|  | **Disciplina**: Laboratório de Programação 2  **Professor**: Sidney Nogueira |
| --- | --- |

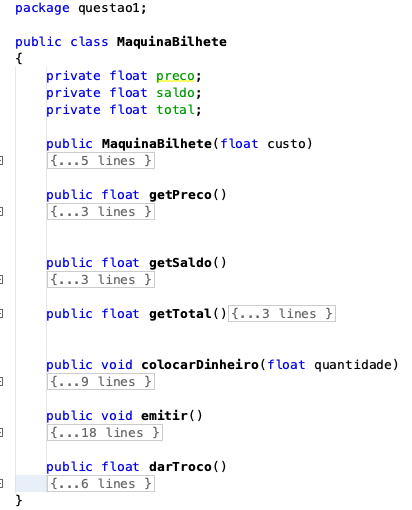
Lembrete:

* Formate o seu projeto de acordo com este guia <https://docs.google.com/document/d/1gfR_gt7jW0t1kXysoytQfUI0evximpDy8AubFLDU52E/edit>

Quando concluir as duas questões a seguir, faça um zip (RAR não é aceito) do projeto e envie o zip na sala de aula. Os pacotes do projeto devem ser chamar questao1 e questao2, conforme é pedido no guia acima.

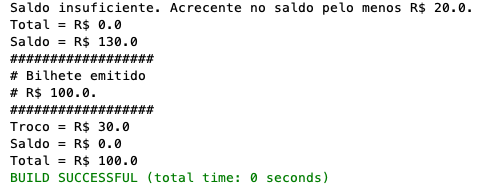
**Lista L3 - Primeiras classes**

**Questão 1 -** Baixe este projeto [aqui](https://drive.google.com/file/d/1_gTu05AQX-iUNYB0sytRY3cK2IEdSoAD/view?usp=sharing) e modifique o código da classe MaquinaBilhete conforme é pedido.

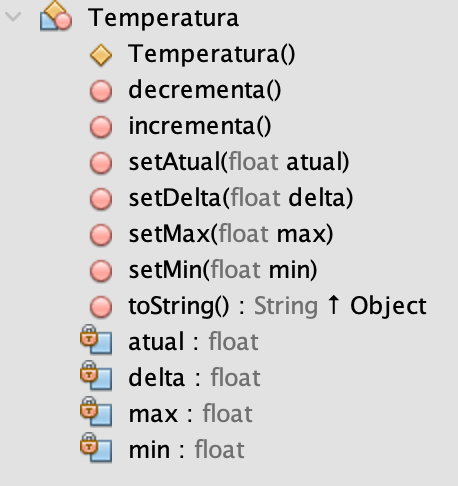


1. O corpo do construtor deve inicializar o atributo preco com o valor de custo, e os atributos saldo e total devem ter valor 0.
2. Os métodos getPreco, getSaldo e getTotal devem retornar o valor dos respectivos atributos.
3. O método colocarDinheiro deve atualizar o saldo somando com o valor do parâmetro quantidade;
4. O método emitir deve emitir um bilhete, se o saldo é maior ou igual a preco do bilhete. Neste caso, o valor de total deve ser incrementado pelo valor do preco do bilhete emitido e o saldo decrementado. Se valor for menor, deve imprimir mensagem indicando quanto deve ser adicionado ao saldo, de forma que seja igual ao preço do bilhete. O formato das mensagens que serão impressas é mostrada na figura a seguir.
5. O método darTroco deve retornar o valor que restou no saldo e atribuir 0 ao saldo.

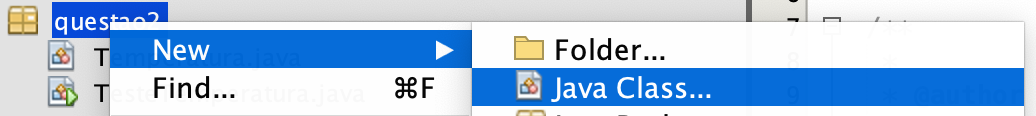
Execute a classe TesteMaquina, que está no mesmo projeto, de forma que a saída seja parecida com a saída a seguir. O código da TesteMaquina não deve ser modificado, apenas o código da classe MaquinaBilhete pode ser modificado.



**Questão 2 -** No pacote questao2 do mesmo projeto, crie uma nova classe chamada Temperatura. Que após ser implementada, deve ser mostrada como segue pelo Netbeans.



Para criar uma nova classe clique no pacote questao2 e selecione New -> Java Class



A classe Temperatura deve ter:

1. Atributos privativos atual, delta, max e min do tipo float.
   1. atual : representa o valor atual da temperatura;
   2. delta : o quanto que a temperatura é aumentada ou diminuída (este valor não pode ser negativo);
   3. max : valor máximo para a temperatura (atual não pode ser maior);
   4. min : valor mínimo para a temperatura (atual não pode ser menor);
2. Um construtor sem parâmetros que inicializa os atributos para 0, 5, 50 e -50;
3. Método toString que retorna uma string que mostra o valor de todos os atributos. A string retornada deve seguir o padrão a seguir:



1. Método decrementa (incrementa), que diminui (aumenta) o valor atual da temperatura pelo valor do delta. Se o valor atual for menor (maior) do que o valor mínimo (máximo), então o valor atual deve se tornar igual ao mínimo (máximo).
2. Método setAtual, que recebe um valor como parâmetro e atualiza a temperatura atual para ser igual ao parâmetro. Se o valor do parâmetro não estiver dentro do intervalo fechado de valores [min,max] (inclui min e max), o valor de atual não é modificado.
3. Método setDelta, que recebe como parâmetro o novo valor para ser atribuído ao delta. Se o valor for negativo, o valor de delta não é alterado.
4. Método setMax (setMin), que recebe como parâmetro o novo valor para max (min). O valor para max (min) só é atualizado se o valor recebido foi maior (menor) do que o mínimo (máximo).

Depois de implementar Temperatura, descomente o código da classe TesteTemperatura e rode esta última classe. A saída deve ser parecida com a saída que segue.

