

Programação Orientada a Objectos

17/18

Exemplo de um relatório

Este é um exemplo do relatório que é esperado na meta 1 do trabalho prático. Para o exemplo considerou-se que o trabalho seria o exemplo do desenho/rectângulos dado nas aulas. O relatório real a elaborar seguirá a estrutura e temas que aqui se encontram, mudando apenas as coisas acerca das quais se está a falar.

O exercício usado neste exemplo, muito brevemente é o seguinte: Pretendia-se uma aplicação com o objectivo de representar um desenho formado por uma colecção de rectângulos com lados paralelos aos eixos coordenados. O desenho tem um nome. Relativamente a cada rectângulo conhecemos o seu canto superior esquerdo (um ponto) e as suas dimensões: largura (medida do lado paralelo ao eixo dos xx) e altura (medida do lado paralelo ao eixo dos yy). Os pontos de um plano são representados pelas suas coordenadas cartesianas x e y. Deve ser possível acrescentar um rectângulo ao desenho, obter a soma das áreas de todos os rectângulos, obter o conjunto de rectângulos cujo canto superior esquerdo esteja num dado ponto e obter uma representação textual do conteúdo do desenho.

De seguida apresenta-se o relatório (exemplo), propriamente dito.

Organização do código apresentado

1. Quais foram as classes consideradas na primeira versão da aplicação que foi testada?

A primeira versão tinha as seguintes classes e principais funcionalidades:

- Ponto
- Rectangulo
 - Obter a descrição textual dos seus atributos
- Desenho
 - Acrescentar rectângulo
 - Obter a descrição textual dos seus dados

Estas classes e funcionalidades foram testadas com uma simples função main.

2. Quais os conceitos/classe que identificou ao ler o enunciado?

Os conceitos identificados foram:

- Ponto
- Rectângulo
- Desenho
- Nome
- Canto superior esquerdo
- Largura
- Altura
- Coordenadas cartesianas x e y

3. Relativamente a duas das principais classes da aplicação, identifique em que classes ou partes do programa são criados, armazenados e destruídos os seus objectos.

Ponto: os objectos desta classe são criados, armazenados e destruídos na classe Rectangulo.

Rectângulo: os objectos desta classe são criados, armazenados e destruídos na classe Desenho

4. Indique um exemplo de uma responsabilidade atribuída a uma classe que esteja de acordo com a orientação dada acerca de *Encapsulamento*.

A responsabilidade “calcular a soma das áreas dos rectângulos” está atribuída à classe Desenho porque tem a colecção de rectângulos.

A responsabilidade “calcular a área de um rectângulo” está na classe Rectangulo porque tem os dados (largura e altura) para calcular a área.

5. De entre as classes que fez, escolha duas e justifique por que considera que são classes com objectivo focado, coeso e sem dispersão.

Classe Ponto: tem dados e responsabilidades relativos apenas às coordenadas cartesianas, como comparação de dois pontos e distância entre dois pontos.

Classe Rectangulo: está focada na localização, dimensões do rectângulo e cálculo da sua área.

6. Relativamente à aplicação entregue, quais as classes que considera com responsabilidades de interface com o utilizador e quais as que representam a lógica?

Responsabilidade de interface: Interacao

Responsabilidades da lógica da aplicação: Desenho, Rectangulo, Ponto

7. Identifique o primeiro objecto para além da camada de interacção com o utilizador que recebe e coordena uma funcionalidade de natureza lógica?

As ordens vindas da camada de interacção com o utilizador são recebidas e processadas por um objecto da classe Desenho.

8. A classe que representa a envolvente de toda a lógica executa em pormenor muitas funcionalidades, ou delega noutras classes? Indique um exemplo em que esta classe delega uma funcionalidade noutra classe.

A classe Desenho representa a envolvente de toda a lógica. Para calcular a soma das áreas dos rectângulos, delega o cálculo da área de cada rectângulo na classe Rectangulo.

9. Dê um exemplo de uma funcionalidade que varia conforme o tipo do objecto que a invoca. Indique em que classes e métodos está implementada esta funcionalidade. *(Não é necessário responder a esta pergunta na meta 1).*

O desenho era formado por um conjunto de rectângulos. Tornou-se necessário considerar outras figuras geométricas, como por exemplo círculos. O desenho deverá poder integrar rectângulos e círculos. Uma das funcionalidades do desenho é obter a soma das áreas das figuras geométricas que o integram.

A fórmula para o cálculo da área varia conforme o tipo de figura geométrica.

Esta funcionalidade é uma função abstracta da classe Figura:

```
class Figura {  
    // . . .  
    // calcula e retorna a area  
    virtual double calculaArea()const = 0;  
};
```

O cálculo da área é implementado nas classes concretas derivadas de Figura:

```
class RectanguloFigura: public Figura {  
    // . . .  
    // calcula e retorna a area  
    virtual double calculaArea()const;  
};
```

```
class CirculoFigura : public Figura {  
    // . . .  
    // calcula e retorna a area  
    virtual double calculaArea()const;  
};
```

10. Apresente as principais classes da aplicação através da seguinte informação:

Classe: Ponto

Responsabilidades:

Obter as coordenadas cartesianas de um ponto

Obter a distância entre dois pontos

Verificar se dois pontos são iguais

Colaborações: Rectangulo

Classe: Rectangulo

Responsabilidades:

- Obter a localização
- Obter as dimensões
- Calcular a área

Colaborações: Ponto, Desenho

Classe: Desenho

Responsabilidades:

- Representar uma colecção de rectângulos
- Gerir a colecção (acrescentar, eliminar, pesquisar)
- Obter a soma das áreas dos rectângulos
- Pesquisar os rectângulos com o mesmo canto superior esquerdo.

Colaborações: Rectangulo, Ponto

Classe: Interacao

Responsabilidades:

- Obter e mostrar informação acerca do desenho
- Mostrar as funcionalidades da aplicação
- Obter do utilizador as opções pretendidas
- Transmitir as ordens do utilizador ao desenho

Colaborações: Desenho

Funcionalidades implementadas

Na tabela abaixo está assinalado o grau de realização dos requisitos pedidos para esta meta.

(Esta secção já não se refere ao exemplo anterior, mas às componentes de avaliação da primeira meta do trabalho prático.)

Componente do trabalho	Realizado	Realizado parcialmente	Não realizado
Leitura e validação de comandos			
Leitura de comandos em ficheiro			
Criação de mundo ninho e formigas			
Visualização do mundo e conteúdo			
Formiga exploradora (move)			
Avanço de iteração			
Listagens (listamundo, listaninho, listaposicao)			