

Departamento de Engenharia Informática e de Sistemas

Instituto Superior de Engenharia de Coimbra

Engenharia Informática

# Programação Orientada a Objetos

2018/2019

TP 8

**Trabalho realizado por:**

Luís Freitas - 21270475

João Correia – 21250416

Índice

Classes consideradas...........................................................3/4/5/6/7

Funcionalidades implementadas................................................9/10

Classes consideradas

Classe: **Jogador**

Responsabilidades:

class Jogador{

int moedas;

public:

vector<Ship> Navios;

Jogador();

int compranav(int x0,int y0,char tipo0);**Compranav-> permite comprar navios.**

int getMoedas()const; **GetMoedas ->consultar as moedas.**

char getTipoNavviaID(int id0)const;**GetTipoNavviaID-> retorna tipo de navio através do ID do navio.**

vector<Ship> getNavios()const;

void printNavios()const;**printNavios-> imprime os navios do jogador**

void setMoedas(int moedas0);**setMoedas-> muda as moedas do jogador**

string getAsString() const; };

Colaborações: A classe Jogador contem um vector com objetos da classe Ship.

Classe: **Ship**

Responsabilidades:

class Ship{

int x,y;

int id;

char tipo;

static int conta;

int velocidade;

int Tpeixe = 0;

int agua;

int nsoldados = 0;

public:

Ship(int x0,int y0,char tipo0);**Construtor objeto ship.**

int geraId();**Gera ID do ship.**

int getX()const;**Obtem posição x.**

int getY()const;**Obtem posição y.**

int getId()const;**Obtem ID do ship.**

char getTipo()const;**Obtem Tipo Ship.**

int getVelocidade()const;**Obtem velocidade Ship.**

int getTpeixe()const;**Obtem quantidade de peixe do Ship.**

int getAgua()const;**Obtem Agua do Ship.**

int getNsoldados()const;**Obtem Soldados do Ship.**

void setX(int x0);**Muda a pos X do Ship.**

void setY(int y0);**Muda a pos Y do Ship.**

void setId(int id0);**Muda o ID do Ship.**

void setVelocidade(int velocidade0);**Muda o ID do Ship.**

void setTpeixe(int tpeixe0);**Muda a quantidade do Ship.**

void setAgua(int agua0);**Muda a água do Ship.**

void setNsoldados(int nsoldados0);**Muda a quantidade de soldados do Ship.**

string getAsString()const;

string getAsString2()const;};

Colaborações: Usa as settings no construtor da class Ship.

Classe: **Settings**

Responsabilidades:

class Settings{

int ncols,nlinhas,probpirata,soldadosport,preconavio;

public:

const int tpeixeVMAX = 40,tpeixeGMAX = 70,tpeixeEMAX = 20,tpeixeFMAX = 0,tpeixeSMAX = 40;

const int capAguaVMAX = 200, capAguaGMAX = 400,capAguaEMAX = 100,capAguaFMAX = 500,capAguaSMAX = 40;

const int velVMAX = 2,velGMAX = 1,velEMAX = 1,velFMAX = 2,velSMAX = 2;

int turnos=0;

Settings();**Cria classe Settings.**

int getLinhas()const;**Obtem as linhas.**

int getColunas()const;**Obtem as colunas.**

int getProbpirata()const;**Obtem a probabilidade pirata.**

int getSoldadosPort()const;**Obtem a quantidade de soldados no porto.**

int getPrecoNavio()const;**Obtem o preco navio.**

void setLinhas(int linhas0);**Muda as linhas.**

void setColunas(int colunas0);**Muda as colunas.**

void setProbpirata(int probpirata0);**Muda a probabilidade pirata.**

void setSoldadosPort(int soldadosport0);**Muda a quantidade de soldados no porto.**

void setPrecoNavio(int preconavio0);**Muda o preço de custo do navio.**};

Colaborações: Podem ser acedidos por qualquer classe.

