

Programação Orientada a Objectos 2018/2019

META 2

Índice

<i>Organização do código apresentado.....</i>	<i>2</i>
<i>Funcionalidades implementadas.....</i>	<i>4</i>
<i>Bugs Conhecidos.....</i>	<i>4</i>

Organização do código apresentado

“Quais foram as classes consideradas na primeira versão que foi testada?”/“Quais os conceitos(classe) que identificou ao ler o enunciado?”

Calmaria
Celula
Escuna
Especial
Evento
Fragata
Galeao
Interface
Jogo
Mapa
Mar
Motim
Navio
Porto
Sereia
Tempestade
Terra
Terreno
User
Veleiro

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”

As classes duas classes principais da aplicação são a classe Jogo e User, sendo o Jogo e o User ambos criados/armazenados e destruidos no main acabando apenas quando a aplicação termina.

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

A classe mapa tem a responsabilidade de armazenar o vector de celulas que apenas pode ser acedido através da classe mapa, sendo por isto um exemplo de encapsulamento, pois é necessário pedir à classe mapa para aceder ao vector de celulas.

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

A classe Evento e a classe Navio pois são duas classes cujos objectivos é permitir criar os respectivos eventos/navios, não tende qualquer outra responsabilidade.

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

A classe interface tem toda a responsabilidade com o utilizador servindo como um buffer entre a lógica do programa e o utilizador, em termos lógicos as classes Jogo e Mapa são consideradas as principais classes responsáveis pela lógica da aplicação.

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

O primeiro objecto que coordena e recebe a funcionalidade de natureza lógica é o user.

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(jogo) que representa a envolvente de toda a lógica executa muitas funcionalidade noutras classes, por exemplo, esta contem vários vectores de objectos que são tratadas nas classes dos objectos associados.

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)”

A classe navio varia conforme o tipo de objecto que a invoca, estando esta distribuida nas classes Navio(classe pai), Fragata,Galeao,Escuna,Veleiro e Especial. Sendo o metodo principal onde isto se encontra implementado o metodo virtual string getAsString() const;

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

Classe: Jogo

Responsabilidades:

- Permite criar toda a “inteligencia artificial” da aplicação.
- Permite consultar a matriz de mapa da aplicação.

Colaborações:

- Colabora com as N classes descendentes da classe navio(Fragata,Galeao,Especial,Escuna,Veleiro) através de um vector de ponteiros para objectos.
- Colabora com as N classes descendentes da classe Evento tendo 2 vectores de ponteiros para objectos.
- Colabora com a classe mapa contendo um ponteiro para a classe mapa.

Classe:User

Responsabilidades:

- Permite criar todos os objectos controlados pelo utilizador na aplicação.

Colaborações:

- Colabora com as N classes descendentes da classe navio(Fragata,Galeao,Especial,Escuna,Veleiro) através de um vector de ponteiros para objectos.

Classe:Mapa

Responsabilidades:

- Apresentar toda a informação da aplicação (vector de vector de celulas)
- Gerir os portos dentro do mapa

Colaborações:

- Colabora com as N classes descendentes da classe Celula/Terreno(Mar,Porto,Terra)
- Colabora com as classes User e Jogo apresentando a informação proveniente destes.

Funcionalidades implementadas

	Estado
Configuração/leitura do mapa	Implementado
Combate	Implementado
Movimento dos navios	Maioritariamente Implementado
Visualização do mapa	Implementado
Interpretação de comandos	Maioritariamente Implementado
Compra e venda de navios	Implementado
Visualização do movimento dos navios	Implementado
Eventos	Implementado
Peixe	Implementado
Turnos de cada navio	Implementado
Guardar/Carregar/Apagar jogo	Por Implementar
Comandos Vaipara	Por Implementar

Bugs Conhecidos

Os navios caso se desloquem mais do que um espaço de distancia, por vezes saltam por cima de terrenos terra.

Ocasionalmente o movimento dos piratas faz com que os crashem o programa.

Poderam existir mais bugs que de momento não teremos conhecimento sobre eles.