



TEMA 1: DOENÇA COVID-19

O momento de acompanhar os dados com precisão é relevante contra a Covid-19 e você estará elaborando um software respeitando todas as propriedades de Orientação a Objetos já estudadas para cadastrar as pessoas com nome completo (tem que possuir mais que 3 caracteres, sem espaços excessivos no início ou fim e estar em **String mutável**), gênero (**M**-masculino ou **F**-feminino) e um identificador único que nunca pode se repetir com valor a partir de um (1 valor definido em constante inteira chamada MINIMO).

Somente para as pessoas não contaminadas deverá ser armazenado também a idade em anos completos (zero é válido e indica bebê que não completou 1 ano ainda, mas o valor abaixo de zero ou acima de 130 serão inválidos, devendo essa idade máxima estar definida na constante MAXIMO), enquanto que para as pessoas contaminadas serão cadastradas a atual situação de saúde para cada uma das registradas em seu software em:

C => Curada;

E => Em tratamento;

F => Falecida.

Os dados cadastrados de cada pessoa serão armazenados em uma única estrutura de dados dinâmica em JAVA que os colecionará na memória para processamento, mas serão registrados, de maneira persistente, no Sistema Gerenciador de Banco de Dados Relacional **MySQL** na versão 8 ou superior. O nome completo informado pelo usuário deverá ser cadastrado no **MySQL** exatamente como o usuário digitou, com relação aos caracteres maiúsculos e minúsculos, e respeitando as exigências de validação esclarecidas anteriormente neste enunciado.

Analise este problema e elabore um programa orientado a objeto, completamente em ambiente gráfico (JAVA **Swing**), que cadastre os dados sobre a Covid-19, sendo validados todos os dados de entrada conforme as definições e restrições esclarecidas no início deste enunciado. Os valores das constantes que correspondem aos limites do identificador único e a idade das pessoas deverão ser sempre usados para tais verificações (validações), caso contrário a lógica de tais implementações estarão incoerentes a existência e facilidades dessas constantes em sua solução, sendo mensurado diretamente em sua avaliação.

Todos os dados informados neste enunciado são obrigatórios e serão apresentados em forma de tabela somente quando o usuário informar a opção correspondente a saída de dados que ele deseje visualizar (itens **b** e **d** esclarecidos na sequência). No entanto, essa forma de tabela deverá ser apresentada também usando um recurso gráfico de JAVA chamado **JTable** e não na console. O aspecto relacionado a pesquisa neste Trabalho Final acontecerá sobre o componente **JTable** que deverá ser usado, obrigatoriamente, na apresentação dessa forma de resultado tabelar para mostrar as pessoas cadastradas. Todas as *string* apresentadas no **JTable** deverão estar completamente em maiúsculo, não podendo ser alterado o dado original informado pelo usuário que está armazenado no **MySQL**.

Sua solução deverá oferecer para o usuário uma forma de menu de opções ou botões que propiciará ao usuário final escolher entre as opções relacionadas a seguir, estando todas



elas sempre disponíveis para o usuário acionar quando desejar, até que seja escolhida a opção **Encerrar**:

- a) **Cadastrar** – conforme desejo do usuário uma nova Pessoa será cadastrada no programa, respeitando todas as validações de entrada de dados coerentes (indicadas anteriormente neste enunciado);
- b) **Listar** - todas as pessoas cadastradas de maneira tabelar no *JTable*, respeitando o formato tabelar com uma única linha de cabeçalho indicando todos os dados fornecidos corretamente nesta forma tabelar (relatório com todos os cadastros e todos os dados cadastrados respeitando a sequência das colunas de Identificador, Nome completo (todo em maiúsculo), Gênero (por extenso), Idade e Situação de saúde (por extenso), estando ordenados pelo identificador de cada pessoa);
- c) **Consultar** - uma pessoa específica por meio de seu número identificador, em que todos os seus dados serão apresentados em uma **janela de diálogo**, sem nenhum ícone (figura do tipo de mensagem), estando cada dado cadastral em uma linha desta janela por extenso;
- d) **Pesquisar** - entre todas as pessoas, por meio do fornecimento de qualquer parte de seu nome, em que todos os registros que satisfizerem este nome serão recuperados e apresentados em ordem alfabética do nome das pessoas, de maneira tabelar aproveitando o mesmo *JTable* anterior (item *b*), mas com a ordenação dos dados diferente. Sua solução também deverá mostrar ao usuário desta pesquisa a quantidade de registros que foram recuperados e mostrados no final dessa apresentação (na mesma janela do *JTable*);
- e) **Encerrar** – termina a execução do programa apresentando exatamente os dados indicados abaixo na **console**, mostrando sempre os números com no mínimo duas casas numéricas inteiras (confira no exemplo abaixo que deverá possuir um salto de 20 espaços à esquerda para dar a impressão de centralizado ao usuário, mas note que todas as linhas abaixo estão alinhadas à esquerda – começam na mesma coluna). A última linha que totaliza as quantidades apresentadas deverá estar em cor diferente na console para chamar a atenção do usuário.

