

Programação Orientada a Objetos

2017/2018

Trabalho Prático Meta 1

Hugo Moreira – 21240034
Filipe Ribeiro – 21220620

Índice

Classes consideradas.....	3
Conceitos/classe que identificou ao ler o enunciado	3
Identifique em que classes ou partes do programa são criados, armazenados e destruídos os seus objetos	3
Exemplo de uma responsabilidade atribuída a uma classe que esteja de acordo com a orientação dada acerca de Encapsulamento.....	3
De entre as classes que fez, escolha duas e justifique por que considera que são classes com objetivo focado, coeso e sem dispersão	3
Quais as classes com responsabilidades de interface com o utilizador e quais as que representam a lógica	3
A classe que representa a envolvente de toda a lógica executa em pormenor muitas funcionalidades, ou delega noutras classes.....	4
Principais classes da aplicação	4

Classes consideradas

Para a primeira meta consideramos as seguintes classes:

Classe Comunidade, guarda um ponteiro para um Ninho e um vetor de ponteiros para Formigas;

Classe Ninho, guarda num vetor de ponteiros todas as Formigas pertencentes a esse Ninho/Comunidade;

Classe Formiga, guarda um ponteiro para a sua Comunidade e um Ponteiro para o seu Ninho e lida com a parte lógica das Formigas como o seu movimento e as suas características;

Classe Iteracao, cria o Mapa e o aspeto visual da interface, recebe os comandos do Utilizador, guarda um ponteiro para o Motor;

Classe Motor, recebe os pedidos da Iteracao e faz o seu processamento, guarda todas as Formigas num vetor de ponteiros, guarda todos os Ninhos num vetor de ponteiros e guarda todas as Comunidades num vetor de ponteiros;

Conceitos/classe que identificou ao ler o enunciado

Para além dos já referidos em cima, identificamos também que será necessário criar classes derivadas para os vários tipos de Formigas, uma Classe para as Migalhas, uma Classe para as Regras

Identifique em que classes ou partes do programa são criados, armazenados e destruídos os seus objetos

As classes têm os construtores, responsáveis pela inicialização directa dos membros das classes e quando chamadas a criação de objetos e, têm destrutores responsáveis pela destruição dos objectos. São declaradas na parte pública da classe.

Exemplo de uma responsabilidade atribuída a uma classe que esteja de acordo com a orientação dada acerca de Encapsulamento

A Classe Motor contém um vetor de Comunidades, que contém um vetor com as Formigas que pertencem a essa Comunidade, para a Classe Motor ter acesso a todas as Formigas terá de percorrer o vetor de Comunidades e este percorrer o seu vetor de Formigas;

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

A classe Motor é a responsável por validar comandos, ler ficheiros, criar ninhos, criar formigas e criar comunidades por acesso indireto, às funções implementadas na parte pública nas respetivas classes. Outra classe importante é a classe Iteracao que irá ser responsável por criar mapa, mostrando a informação das formigas, ninhos conseguindo a informação usando funções públicas necessárias das respetivas classes para conseguir criar o mapa.

Sem estas duas funções, não conseguiríamos estruturar o código da maneira que se encontra, sendo por isso a sua existência de extrema importância para o decorrer normal do programa.

Quais as classes com responsabilidades de interface com o utilizador e quais as que representam a lógica

A Classe com responsabilidade de interface com o Utilizador é a Classe Iteracao que recebe os comandos do Utilizador e imprime no ecrã as informações e o mapa com as Formigas a mexerem e os Ninhos, enquanto que a Classe Motor lida com a lógica, onde cria Formigas, Ninhos e Comunidades e chama as suas funções.

A classe que representa a envolvente de toda a lógica executa em pormenor muitas funcionalidades, ou delega noutras classes

A Classe Motor é a Classe que lida com a lógica, mas delega as funcionalidades mais específicas para outras Classes, como por exemplo com o movimento das Formigas, a Classe Motor chama a função para mover na Classe Formiga;

Principais classes da aplicação

Classe Comunidade;

Permite consultar o seu ID;

Guarda um ponteiro para Ninho de forma a esse Ninho ter apenas uma Comunidade, guarda um vetor de Formigas que contem todas as Formigas daquela Comunidade de forma a elas se “reconhecerem” em campo;

Classe Formiga;

Permite consultar as suas características, como a posição, a energia, o ID, ...;

Tem a função para o seu movimento;

Guarda um ponteiro para o Ninho a que pertence e também para a sua Comunidade, de forma a pertencer a apenas uma Comunidade e a um Ninho;

Classe Ninho;

Permite consultar a sua posição, a sua energia e o seu ID;

Guarda um vetor de Formigas, para poder adicionar energia às Formigas que com as mesmas coordenadas que o Ninho e pertencentes a esse Ninho;

Classe Motor;

Permite consultar todas as Comunidades, todas as Migalhas e todos os Ninhos;

Valida os Comandos fornecidos pela Classe Iteracao, cria Formigas, cria Ninhos, cria Comunidades, liga os Ninhos às respetivas Comunidades e Formigas,

Guarda num vetor de ponteiros todas as Comunidades, todos os Ninhos num vetor de ponteiros e todas as Migalhas num vetor de ponteiros;

Classe Iteracao;

Recebe os comandos dados pelo utilizador e envia para a Classe Motor, cria o Mapa, cria a interface;

Guarda um ponteiro para o motor para ligar a interface à lógica;

Exemplo de uma funcionalidade que varia conforme o tipo do objecto que a invoca. Indique em que classes e métodos está implementada esta funcionalidade.

Como havia vários tipos de Formigas tornou-se necessário dividir por diferentes objectos, como por exemplo “Cuidadoras”. A classe Formigas deverá poder integrar Cuidadoras, Vigilantes e Exploradoras. Uma das funcionalidades da Formiga é obter as regras dos vários tipos de Formigas que o integram.