

Prova 1

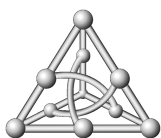
Linguagem de Programação Orientada a Objetos — Turma P01 — 2020/01

Instruções para a realização da prova:

1. A prova é **INDIVIDUAL** e contém **2 questões**, totalizando **10,5 pontos**;
2. **Não tente plagiar a prova do(a) seu(sua) colega**, você pode prejudicar você e seu(sua) colega;
3. **Não é necessário verificar a entrada**; isto é, se seu programa solicita que o(a) usuário(a) informe um número inteiro e o(a) usuário(a) informa uma letra ou qualquer outra coisa diferente de um número, seu programa pode ter qualquer comportamento inesperado;
4. **Utilize apenas o que foi ensinado em aula**. O uso de qualquer estrutura de programação ou estrutura de dados que não foi ensinada em sala de aula anulará a sua questão.
5. A prova tem duração de **5 horas e 30 minutos**. Fiquem tranquilos que, a princípio, essa avaliação foi **dimensionada para 3 horas**. As 2 horas e 30 minutos a mais correspondem a um tempo extra para evitar qualquer tipo de contratempo de envio, bem como para permitir que vocês a realizem com mais tranquilidade;
6. **Instruções** em relação ao formato dos arquivos de cada questão estarão na **descrição da própria questão**;
7. Após finalizar a prova, compacte todos os arquivos e pastas em um único arquivo com extensão **.ZIP** e **envie via AVA** na tarefa Prova 1;
8. O AVA ficará aberto para recebimento da avaliação das 7hs às 11h30. Após esse horário, **NÃO SERÁ POSSÍVEL ENVIAR ARQUIVOS**. Portanto, se programe para enviar os arquivos com pelo menos 15 minutos de antecedência;
9. Você pode submeter a prova quantas vezes achar necessário, apenas o último envio contará;
10. Qualquer dúvida em relação à prova deve ser escrita no grupo público do WhatsApp da turma, cujo link é o seguinte:
<https://chat.whatsapp.com/HGcUOfYVJblEo2owUbQj2K>
11. Apenas mensagens públicas direcionadas a esse grupo serão respondidas durante a prova. Ou seja, o professor não responderá e-mails ou mensagens privadas durante o horário da prova.

Boa prova!

Bem-aventurados os pacificadores, pois serão chamados filhos de Deus. Mateus 5:9

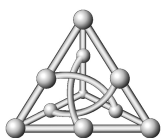


Questões

1. [3,00 pontos]

O objetivo desta questão é avaliar a sua compreensão em relação aos conceitos de classes e objetos. Para isso, considere as seguintes classes escritas em Java:

```
1  public class Pais {
2
3      private String nome;
4      private int populacao;
5      private Pais[] fronteiras;
6      private int quantidadeAtualDeFronteiras;
7
8      public Pais(String nome, int populacao) {
9          setNome(nome);
10         setPopulacao(populacao);
11         quantidadeAtualDeFronteiras = 0;
12     }
13
14     public void adicionaFronteira(Pais pais) {
15         if(fronteiras == null)
16             fronteiras = new Pais[3];
17
18         if(quantidadeAtualDeFronteiras >= fronteiras.length)
19             System.out.println("Impossível adicionar uma nova fronteira!");
20         else {
21             fronteiras[quantidadeAtualDeFronteiras] = pais;
22             quantidadeAtualDeFronteiras++;
23         }
24     }
25 }
```



```
1  public class TestePais {  
2      public static void main(String[] args) {  
3          Pais pais1, pais2;  
4          pais1 = new Pais("Brasil", 209000000);  
5          pais2 = new Pais("Uruguai", 3449000);  
6  
7          pais1.adicionaFronteira(uruguai);  
8  
9      }  
10 }
```

Com base nos códigos apresentados acima, responda os itens abaixo:

- [1,00 ponto]** Represente um **POSSÍVEL** estado das variáveis da classe *TestePais* na memória RAM **LOGO APÓS** a execução da linha 3 e **ANTES** da execução da linha 4.
- [1,00 ponto]** Represente um **POSSÍVEL** estado das variáveis da classe *TestePais* na memória RAM **LOGO APÓS** a execução da linha 5 e **ANTES** da execução da linha 7.
- [1,00 ponto]** Represente um **POSSÍVEL** estado das variáveis da classe *TestePais* na memória RAM **LOGO APÓS** a execução da linha 7.

Por questões de simplicidade, assuma os seguintes fatos nesse exercício:

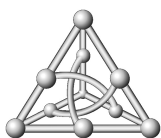
- Uma variável que armazena um endereço (ponteiro) ocupa 1 posição de memória;
- Uma variável do tipo *int* ocupa 1 posição de memória;
- O tipo *String* é um tipo **primitivo** e ocupa 1 posição de memória.