05 = NÃO CONTAMINADAS

03 = CONTAMINADAS EM TRATAMENTO

27 = CONTAMINADAS CURADAS

07 = MULHERES CONTAMINADAS FALECIDAS

08 = HOMENS CONTAMINADOS FALECIDOS

50 = TOTAL DE PESSOAS CADASTRADAS

Diante das características do problema apresentado deverá ser elaborada a especificação da solução que será desenvolvida, individualmente, por cada estudante, respeitando, exatamente, o modelo definido por seu professor, além do código que implementará tal solução. Esta solução só será aceita se for desenvolvida no ambiente gráfico JAVA (*Swing*) e possuir a modelagem usando o ASTAH com Diagrama de Casos de Uso e Diagrama de Classes completos da solução entregue que respeitará o padrão MVC e DAO.



Universidade de Brasília – UnB (FGA)

Engenharia de Software – Trabalho Final de OO

Esta especificação/documentação possuirá um único arquivo do ASTAH que corresponderá a implementação realizada, não podendo ser elaborada a leitura (entrada) de dados através de Janelas de Diálogo na aplicação (deverá existir uma janela **Swing** para o programa, correspondendo a janela base da aplicação, em que podem ser abertas outras janelas, de acordo com a interação planejada por cada estudante em sua proposta de solução, caso ache necessário várias janelas, mas tendo o controle sobre todas).

Todos os dados de entrada devem ser validados, sendo implementado o tratamento de exceções onde for necessária à correta leitura de informações essenciais ao funcionamento de sua solução, impedindo qualquer interrupção na execução, estando a solução proposta com pacotes e classes coerentes a uma implementação eficiente.

Apure todas as possibilidades de exceção em sua solução e as trate, pois o programa não poderá ser interrompido em nenhum momento até que o usuário final decida encerrá-lo. Outro dado importante é que nunca se saberá inicialmente quantas pessoas serão cadastradas por cada usuário.

Os nomes cadastrados nesta solução deverão ser apresentados sempre em maiúsculo e sem espaços em branco excessivos (antes ou depois do nome informado). É importante destacar que cada pessoa NÃO poderá ser instanciada sem argumentos em seu método construtor. Assim, analise o problema proposto e elabore uma solução adequada a todo conteúdo estudado até o momento em POO (Programação Orientada a Objetos).

Observações sobre o Material a ser Entregue para Avaliação

- Este trabalho é individual e cada estudante deverá entregar um programa orientado a objeto em JAVA, diretamente no ambiente MOODLE da disciplina, assim como a especificação (documentação) deste projeto (criar pasta **modelagem** no projeto Eclipse e colocar o arquivo ASTAH com dos dois diagramas dentro desta pasta para entrega correta, conforme tem sido feito nas atividades regulares da disciplina);
- A entrega de projeto plagiado, parcial ou totalmente copiado, resultará na nota zero para todos os envolvidos, independente de qualquer justificativa. Ressalto que o Plano de Ensino da disciplina define que para aprovação é necessária obter a nota mínima de 4,0 (quatro pontos) neste Trabalho Final, em que a cópia já o reprova diretamente, independente de outras notas;
- Sua implementação deverá ser feita somente com **Swing**, ao menos nos itens em que o enunciado exija outra forma de interação e apresentação para o usuário (enunciado indica explicitamente que poderá ou deverá mostrar na **CONSOLE**);
- O nome do projeto a ser entregue para avaliação deverá possuir a expressão **TF** seguida do número do tema e do primeiro e último nome do estudante, sem espaço em branco ou acento, além de sua matrícula. Por exemplo: a entrega pela aluna Ana Maria Braga, com matrícula 139874563, que elaborou o tema 1 teria o nome do projeto JAVA **TF1_AnaBraga**, enquanto que o arquivo físico e compactado, somente no **formato ZIP**, seria **TF1_AnaBraga_139874563.zip**;
- Atente a organização do código que será entregue e deverá estar estruturado de maneira eficiente com vários pacotes que sejam coerentes com sua implementação, além de respeitar o(s) padrão(ões) exigido(s) pela disciplina durante seu período letivo;
- O arquivo de documentação deverá estar no projeto JAVA, na pasta **modelagem** que você criará no Eclipse (na raiz da estrutura do projeto), com o mesmo nome do projeto, ou seja, **TF1_AnaBraga.asta** para o exemplo usado anteriormente.



