



Lista 6: Coleções Java, Herança e Testes de unidade

Professor: Emerson Ribeiro de Mello

<http://docente.ifsc.edu.br/mello/poo>



Nota:

- Antes de iniciar, sincronize seu repositório local com o seu repositório remoto “Listas de exercícios” do Github. É possível que o professor tenha feito um commit por lá após a entrega da lista de exercício anterior;
- Crie um diretório com o nome `lista-06` na raiz do repositório e dentro deste, crie um subdiretório para cada um dos exercícios abaixo;
- Edite o arquivo `Readme.md` que está na raiz do repositório e adicione um novo item com o nome `lista-06` e este deverá ter um link o qual deverá apontar para o subdiretório com o nome `lista-06`.

1 Aplicativo para desenho vetorial

Faça um diagrama de classes UML (salve o arquivo .png) para representar as formas apresentadas na Figura 1. Deve-se indicar as associações entre as classes (agregação, composição e herança). Um ponto é representado no plano cartesiano pelas coordenadas X e Y. Para as formas que possuem área, deve-se criar um método que retorna o valor da área.

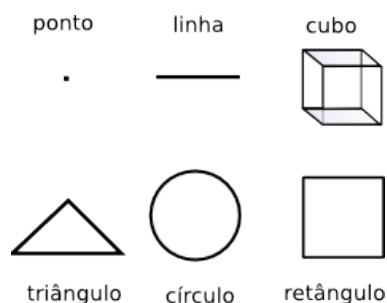


Figura 1: Formas geométricas que poderão ser desenhadas

Crie um projeto Java com gradle e nesse implemente todas as classes da modelagem que fez. Todas classes devem possuir um método `desenhar` que irá imprimir no dispositivo de saída uma mensagem como: Desenhando um ponto em X, Y. Faça um aplicativo Java (classe com método `main`) e instancie alguns objetos das classes que foram modeladas e invoque o método `desenhar` desses objetos.

2 Gerenciador de bibliografias

Um arquivo [BibTeX](#) contém entradas diferentes para representar os diversos tipos de publicação. Cada tipo possui um conjunto de atributos opcionais e um conjunto de atributos obrigatórios. Por exemplo, o tipo `book` tem como atributos obrigatórios: `author`, `publisher`, `title` e `year`. Já como atributos opcionais tem-se: `address`, `edition`, `month`, `number`, `series` e `volume`.

Na Tabela 1 é apresentado um subconjunto de todos os tipos previstos para um arquivo BibTeX, bem como a lista com um subconjunto de todos os atributos obrigatórios e opcionais

para cada um deles. O símbolo **o** é para atributos opcionais e **+** para atributos obrigatórios. Em branco significa que o atributo não pertence àquele tipo de publicação.

Tabela 1: Tipos de entradas em um arquivo BibTeX

Atributos	Tipos de entrada				
	article	book	masterthesis	misc	techreport
address		o	o	o	o
author	+	+	+	o	+
edition		o			
editor		o			
institution					+
journal	+				
month	o	o	o	o	o
number	o	o			
pages	o				
publisher		+			
series		o			
title	+	+	+	+	+
volume	o	o			
year	+	+	+	o	+

Cada entrada no arquivo BibTeX é precedida pelo símbolo @ e tem seu bloco delimitado pelos símbolos { e }. Abaixo é apresentado um exemplo para a entrada do tipo book:

```

1 @book{lamport1994latex,
2   title={LATEX: a document preparation system: user's guide and reference manual},
3   author={Lamport, Leslie},
4   year={1994},
5   publisher={Addison-wesley}
6 }
```

Note que cada entrada é composta por um conjunto pares *chave=valor*. Os valores devem ser colocados dentro de um bloco delimitado pelos símbolos { e }. O símbolo , (vírgula) é usado para separar os pares *chave=valor*.

Cada entrada no BibTeX precisa ter um identificador único. Esse identificador é colocado logo quando se abre o bloco da entrada. No exemplo acima, o identificador da entrada é *lamport1994latex*. Sendo assim, nesse arquivo BibTeX não pode-se ter outra entrada que possua esse mesmo identificador.

Faça uma modelagem em UML (salve o arquivo .png) – deve-se indicar as associações entre as classes (agregação, composição e herança) – para representar um aplicativo que permita a gestão de várias bases de bibliografia do usuário. Por exemplo, eu poderia usar o aplicativo para gerenciar uma base de bibliografia específica para um artigo que estou escrevendo e uma outra base de bibliografia específica para as notas de aulas da disciplina de Programação Orientada a Objetos.

Cada base bibliográfica será composta por publicações do tipo article, book, masterthesis, misc ou techreport. Cada tipo de publicação possui seu conjunto de atributos opcionais e obrigatórios, conforme apresentado na tabela da seção anterior. Deve-se respeitar as particularidades de cada tipo de publicação. Por exemplo, eu não poderia criar uma publicação do tipo article que contenha o atributo *series*.

Por fim, implemente em Java o modelo desenvolvido. O aplicativo a ser desenvolvido deverá atender pelo menos os seguintes pontos:

- Possuir funcionalidade que permita exportar uma base bibliográfica no formato BibTeX. No caso, deve-se gerar uma String que poderá ser impressa na tela;

- Possuir funcionalidade que permita criar, listar e remover bases bibliográficas. E assim, funcionalidade para permitir adicionar, listar e remover entradas em uma base bibliográfica;
- Fazer documentação no formato JavaDOC de todas as classes e dos principais métodos
- Fazer testes de unidade de forma a garantir que não seja possível criar uma entrada BibTeX que não seja válida e que permita criar entradas válidas. Também, deve criar testes de unidade para validar as operações de adicionar, listar e remover.



Atenção:

- **Para receber nota 10:** Os commits para essa entrega deverão ser encaminhados (push) para o Github Classroom até às 12:30 do dia **29/04/2020**;
- **Para receber nota 5:** Os commits deverão ser encaminhados (push) para o Github Classroom até o dia **30/04/2020**;
- Qualquer entrega após o dia **30/04/2020** não será contabilizada e receberá nota 0;
- Entrega parcial da lista receberá no máximo nota 1.