

Universidade Federal do Rio Grande do Norte
Instituto Metropole Digital
IMD0040 - Linguagem de Programação 2
Lista de exercícios

- Esta lista de exercício é composta por 4 questões. Sendo uma fácil, uma média, uma difícil e uma de desafio.
- As questões valem, respectivamente, 20%, 30%, 50% e 100%.
- O aluno poderá escolher entre realizar as 3 primeiras e somar 100% ou apenas o desafio para conseguir os 100%.
- **NÃO HAVERÁ ACUMULO**, portanto, se decidir fazer as 4, não ganhará mais por isso (mas terá aprendido muito mais).
- Não se assustem com o tamanho dos textos.

Questão Fácil (20%) - Heater + jUNIT

A tarefa

Usando como base o exercício do aquecedor (feito na primeira aula de exercícios), você precisa testar seu sistema. Para isso, crie uma classe de teste JUnit e desenvolva métodos para testar todos os métodos da sua classe Heater.

Implementação

Lembre-se que seu aquecedor possui limites em relação ao quanto ele pode aquecer ou diminuir de acordo com o MAX e o MIN. Faça os casos de teste pensando nisso: crie casos positivos e casos negativos.

Obs: Este exercício é muito fácil, portanto caso você tenha perdido seu Heater, baixe o PDF da Aula 3 e refaça-o, não vai demorar nada.

Questão Média (30%) - Técnicas de Teste

A tarefa

Baixe o projeto Bricks do Sigaa. Teste-o. Há pelo menos 4 erros neste projeto. Veja se você consegue localizá-los.

ATENÇÃO: você deverá localizar esses erros sem olhar o código-fonte do projeto em questão.

ATENÇÃO 2: A resposta para esta questão deverá ser submetida acrescentada de um arquivo de texto (.txt) com suas respostas.

A teoria

Abra um bloco de notas e defina os seguintes termos:

Teste positivo.
Teste negativo.
Assertiva.
Fixture.
Verificação passo a passo.

Tente exemplificar em quais casos se deve utilizar cada um deles.

A prática

Quais técnicas você utilizou para localizar os erros? Qual delas foi a mais útil? Anote estas respostas em um bloco de notas.

Questão Difícil (50%) - Calculadora

A tarefa

Este desafio consiste no exercício 6.35 do livro do BlueJ. Você deverá baixar a classe calculador-gui do Sigaa. Esta classe contém a interface da calculadora que você deverá fazer funcionar.

Implementação

Faça alterações, adições ou qualquer outra coisa que ache necessário para fazer com que a calculadora funcione. Ela deve fazer operações de forma correta e se comunicar de maneira coerente com a classe GUI (user interface).

O desafio

Além de funcionar com todas as operações e exibições na tela, crie um teste junit para testar automaticamente sua calculadora.

1. Crie uma classe para representar uma pessoa, com os atributos privados de nome, data de nascimento e altura. Crie os métodos públicos necessários para sets e gets e também um método para imprimir todos dados de uma pessoa. Crie um método para calcular a idade da pessoa. Crie uma classe Agenda que pode armazenar 10 pessoas e seja capaz de realizar as seguintes operações:

```
void armazenaPessoa(String nome, int idade, float altura);  
void removePessoa(String nome);  
int buscaPessoa(String nome); // informa em que posição da agenda está a  
pessoa  
void imprimeAgenda(); // imprime os dados de todas as pessoas da agenda  
void imprimePessoa(int index); // imprime os dados da pessoa que está na  
posição "i" da agenda.
```

Questão Desafio (100%) - Polinômios.

Considere um polinômio de grau n:

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x^1 + a_0 x^0$$

Escreva uma classe **Termo** que represente um termo deste polinômio com os seguintes métodos:

<i>construtor</i>	Recebe dois parâmetros : a_i e i , e cria um objeto em memória na forma $a_i x^i$.
<i>insere</i>	Recebe um objeto da classe Termo e substitui os valores $a_i x^i$ do termo corrente por aqueles do termo recebido como parâmetro.
<i>calcula</i>	Recebe um valor de x como parâmetro e retorna o valor do termo calculado.

Escreva uma classe **Polinomio** que representa polinômio completo na forma de uma sequência de objetos da classe **Termo**, com os seguintes métodos:

<i>construtor</i>	Recebe um objeto da classe Termo e cria um polinômio em memória na forma:
-------------------	---

	$P(x) = a_i x^i$.
insere	Recebe um objeto da classe Termo e adiciona o termo $a_i x^i$ ao polinômio recebido como parâmetro. O polinômio pode ter um termo $a_q x^q$ cujo valor de q seja igual a i, neste caso o método deve unificar ambos em um único termo.
calcula	Recebe como parâmetro outro objeto da classe Polinomio e realiza a fusão do polinômio recebido como parâmetro com o polinômio corrente.

Acrescente os métodos que achar necessários nas classes solicitadas. Utilize alguma coleção Java para armazenar os termos na classe Polinomio.

Em seguida, escreva casos de teste JUnit para sua classe polinômio. Crie pelo menos um teste positivo e um teste negativo para cada um dos métodos solicitados.

Obs: DOCUMENTE SEU CÓDIGO. Javadoc não é apenas opcional, ele é essencial para manter a organização e a transparência do seu código. Código não documentado terá redução na pontuação.