TEMA 2: PANDEMIA DO CORONAVÍRUS

A pandemia decorrente do coronavírus tem demandado da Programação Orientada a Objetos (POO) a manipulação de vários dados e você está responsável por desenvolver uma aplicação que efetue o registro de pessoas e suas situações de saúde. Cada pessoa é cadastrada por um identificador inteiro e sequencial, que não pode ser repetido, e será sempre maior que a constante **MAIOR** de valor 100, além do nome completo da pessoa (tem que possuir mais que 2 caracteres, sem espaços excessivos no início ou fim e estar em **String mutável**) e a situação de saúde dela que pode ser:

T = contaminada em Tratamento;

F = contaminada Falecida;

C = contaminada Curada;

S = Sem contaminação.

Somente para as pessoas do sexo masculino deverá ser solicitada a idade em anos completos (zero é válido e indica bebê que não completou 1 ano ainda, mas o valor abaixo de zero ou acima de 150 serão inválidos, devendo essa idade máxima estar definida na constante **MAXIMO**), enquanto que somente para as pessoas do sexo feminino será cadastrado se ela já foi gestante, sendo validas somente as opções:

S = Sim;

N = Não;

T = não Tem certeza.

Os dados cadastrados de cada pessoa serão armazenados em uma única estrutura de dados dinâmica em JAVA que os colecionará na memória para processamento, mas serão registrados, de maneira persistente, no Sistema Gerenciador de Banco de Dados Relacional **MySQL** na versão 8 ou superior. O nome completo informado pelo usuário deverá ser cadastrado no **MySQL** exatamente como o usuário digitou, com relação aos caracteres maiúsculos e minúsculos, e respeitando as exigências de validação esclarecidas anteriormente neste enunciado.

Analise este problema e elabore um programa orientado a objeto, completamente em ambiente gráfico (JAVA *Swing*), que cadastre os dados sobre a pandemia, sendo validados todos os dados de entrada conforme as definições e restrições esclarecidas no início deste tema do enunciado. Os valores das constantes que correspondem aos limites do identificador único e a idade das pessoas deverão ser sempre usados para tais verificações (validações), caso contrário a lógica de tais implementações estarão incoerentes a existência e facilidades dessas constantes em sua solução, sendo mensurado diretamente em sua avaliação.

Todos os dados informados neste enunciado são obrigatórios e serão apresentados em forma de tabela somente quando o usuário informar a opção correspondente a saída de dados que ele deseje visualizar (itens **b** e **d** esclarecidos na sequência). No entanto, essa forma de tabela deverá ser apresentada também usando um recurso gráfico de JAVA



chamado **JTable** e não na console. O aspecto relacionado a pesquisa neste Trabalho Final acontecerá sobre o componente **JTable** que deverá ser usado, obrigatoriamente, na apresentação dessa forma de resultado tabelar para mostrar as pessoas cadastradas. Todas as *string* apresentadas no **JTable** deverão estar completamente em minúsculo, não podendo ser alterado o dado original informado pelo usuário que está armazenado no **MySQL**.

Sua solução deverá oferecer para o usuário uma forma de menu de opções ou botões que propiciará ao usuário final escolher entre as opções relacionadas a seguir, estando todas elas sempre disponíveis para o usuário acionar quando desejar, até que seja escolhida a opção **Sair**:

- a) **Novo** – cadastrar uma nova Pessoa conforme o desejo do usuário no programa, respeitando todas as validações de entrada de dados coerentes (indicadas anteriormente neste enunciado);
- b) **Lista** – mostrar todas as pessoas cadastradas de maneira tabelar no **JTable**, respeitando o formato tabelar com uma única linha de cabeçalho indicando todos os dados fornecidos corretamente nesta forma tabelar (relatório com todos os cadastros e todos os dados cadastrados respeitando a sequência das colunas de Identificador, Nome completo (todo em minúsculo), Situação de saúde por extenso, Idade e Gestante por extenso, estando ordenados pelo identificador de cada pessoa;
- c) **Mostra** – apresenta uma pessoa específica por meio de seu número identificador, em que todos os seus dados serão apresentados em uma **janela de diálogo**, sem nenhum ícone (figura do tipo de mensagem), estando cada dado cadastral em uma linha desta janela por extenso;
- d) **Pesquisa** – consulta entre todas as pessoas, por meio do fornecimento de qualquer parte de seu nome, em que todos os registros que satisfizerem este nome serão recuperados e apresentados em ordem alfabética do nome das pessoas, de maneira tabelar aproveitando o mesmo **JTable** anterior (item **b**), mas com a ordenação dos dados diferente. Sua solução também deverá mostrar ao usuário desta pesquisa a quantidade de registros que foram recuperados e mostrados no final dessa apresentação (na mesma janela do **JTable**);
- e) **Sair** – termina a execução do programa apresentando exatamente os dados indicados abaixo na **console**, mostrando sempre os números com no mínimo duas casas numéricas inteiras (confira no exemplo abaixo que deverá possuir um salto de 20 espaços à esquerda para dar a impressão de centralizado ao usuário, mas note que todas as linhas abaixo estão alinhadas à esquerda – começam na mesma coluna). A última linha que totaliza as quantidades apresentadas deverá estar em cor diferente na console para chamar a atenção do usuário.

```
39 = CONTAMINADOS CURADOS
21 = CONTAMINADOS EM TRATAMENTO
13 = CONTAMINADOS FALECIDOS
07 = HOMENS SEM CONTAMINAÇÃO
09 = MULHERES SEM CONTAMINAÇÃO
89 = TOTAL DE REGISTRO DE PESSOAS
```