Para resolver esse exercício, siga o modelo de memória RAM disponibilizado no seguinte link do Google Drive: <http://lnk.ufms.br/cLBK8> . Acesse esse link e, para cada item a ser respondido, utilize uma cópia desta planilha. A resposta desse exercício deve conter 3 arquivos no formato PDF, um para cada item solicitado, onde cada um deles deve seguir o modelo disponibilizado.

Observação 1: para criar uma cópia de uma planilha no Google Drive, clique em Arquivo -> Fazer uma cópia.

Observação 2: para exportar uma planilha para o formato PDF, clique em Arquivo -> Fazer o download -> Documento pdf (.pdf).

Observação 3: caso você opte por não utilizar o Google Drive, você pode utilizar a ferramenta de sua preferência para responder o exercício, desde que o modelo e formato dos arquivos de resposta sejam equivalentes.



2. [7,50 pontos] O objetivo deste exercício é avaliar a sua capacidade de abstrair um problema e transformá-lo em um sistema orientado a objetos, em linguagem Java, que utilize princípios de encapsulamento.

Para isso, considere a seguinte solicitação de um cliente.

Olá, meu nome é Jiar Balsonoro, sou enfermeiro de um importante hospital de São Paulo e, apesar de acreditar que o covid-19 não me traria problemas de saúde (tenho histórico de atleta), ando muito preocupado com a saúde pública do nosso País.

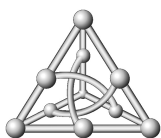
A quantidade de pacientes com covid-19 que necessitam de UTI tem aumentado significativamente em diversos estados do Brasil, e essa situação pode levar à dizimação de centenas de vidas, dado que nosso País não dispõe de uma quantidade de UTIs suficiente para atender à demanda projetada para o próximo estágio epidemiológico dessa doença.

Com o objetivo de auxiliar os profissionais de saúde a entenderem melhor a situação dos pacientes com covid-19 que estão na fila de espera por uma UTI, seria muito importante se a tecnologia nos auxiliasse a registrar todos esses pacientes e seu histórico médico.

Em relação a um paciente, é importante guardar seu nome e idade. Gostaríamos também de armazenar a lista das suas 10 principais comorbidades (fatores/doenças que podem agravar seu estado de saúde, como por exemplo, diabetes). Para cada comorbidade, é importante armazenar o nome (diabetes, hipertensão, obesidade, baixa imunidade, asma, câncer, etc.) e há quantos anos o paciente convive com essa comorbidade.

Precisamos de um sistema que basicamente registrará todos os pacientes que estão na fila de espera por uma UTI e permitirá a emissão de alguns relatórios. Nesse sistema deve ser possível:

- 1 - Cadastrar um paciente juntamente com sua lista de comorbidades (caso haja);*
 - 2 - Listar todos os pacientes e todas as suas informações;*
 - 3 - Listar todos os pacientes com uma determinada comorbidade. Para isso, o usuário precisa informar o nome da comorbidade;*
 - 4 - Listar os pacientes sem comorbidades.*
-



Crie um sistema orientado a objetos capaz de atender às demandas solicitadas pelo enfermeiro Jiar. Para isso, considere as seguintes informações técnicas:

- Seu sistema deve utilizar todos os princípios de encapsulamento vistos em sala de aula. Como haverão poucos *getters* e *setters*, implemente-os de maneira completa e condizente com a realidade dos dados apresentados;
- Para a leitura de informações do teclado, utilize um objeto da classe *Leitor.java* (vide exemplo na classe *TesteLeitor.java*), disponibilizado juntamente com o arquivo de prova;
- Não há como prever o tamanho da lista de pacientes que serão cadastrados. Isso te remete à criação de uma classe capaz de gerenciar uma lista 'infinita' de pacientes, que permitirá as operações mais básicas possíveis para que essa estrutura atenda às necessidades do exercício. Se me permite uma dica, faça isso por último! Ou seja: primeiro imagine que há uma lista finita de pacientes e, após isso estar funcional, migre seu programa para permitir uma lista 'infinita' de pacientes. Puxa, falei demais... agora já foi.

Nesse exercício você pode utilizar o editor de texto ou IDE de sua preferência. Ao final do exercício, coloque a pasta principal do seu projeto com todas as classes no arquivo final da prova.

Lembrem-se: vocês serão avaliados, principalmente, com base na sua capacidade de utilizar de maneira efetiva a orientação a objetos e seus princípios básicos na modelagem e implementação de um problema.