

GCC178 – Práticas de Programação Orientada a Objetos

Exercício 1

O Teste de Tolerância à Glicose é um dos testes realizados por pacientes para determinar se estão diabéticos ou não. Nesse teste, deve-se tomar um líquido que contém glicose e, em intervalos de tempo pré-definidos, uma amostra de sangue é colhida para avaliar a taxa de glicemia do paciente (taxa de “açúcar” no sangue). O resultado médio das amostras define o diagnóstico do paciente, que pode ser:

- **Glicemia normal:** se resultado for menor do que 140 mg/dl;
- **Risco de diabetes:** se resultado ficar entre 140 e 199 mg/dl;
- **Diabetes Mellitus:** se resultado for maior ou igual a 200 mg/dl.

Implemente em Java um programa que receba os dados do paciente e dos exames realizados pelo mesmo e retorne o diagnóstico. Para isso, crie uma classe denominada **Paciente** que contenha um atributo (**nome**) do tipo *String* para armazenar o nome do paciente e um vetor (**exames**) para armazenar *n* valores (do tipo *double*) de resultado da análise das amostras de sangue coletadas. Essa classe deve ter um único construtor responsável por inicializar o nome do paciente e por definir o tamanho (*n*) do vetor **exames**. A classe **Paciente** também deve ter um método para cada uma das seguintes tarefas:

- Retornar o nome do paciente.
- Retornar o resultado da análise de uma amostra de sangue específica (o número da amostra deve ser passado por parâmetro).
- Retornar a média dos resultados das análises das amostras de sangue.
- Retornar o diagnóstico do paciente.
- Inserir no vetor **exames** o resultado da análise de uma amostra de sangue coletada, cujo valor deve ser o único parâmetro desse método.

Vale observar que nenhum método da classe **Paciente** deve imprimir dados na tela.

Além disso, implemente uma classe denominada **TesteDiabetes** que possa ser executada (contenha o método *main*) e dentro dela exista o código que é necessário para instanciar um objeto da classe **Paciente** e chamar os métodos necessários para produzir uma saída na tela como exemplificado a seguir:

```
##### RELATÓRIO FINAL #####  
Resultado médio = 138,766667  
Diagnóstico: Glicemia normal
```

Para instanciar um objeto da classe **Paciente**, a classe **TesteDiabetes** deve conter o código responsável por solicitar ao usuário o nome do paciente, a quantidade de amostras de sangue coletadas e os resultados de cada uma das análises realizadas a partir das amostras de sangue, assim como demonstrado no exemplo a seguir:

```
Digite o nome do paciente: Luiz Henrique  
Entre com a quantidade de amostras coletadas: 3  
Insira o resultado do teste 1: 134,6  
Insira o resultado do teste 2: 147,2  
Insira o resultado do teste 3: 134,5
```

Exercício 2

ETAPA 1

Este exercício se refere a um programa para ajudar alunos da escola primária no aprendizado da multiplicação. Nesta primeira etapa, crie uma classe **Questao** que atenda às necessidades listadas a seguir. Lembre-se de usar adequadamente a visibilidade (public/private) dos atributos e métodos.

- As questões devem possuir um enunciado e um gabarito, ou seja, a sua resposta correta.
- O construtor dessa classe deve gerar o enunciado do exercício, o qual sempre tem o formato “**Quanto é** *num1* * *num2*?”, sendo os dados que compõem esse enunciado (*num1* e *num2*) devem ser gerados aleatoriamente (entre 0 e 9) utilizando-se o método **nextInt** da classe **Random** do Java (pacote **java.util**).¹

Antes da declaração da classe **Questao**, não se esqueça da **declaração import** (import java.util.Random;) para fazer uso da classe **Random**.

- Deve existir um método para ser possível consultar o enunciado.
- Deve existir um método que receba por parâmetro a resposta do aluno e retorne um **booleano** indicando se o aluno acertou a questão ou não.

Crie uma segunda classe denominada **Prova**, que atenda aos seguintes requisitos:

- Cada prova deve conter uma única questão.
- Deve existir um método denominado **aplicar** que, quando chamado, apresente uma questão para o usuário e, a partir da resposta do mesmo, verifique a sua correção e imprima na tela uma das mensagens a seguir:
 - **Muito bem, você acertou!**
 - **Infelizmente você errou!**

Para testar o seu programa, crie uma terceira classe denominada **ProvaTeste** que contenha o método **main**. No método **main**, apenas instancie um objeto da classe **Prova** e chame o método **aplicar**.

