecho de texto pleno para produzir um trecho de texto cifrado.
- Descriptografar um trecho de texto cifrado para produzir um trecho de texto pleno.

Para isso, você deve obrigatoriamente utilizar o padrão de projeto Strategy para permitir que o usuário escolha, em tempo de execução, a estratégia de criptografia ou descriptografia a ser utilizada.

A primeira estratégia que deve ser disponibilizada é o algoritmo cifrador Caesar, que substitui cada letra por aquela que se encontra a 3 posições adiante na codificação ASCII. Nesse caso, a letra 'B' torna-se 'E'; 'e' torna-se 'h', etc.

A codificação ASCII define a ordem dos caracteres, com cada um destes sendo representado por um valor inteiro no intervalo [0, 255]. (Pode-se encontrar facilmente a tabela ASCII completa em vários livros de programação e sítios da internet).

Dica: Java permite que você trabalhe com aritmética em caracteres. Em particular, se você adicionar 3 a uma variável char, terá como resultado o caractere encontrado 3 posições à frente na tabela ASCII. Os casos especiais a serem tratados envolvem apenas os valores ASCII nas últimas posições na tabela, que devem ser mapeados, de forma circular, nos que se encontram no início da mesma. Por exemplo, o caracter de valor ASCII 255 deve ser substituído pelo de valor ASCII 2.

A segunda estratégia de criptografia deve ser implementada pelo algoritmo cifrador Rail Fence, que usa transposição ao invés de substituição. O Rail Fence, cuja tradução literal é "paliçada", é considerado um procedimento de transposição geométrica porque segue um padrão fixo em zig-zag.

Nesse caso, considerando a utilização de 3 trilhas (rails), a codificação da frase "como vai?" seria realizada da seguinte forma:

Passo 1: Geração das trilhas em zig-zag:

c	-	-	-	-	-	-	?
-	o	-	o	-	v	-	i
-	-	m	-	-	-	a	-

Passo 2: Varredura das trilhas, de forma sequencial, para gerar a frase cifrada:

c ?oovima

A entrada do aplicativo deve ser um trecho de texto fornecido pelo usuário e a saída deverá apresentar o texto original, seguido, abaixo, pelo texto cifrado.

A utilização de interface gráfica vale até 1 ponto adicional na nota do exercício.