Diante das características do problema apresentado deverá ser elaborada a especificação da solução que será desenvolvida, individualmente, por cada estudante, respeitando, exatamente, o modelo definido por seu professor, além do código que implementará tal solução. Esta solução só será aceita se for desenvolvida no ambiente gráfico JAVA (**Swing**) e possuir a modelagem usando o ASTAH com Diagrama de Casos de Uso e Diagrama de Classes completos da solução entregue que respeitará o padrão MVC e DAO.

Esta especificação/documentação possuirá um único arquivo do ASTAH que corresponderá a implementação realizada, não podendo ser elaborada a leitura (entrada) de dados através de Janelas de Diálogo na aplicação (deverá existir uma janela **Swing** para o programa, correspondendo a janela base da aplicação, em que podem ser abertas outras janelas, de acordo com a interação planejada por cada estudante em sua proposta de solução, caso ache necessário várias janelas, mas tendo o controle sobre todas).

Todos os dados de entrada devem ser validados, sendo implementado o tratamento de exceções onde for necessária à correta leitura de informações essenciais ao funcionamento de sua solução, impedindo qualquer interrupção na execução, estando a solução proposta com pacotes e classes coerentes a uma implementação eficiente.

Apure todas as possibilidades de exceção em sua solução e as trate, pois o programa não poderá ser interrompido em nenhum momento até que o usuário final decida finalizá-lo. Outro dado importante é que nunca se saberá inicialmente quantas pessoas serão cadastradas por cada usuário.

Os nomes cadastrados nesta solução deverão ser apresentados sempre em minúsculo e sem espaços em branco excessivos (antes ou depois do nome informado). É importante destacar que cada pessoa **NÃO** poderá ser instanciada sem argumentos em seu método construtor. Assim, analise o problema proposto e elabore uma solução adequada a todo conteúdo estudado até o momento em POO.

Observações sobre o Material a ser Entregue para Avaliação

- Este trabalho é individual e cada estudante deverá entregar um programa orientado a objeto em JAVA, diretamente no ambiente MOODLE da disciplina, assim como a especificação (documentação) deste projeto (criar pasta **modelagem** no projeto Eclipse e colocar o arquivo ASTAH com dos dois diagramas dentro desta pasta para entrega correta, conforme tem sido feito nas atividades regulares da disciplina);
- A entrega de projeto plagiado, parcial ou totalmente copiado, resultará na nota zero para todos os envolvidos, independente de qualquer justificativa. Ressalto que o Plano de Ensino da disciplina define que para aprovação é necessária obter a nota mínima de 4,0 (quatro pontos) neste Trabalho Final, em que a cópia já o reprova diretamente, independente de outras notas;
- Sua implementação deverá ser feita somente com **Swing**, ao menos nos itens em que o enunciado exija outra forma de interação e apresentação para o usuário (enunciado indica explicitamente que poderá ou deverá mostrar na **CONSOLE**);
- O nome do projeto a ser entregue para avaliação deverá possuir a expressão **TF** seguida do número do tema e do primeiro e último nome do estudante, sem espaço em branco ou acento, além de sua matrícula. Por exemplo: a entrega pela aluna Ana Maria Braga, com matrícula 139874563, que elaborou o tema 2 teria o nome do projeto JAVA **TF2_AnaBraga**, enquanto que o arquivo físico e compactado, somente no **formato ZIP**, seria **TF2_AnaBraga_139874563.zip**;



Universidade de Brasília – UnB (FGA)

Engenharia de Software – Trabalho Final de OO

- Atente a organização do código que será entregue e deverá estar estruturado de maneira eficiente com vários pacotes que sejam coerentes com sua implementação, além de respeitar o(s) padrão(ões) exigido(s) pela disciplina durante seu período letivo;
- O arquivo de documentação deverá estar no projeto JAVA, na pasta **modelagem** que você criará no Eclipse (na raiz da estrutura do projeto), com o mesmo nome do projeto, ou seja, **TF2_AnaBraga.asta** para o exemplo usado anteriormente.