



MATÉRIA: Introdução a programação - Profº Danton Cavalcanti Franco Junior

Nome da dupla: Jhuly Zandomeneco Gonçalves
Amanda Cristina Suave Vieira

BATALHA NAVAL

O primeiro passo para inicializarmos nosso projeto, foi criar a classe chamada BatalhaNaval, conforme imagem.

No print também está mostrando dois import, o primeiro é o java.util.Random que utilizamos para gerar vários números aleatórios e o segundo o java.util.Scanner, que utilizamos para conseguir ler a entrada do usuário.

```
import java.util.Random;
import java.util.Scanner;

public class BatalhaNaval {
```

Posteriormente acionamos o Scanner, que será responsável por ler a entrada do jogador. Logo após, criamos as duas matrizes, a matriz tabuleiro que ficará responsável por guardar as posições dos navios e a matriz tabuleiroVisivel que será a matriz que ficará visível para o jogador e que será modificada durante o jogo.

```
Scanner scanner = new Scanner(System.in);
char[][] tabuleiro = new char[8][8];
char[][] tabuleiroVisivel = new char[8][8];
```

Na sequência, inicializamos duas variáveis. A primeira int naviosRestantes será a que ficará armazenado o número de navios restantes no jogo e a segunda int tentativasRestantes que armazenará o número de tentativas que ainda restam para o jogador.

```
int naviosRestantes = 10;
int tentativasRestantes = 30;
```

Após isso, criamos o construtor que ficará responsável por inicializar e executar o jogo. A baixo dele, já começamos a inicializar os métodos, sendo eles:



MATÉRIA: Introdução a programação - Profº Danton Cavalcanti Franco Junior

Nome da dupla: Jhuly Zandomeneco Gonçalves
Amanda Cristina Suave Vieira

inicializarTabuleiro(tabuleiro); que ficará responsável por inicializar o tabuleiro com água.

InicializarTabuleiro(tabuleiroVisivel); que ficará responsável por inicializar o tabuleiro com água que ficará visível para o jogador.

PosicionarNavios(); que ficará responsável por posicionar os navios de forma aleatória no tabuleiro.

```
Run | Debug
public static void main(String[] args) {
    new BatalhaNaval();
}

public BatalhaNaval() {
    inicializarTabuleiro(tabuleiro);
    inicializarTabuleiro(tabuleiroVisivel);
    posicionarNavios();
}
```

Depois nós fizemos o while que ficará responsável por criar o loop do jogo e que executará continuamente até que todos os navios sejam encontrados ou o número de tentativas se esgote.

Também inicializamos o método exibirTabuleiro que ficará responsável por exibir o tabuleiro visível para o jogador no início do jogo e após cada tentativa.

```
while (naviosRestantes > 0 && tentativasRestantes > 0) {
    exibirTabuleiro(tabuleiroVisivel);
}
```

Na sequência colocamos textos na tela do jogador informando o número de tentativas restantes e o número de navios restantes. Logo após, exibimos uma mensagem na tela solicitando as coordenadas tanto da linha quanto da coluna. Isso tudo executará continuamente até a finalização do jogo.

MATÉRIA: Introdução a programação - Profº Danton Cavalcanti Franco Junior

Nome da dupla: Jhuly Zandomeneco Gonçalves
Amanda Cristina Suave Vieira

```
System.out.println("Tentativas restantes: " + tentativasRestantes);
System.out.println("Navios restantes: " + naviosRestantes);

System.out.print("Digite a coordenada (linha): ");
int linha = scanner.nextInt();
System.out.print("Digite a coordenada (coluna): ");
int coluna = scanner.nextInt();
```

Fizemos um if para validar se as coordenadas informadas pelo jogador são válidas e posteriormente fizemos outro if para verificar se as coordenadas informadas já não foram utilizadas anteriormente. Esses if sempre irá fazer essas validações após cada tentativa do jogador.

```
if (validarCoordenadas(linha, coluna)) {
    if (tabuleiroVisivel[linha][coluna] == 'X' ||
        tabuleiroVisivel[linha][coluna] == 'O') {
        System.out.println("Você já atacou essa posição.");
    }
}
```

Após fizemos um else if que ficará responsável por verificar se as coordenadas informadas pelo jogador acertou um navio. Caso o jogador tenha acertado, iremos marcar essa posição (linha + coluna) na matriz tabuleiro informando que ali existe um navio. Na sequência, será exibido uma mensagem em tela informando que o jogador acertou um navio. Iremos também substituir essa posição informada na matriz tabuleiroVisivel com um x, informando assim que ali existe um navio e permitindo que o jogador visualize no tabuleiro o navio que ele acertou. Na sequência, iremos decrementar a quantidade de navios restantes na variável. Esse else if será executado sempre que as coordenadas informadas pelo jogador acertarem um navio.

```
else if (tabuleiro[linha][coluna] == 'N') {
    System.out.println("Você acertou um navio!");
    tabuleiroVisivel[linha][coluna] = 'X';
    naviosRestantes--;
```

Fizemos também um else que será executado sempre que o jogador não acertar um navio, ou seja, errar uma posição. Nele será exibido uma mensagem em tela informando que o jogador errou e irá marcar na matriz tabuleiroVisivel a posição informada com um 'O' permitindo assim marcar a posição errada no tabuleiro visível para o jogador para que ele visualize. Iremos também decrementar o número de tentativas restantes na variável.



MATÉRIA: Introdução a programação - Profº Danton Cavalcanti Franco Junior

Nome da dupla: Jhuly Zandomeneco Gonçalves
Amanda Cristina Suave Vieira

```
else {  
    System.out.println("Você int coluna = BatalhaNaval.BatalhaNaval()  
    tabuleiroVisivel[linha][coluna] = 'O';  
}  
tentativasRestantes--;  
}
```

Também criamos um else que irá ser executado sempre que as coordenadas informadas pelo jogador forem inválidas. Onde exibirá um texto em tela informando que as coordenadas são inválidas e solicitando para que o usuário tente novamente.

```
else {  
    System.out.println("Coordenadas inválidas. Tente novamente.");  
}
```

Logo após saindo do loop do while, fizemos um if que se caso o número contido na variável naviosRestantes for igual a 0, informa que o usuário encontrou todos os navios e portanto ele ganhou o jogo e uma mensagem de parabéns será exibida.

Fizemos também um else onde será executado sempre que ele não executar o if, pois nesse caso, a quantidade de tentativas se esgotaram e o jogador perdeu o jogo. Uma mensagem de suas 30 tentativas acabaram será exibida.

```
if (naviosRestantes == 0) {  
    System.out.println("Parabéns! Você destruiu todos os navios.");  
} else {  
    System.out.println("Suas 30 tentativas acabaram!");  
}
```