

1. Como sugerido no slide 20 da Unidade 9, grave um arquivo texto com o valor “12345” (String) e grave um arquivo binário com o inteiro 12345 (int). Faça a leitura de ambos os arquivos no formato texto (Reader, readLine) e no formato binário (DataInputStream, readInt). Não deixe que exceções interrompam a execução. Qual o resultado? O que é possível concluir?
2. Faça um programa em Java que gere uma versão criptografada de um arquivo texto trocando cada caractere de código ASCII c pelo caractere de código ASCII $c+k$, onde k é um valor especificado pelo usuário entre 1 e 200. Por exemplo, o caractere ‘A’ cujo código ASCII é 65, sendo a chave $k=15$, será substituído pelo caractere de código 80 (‘P’). Fique atento para não gerar códigos ASCII fora da faixa permitida (neste caso, o que fazer? Qual sua sugestão?)
Seu programa deve abrir a caixa de seleção de arquivos do sistema operacional (JFileChooser) para que o usuário escolha o arquivo texto a criptografar. O arquivo gerado deve ter o mesmo nome do arquivo de entrada acrescido da expressão “-saída”. Exemplo: MeuTexto.txt → MeuTexto-saida.txt
3. Escreva uma classe Java com um método capaz de ler uma imagem de dimensões 400x400 em formato RAW (arquivo binário contendo apenas uma sequência de bytes) em que cada byte indica um tom de cinza entre 0 e 255. Exemplo de uma imagem em tons de cinza:



Este método tem como entrada o Path referente ao arquivo da imagem e como retorno um vetor (int[]) com a quantidade de ocorrências de cada tom na imagem: `public int[] histograma(Path p).`

Para testar, crie uma classe de testes que gere automaticamente um arquivo RAW 400x400 e proceda o teste por meio de assertivas de teste.

4. Uma estação meteorológica automatizada gera um arquivo com os dados que coleta diariamente. Numa hora programada, seus sensores captam o estado do ambiente e isto é gravado num arquivo binário, no seguinte formato:

Data – String com formato “dd/mm/aaaa”

Direção do vento – char + char (N , NE, NW, S , SE, SW, E , W)

Velocidade do vento – int (em km/h)

Índice pluviométrico – int (em mm)

Temperatura – double (em °C)

Num arquivo binário podem existir dados de vários meses, porém os dados sempre estão em ordem cronológica. Exemplo de arquivo binário de entrada está disponível no AVA – Bloco 3.



Estes dados precisam ser convertidos para um objeto correspondente, ou seja, para cada dia haverá um objeto da classe `ClimaDoDia`:

<code>ClimaDoDia</code>
<code>-data : Date</code>
<code>-ventoDirecao : String</code>
<code>-ventoVelocidade : int</code>
<code>-indicePluviometrico : int</code>
<code>-temperatura : double</code>

Para cada mês encontrado, deve-se gerar um arquivo de objetos com os respectivos dias, cujo nome deve ser “AAAA-MM.dat”, em que AAAA significa o ano e MM o mês.

5. A partir do exercício 2 da Lista 4 (Alunos da FURB) implemente serialização dos objetos, oferecendo ao usuário uma opção de salvar os alunos cadastrados. Caso haja um arquivo de objetos previamente criado, seu programa deve automaticamente carregar estes objetos, permitindo que o usuário continue a cadastrar novos alunos.