Considere que você é um programador de uma determinada instituição de ensino superior. Assim, você foi designado(a) a desenvolver uma aplicação Java para ler um arquivo texto, com informações de alunos, que foi enviado de um polo de ensino de outra região. Dessa forma, deve-se ler esse arquivo e escrever as informações via System.out.print.... A estrutura de cada linha do arquivo é a seguinte:

mmmmAAAAAAAAAAAAAAAAAAAAAAAAA99999999999SXXXXXXXXDDDDDDD...

Onde:

mmmm = Matricula do aluno com quatro dígitos

AAAAA.... = Nome do aluno com até 25 caracteres

99999999999 = cpf do aluno

S = código do sexo (1-Feminino, 2-Masculino)

XXXXXXXX = data de nascimento do aluno no formato ddmmaaaa (dd=dia, mm=mês e aaaa=ano)

DDDDDDD = código da disciplina. Formado por três letras e quatro dígitos. Exemplo: CMP2030, ADS1234..

Considerações:

a)O arquivo deve possuir, no mínimo, 10 linhas. Ou seja, pelo menos 10 alunos.

b)Você irá editar o seu próprio arquivo de texto em um editor de texto (por exemplo, bloco de notas ou outro de sua escolha), gravando esse arquivo com o nome **alunos.txt**;

c)O nome de cada aluno deve ser escrito em caixa alta (letras maiúsculas). Caso o nome possua menos de 25 caracteres, complete com espaços à direita até atingir os 25 caracteres;

d)Informe, pelo menos, 3 cpfs inválidos.

e)A data deve ser informada como indicado ddmmaaaa. Exemplos: 30052000, 18102005;

f)Um aluno pode estar matriculado em nenhuma, uma ou em várias disciplinas. Assim, cadastre pelo menos: dois alunos com uma disciplina; dois com duas disciplinas; um com três disciplinas; três com quatro disciplinas; dois com cinco disciplinas. No entanto, considere que o professor acesse seu arquivo e informa 8 disciplinas para um

determinado aluno. Portanto, o programa deve processar cada linha independentemente da quantidade de disciplinas que um aluno possui.

Esse programa deve apresentar um relatório na tela do computador (através de System.out.print....) no seguinte formato:

RELAÇÃO DE ALUNOS DE OUTROS PÓLOS DE ENSINO

Seq. Matr. Nome

001 1234 Jose Bonifácio 123456789.00* Masc 30/08/2005

DISCIPLINAS: AAA1234, CMP0001, ADS9090

002 3790 Elvis Presley Barbosa 789012345.76 Masc 10/05/1998

DISCIPLINAS: XXX456, ABC1010

003 8569 Maria Antonieta Xavier 789012345.76 Fem. 30/02/1998*

DISCIPLINAS: Sem disciplinas

.....

O programa deve ser desenvolvido utilizando os conceitos de orientação a objetos e do uso de camadas. Assim, o programa deve ler cada linha do arquivo e armazenar as informações em objetos específicos de acordo com os respectivos atributos para as classes Aluno e Disciplina, respectivamente. A classe Aluno possui o atributo **disciplinas** que é uma collection (Arraylist) de disciplinas. Cada objeto aluno deve ser armazenado em uma collection (Arraylist) denominada de **alunos**. Após a leitura de todas as linhas, o programa deverá ler a collection **alunos** para escrever o relatório no formato indicado anteriormente.

Além disso, o programa deverá solicitar (via JOtionPane) o valor inicial da sequência (Seq.) que será impressa no relatório, sendo que esse valor deve ser validado para ser informado dentro do intervalo 1 até 98. Caso o usuário informe um valor fora desse intervalo, o sistema deverá apresentar mensagem de erro e solicitar um valor correto.

Note que a sequência possui zeros á esquerda do número. Se o usuário informar 1, a sequência deverá ser impressa iniciando-se com 001, depois 002 e assim, sucessivamente. Se o usuário informar 50, a sequência deverá ser impressa iniciando-se com 050, depois 051 e assim, sucessivamente.

Opcionalmente, como **PRIMEIRO DESAFIO**, você pode desenvolver esse recurso de tal forma que o professor pode testar o programa para que a sequência seja impressa com uma quantidade específica de dígitos (4 ou 5, por exemplo). Neste caso, o sistema deverá inserir a quantidade correta de zeros á esquerda no relatório.

Note que o nome de cada aluno possui apenas a primeira letra (de cada parte do nome) em maiúsculas, sendo que o nome no arquivo está todo em maiúsculas.

Note que uma das linhas do exemplo do relatório possui um asterisco no CPF, indicando que este está incorreto. Para detectar se um CPF está ou não correto utilize a regra de cálculo dos dois últimos dígitos do CPF (dígitos de controle). Para utilizar esse cálculo você deve pesquisar suas regras e aplica-las no código do seu programa de tal forma que retorne se o CPF é válido ou não. Esse é o **SEGUNDO DESAFIO**.

Como **TERCEIRO DESAFIO** utilize no seu programa um recurso para que ele possa analisar se uma data de nascimento, informada no arquivo, está ou não correta. Caso esteja incorreta, escreva um asterisco logo após a data, como está informado no exemplo do relatório. Note que a data está formatada, diferentemente como está no arquivo de leitura.

A leitura de cada linha deve utilizar recursos de manipulação de String, pois as informações devem ser obtidas em posições específicas.

A data de apresentação do seu programa será estipulada pelo professor.

Sucesso na sua jornada!!