Exercícios POO

3 Desenvolvimento da aplicação ShapesCollection

3.1 Descrição do problema

O objectivo deste trabalho é o desenvolvimento de uma aplicação para manter uma colecção de formas geométricas. Cada fora geométrica tem um identificador único, para além do valor da abcissa e ordenada do centro. Vamos considerar apenas duas formas geométricas: círculos e rectângulos. Para os círculos é mantido o valor do raio. Enquanto para os rectângulos é mantido o valor da altura e largura. Neste exercício deve usar as interfaces ShapesCollection que representa uma colecção de formas geométricas; Shape que representa uma formas geométrica; e Iterator disponibilizadas no Moodle e apresentadas na Figura 1.

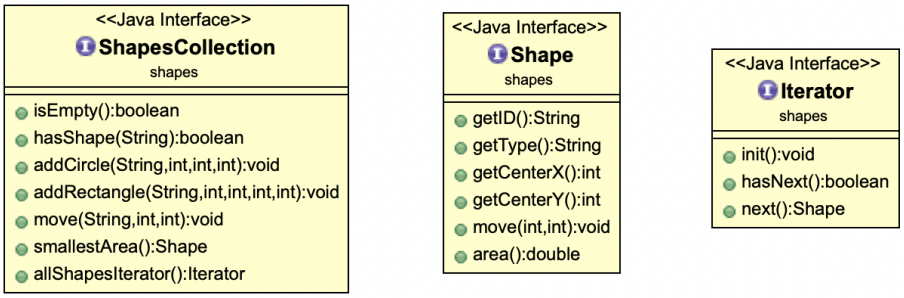

Figura 1: Interfaces da aplicação ShapesCollection

A aplicação deve então permitir:

1. Adicionar um circulo (comando `CIRCLE`). São fornecidos o identificador, o valor da abcissa e ordenada do centro, e o valor do raio. Em caso de sucesso o círculo é adicionado à colecção ("A new circle was added."). A operação falha se: (1) já existir uma forma geométrica com o identificador dado ("Identifier already exists.").

2. Adicionar um rectângulo (comando `RECTANGLE`). São fornecidos o identificador, o valor da abcissa e ordenada do centro, e o valor da altura e largura. Em caso de sucesso o rectângulo é adicionado à colecção ("A new rectangle was added."). A operação falha se: (1) já existir uma forma geométrica com o identificador dado ("Identifier already exists.").

3. Listar todas as formas geométricas (comando `LIST`). A listagem deve ser precedida por um cabeçalho com a mensagem ("All shapes in the collection:"), apresentado as formas geométricas, uma por linha pela ordem de introdução. Cada linha deve apresentar o identificador, o valor abcissa e ordenada do centro, seguido do tipo da forma ("CIRCLE") ou ("RECTANGLE"). A operação falha se: (1) não existirem formas geométricas na colecção ("Empty collection.").

4. Mover uma forma geométrica (comando `MOVE`). São fornecidos o identificador, seguido do novo valor da abcissa e ordenada. Em caso de sucesso, o centro da forma geométrica é alterado ("Shape was moved."). A operação falha se: (1) não existir uma forma geométrica com o identificador dado ("Identifier does not exist.").

5. Apresentar a forma geométrica da colecção com a menor área (comando `MINAREA`). Se existirem várias formas geométricas que satisfaçam este critério, deve ser apresentada a forma geométrica mais recente. Em caso de sucesso, deve ser apresentado o identificador, o valor da abcissa e ordenada do centro, seguido do tipo da forma ("CIRCLE") ou ("RECTANGLE"). A operação falha se: (1) não existirem formas geométricas na colecção ("Empty collection.").

6. Sair da aplicação (comando `EXIT`). A operação tem sempre sucesso ("Exiting...").

3.2 Exemplo de interacção com a aplicação

Desenvolva a sua aplicação para que esta garanta, pelo menos, o modelo de interacção ilustrado no exemplo seguinte (o caracter ↵ representa uma mudança de linha):

```
LIST↵
Empty collection.↵
↵
CIRCLE circle1 0 0 12↵
A new circle was added.↵
↵
CIRCLE circle1 4 2 12↵
Identifier already exists.↵
↵
RECTANGLE rectangle1 0 7 2 8↵
A new rectangle was added.↵
↵
RECTANGLE circle1 8 5 6 7↵
Identifier already exists.↵
↵
CIRCLE circle2 4 2 5↵
A new circle was added.↵
↵
CIRCLE circle3 -1 0 5↵
A new circle was added.↵
↵
RECTANGLE rectangle2 -1 4 2 78↵
A new rectangle was added.↵
↵
RECTANGLE rectangle3 6 7 4 9↵
A new rectangle was added.↵
↵
RECTANGLE rectangle3 -1 0 23 9↵
Identifier already exists.↵
↵
LIST↵
All shapes in the collection:↵
circle1 (0, 0) CIRCLE↵
rectangle1 (0, 7) RECTANGLE↵
circle2 (4, 2) CIRCLE↵
circle3 (-1, 0) CIRCLE↵
rectangle2 (-1, 4) RECTANGLE↵
rectangle3 (6, 7) RECTANGLE↵
↵
MOVE circle3 4 4↵
Shape was moved.↵
↵
MINAREA↵
rectangle1 (0, 7) RECTANGLE↵
↵
EXIT↵
Exiting...↵
```

4 Desenvolvimento

Desenvolva a sua aplicação de acordo com as seguintes fases:

1. Desenvolva o(s) interface(s) de suporte à aplicação. Utilize o conceito de polimorfismo. Desenhe um diagrama de classes e interfaces para ilustrar a proposta de modelação.

2. Implemente e teste a aplicação.
3. Submeta o código fonte da aplicação ao *Mooshak*.

Ficheiros de testes

Os testes do Mooshak verificam a implementação dos vários comandos:

- Ficheiro de teste: `1_in_test.txt` (40 pontos)
- Ficheiro de teste: `2_in_test.txt` (30 pontos)
- Ficheiro de teste: `3_in_test.txt` (30 pontos)