Teste sua implementação!

ETAPA 2

Altere o método **aplicar** da classe **Prova** de modo a permitir que o usuário tenha mais uma chance de responder corretamente a questão se a primeira resposta dele estiver incorreta. Nesse caso, deve ser **adicionada** a seguinte mensagem para o usuário: "Você ganhou mais uma chance! Digite outra resposta para a questão: ". Além disso, após verificar a correção da(s) resposta(s), imprima na tela uma das mensagens a seguir (onde *x* corresponde ao número de vezes que o usuário respondeu aquela questão):

- **Você tentou *x* vez(es) e acertou a questão.**
- **Você tentou *x* vez(es) e errou a questão.**

Teste suas alterações!

ETAPA 3

Altere sua classe **Questao** inserindo um atributo que corresponda ao número da questão (por exemplo, *idQuestao*). Além disso, crie um método para consultar esse atributo.

¹ Criando um objeto Random: **Random** numeroAleatorio = new **Random**();
Obtendo um valor aleatório **int** entre 0 e y: **int** valor = *numeroAleatorio.nextInt*(y);

Queremos tratar a numeração dos exercícios de forma que ela seja automática (sem precisar ser informada pelo usuário). Para isso, crie um atributo **estático** na classe **Questao** que armazena o número da última questão criada (inicialmente tem o valor zero). No **construtor** da classe **Questao** esse atributo deve ser incrementado e seu valor usado como o número do exercício que está sendo criado (ao invés de ser recebido por parâmetro).

Agora, ao apresentar uma questão para o usuário, antes de seu enunciado, exiba na tela uma mensagem com o número da questão (para que o usuário saiba qual questão ele está respondendo), como por exemplo, "Questão 1" .

O próximo passo é alterar a implementação da classe **Prova** para que uma prova seja composta por várias questões (crie um vetor de questões). Desse modo, cada vez que um objeto da classe **Prova** for instanciado, uma prova contendo 5 questões deve ser criada. Portanto, você terá que alterar o **construtor** da classe **Prova** para que isso aconteça. Além disso, a partir de agora, cada vez que o método **aplicar** (classe **Prova**) for chamado, devem ser apresentadas para o usuário cada uma das 5 questões da prova. Todas as mensagens de acerto ou erro de resposta implementadas nos exercícios anteriores devem continuar sendo exibidas para cada questão da prova.

Teste suas alterações!

ETAPA 4

Após os alunos responderem todas as questões da prova, queremos ter a opção de apresentar um relatório com o histórico de seu desempenho ao longo da prova, como no exemplo a seguir:

Questão	Situação	Número de Tentativas
1	acertou	1
2	acertou	1
3	errou	2
4	acertou	1
5	acertou	2

Para isso, crie uma classe que representa cada linha do histórico. Vamos chamá-la de **Correcao**. Essa classe deve ter atributos para os campos mostrados no exemplo acima e métodos para consultá-los. O construtor da classe **Correcao** deve receber por parâmetros o valor de todos os atributos e a classe não deve ter nenhum método de alteração dos atributos.

Sugiro que você crie uma classe temporária com um método **main** apenas para testar essa nova classe (Correcao).

Agora vamos tratar as correções na classe **Prova**. Ela deve passar a possuir um histórico da correção de cada questão. Para isso, crie um atributo para guardar todas as correções da prova, ou seja, uma coleção de objetos da classe **Correcao**. Para isso, vamos novamente utilizar um vetor. Em seguida, altere o método **aplicar** (classe **Prova**) para que a cada nova correção realizada, ela seja adicionada ao vetor.

Feito isso, crie um novo método na classe **Prova** denominado **gerarRelatorio**. Quando esse método for chamado, um relatório (histórico), como o mostrado no exemplo acima, deve ser exibido na tela. A chamada a esse método deve ser feita na classe **ProvaTeste**, logo após a chamada ao método **aplicar**.

Teste suas alterações!