Classe: **Celula**

Responsabilidades:

class Celula{

int x,y;

char c;

int peixe;

int lot;

vector<int> ids;**//vetor com ids dos barcos que estao nas celulas**

public:

Celula(int x0, int y0, char c0);**Construtor da classe celula.**

void setX(int x0);**Muda a posição X.**

void setY(int y0);**Muda a posição Y.**

void setCaract(char c0);**Muda o caractere.**

void addId(int id);**Adiciona id ao vetor dos ids.**

void removeId(int id);**Remove id ao vetor dos ids.**

vector<int> getidsNavios()const;**Manda vector com os ids dos navios na celula específica.**

int getX()const;**Obtem posição X.**

int getY()const;**Obtem posição Y.**

int getId()const;**Obtem ID da celula.**

int getqntbarcos()const;**Obtem quantidade de barcos na celula.**

char getCaract() const;**Obtem char na celula.**

string getAsString()const;};

Colaborações: Não tem.

Classe: **Mapa**

Responsabilidades:

class Celula;

class Porto;

class Mapa **//serve so para ver; nao altera**{

int linhas, colunas;

vector<Porto> Portos;

vector<Celula> grelha;

public:

Mapa( int linhas0, int colunas0, vector <char> m);

int getLinhas()const;**Obtem linhas.**

int getColunas()const;**Obtem colunas.**

vector <int> getPortoPrinc()const;**Retorna posição do primeiro porto amigo.**

void setLinhas(int linhas0);**Muda as linhas.**

void setColunas(int colunas0);**Muda as colunas.**

void printmapa()const;**Printa o mapa para a consola.**

void printPortos()const;**Printa os portos.**

vector<Porto>getPortos()const;**Retorna portos contidos no Mapa.**

void addIdtoCelula(int x0,int y0,int id);**Adiciona ID de navio ao vetor de IDs da célula.**

void remIdtoCelula(int x0,int y0,int id);**Remove ID de navio no vetor de IDs da célula.**

char getCelula(int x, int y) const;**Retorna o caractere na célula desejada.**

vector<int> checkpos(int x,int y,int xobj,int yobj);**Vai confirmar se o navio pode movimentar-se para as células adjacentes.**

int getidNavioemCelulaXY(int x0,int y0)const;**Retorna o ID no navio que se encontra em certa célula.**

void removeidemCelulaXY(int x0,int y0,int id0);**Remove ID do navio em determinada céula.**

void addidemCelulaXY(int x0,int y0,int id0);**Adiciona ID do navio em determinada céula.**

void printIdCelula(int x0,int y0);**Printa os IDs de todos os navios nessa célula.**

string getAsString() const;};

Colaborações: Colabora com a classe porto para criar um vector porto assim como para o vector grelha que é da classe célula.

Classe: **Porto**

Responsabilidades:

class Porto{

int x,y;

char c;

int nsoldados=0;

int relacaoJogador=0;

int relacaoPirata=0;

public:

Porto(int x0,int y0,char c0);**Construtor da class Porto.**

int getX()const;**Obtem posição X;**

int getY()const;**Obtem posição Y;**

char getChar()const;**Obtem caractere.**

int getnSoldados()const;**Obtem quantidade de soldados.**

int getRelacaoJogador()const;**Obtem Relação com jogador.**

int getRelacaoPirata()const;**Obtem Relação com piratas.**

void setX(int x0);**Muda a posição X;**

void setY(int y0);**Muda a posição Y;**

void setChar(char c);**Muda o caractere;**

void setNsoldados(int nsoldados0);**Muda a quantidade de soldados;**

void setRelacaoJogador(int relacaoJogador0);**Muda a Relação jogador;**

void setRelacaoPirata(int relacaoPirata0);**Muda a Relação pirata;**

string getAsString()const;};

Colaborações:Não tem.

Funcionalidades implementadas

Todas as funcionalidades foram implementadas para esta meta.

Configuração/leitura do mapa->

Lê do ficheiro settings/configurações presentes no ficheiro de texto para o programa. É possível criar diferentes tamanhos de mapa e mudar as definições.

Visualização do mapa->

Apresenta no ecrã todo o mapa criado no ficheiro enviado para leitura.

Interpretação de comandos->

Nesta primeira fase ainda apresenta uma interpretação limitada de comandos, estes são:

**compranavdb <X> <Y> <Tipo>**

**compranav <Tipo>**

**prox**

Compra de navios->

É possível comprar navios com o comando de debug (**compranavdb <X> <Y> <Tipo>**) e com o comando normal de jogo (**compranav**) onde o navio dá spawn no porto amigável.

Visualização do movimento dos navios->

Os navios movimentam-se no mapa, têm cor verde,id em cima do tipo e andam aleatoriamente com velocidade pré-definida (a velocidade é quantos saltos no mapa dá por turno).

