



## Trabalho prático

### Instruções

1. O trabalho deve ser implementado utilizando a linguagem Java.
  - 1.1. Deve conter o mínimo de classes do modelo especificado na Figura 3, bem como as classes concretas e as classes de controladoras. Novas classes podem ser criadas conforme a necessidade.
  - 1.2. Poderá ser desenvolvido individualmente ou em equipe de, **no máximo**, duas pessoas.
2. O trabalho de ser entregue via Moodle. Haverá um *link* de entrega no sistema para fazer o *upload* do arquivo. O trabalho deverá ser entregue até as 23:55hs do dia **06/02/2017**. Coloque seu nome e R.A. como nome do arquivo compactado. Exemplo: MariaSilva123456.rar (sem espaço). Se o trabalho for feito em dupla, separe os nomes e respectivos R.A.'s por um “\_” (*underline*). Exemplo: FulanoTal23455\_SicranoOutro67890.rar (sem espaço);
3. Devem ser entregues, via Moodle, os seguintes artefatos compactados em um arquivo:
  - 3.1. Código fonte do programa;
  - 3.2. Documento que descreve as classes definidas no trabalho – ver Apêndice 1.
4. O trabalho deve ser apresentado a partir do dia **06/02/2017** em ordem alfabética constante na chamada.
  - 4.1. Embora o trabalho possa ser feito em dupla, isto não garante mesma nota para os integrantes da equipe.
  - 4.2. A apresentação deve abranger:
    - a) as decisões de projeto do sistema;
    - b) implementação das classes de projeto.

### Objetivo do trabalho

Aplicar os conceitos de orientação a objetos na programação de um sistema de software que simula um jogo de estratégia para conquistar territórios (baseado no jogo de tabuleiro War com regras simplificadas). Um exemplo de funcionamento do jogo de tabuleiro real (War I) pode ser visto em <http://www.growgames.com.br/Jogos/War>

### Descrição do Trabalho

O jogo consiste em disputar territórios definidos no mapa ilustrado na Figura 1. As linhas tracejadas no mapa indicam os caminhos que podem ser tomados ao deslocar os exércitos pelos territórios que não possuem fronteira direta.

Inicialmente os territórios são sorteados entre os jogadores. O objetivo é conquistar dois continentes quaisquer na sua totalidade. Para isso, cada jogador dispõe de um contingente inicial de exércitos terrestre e aéreo. O exército terrestre ocupa os territórios pertencentes ao jogador que pode distribuí-lo conforme sua estratégia, e também participa de batalhas por outros territórios. O exército aéreo só pode ser utilizado no momento do ataque para abater uma parte do número do exército adversário, se assim o jogador desejar.



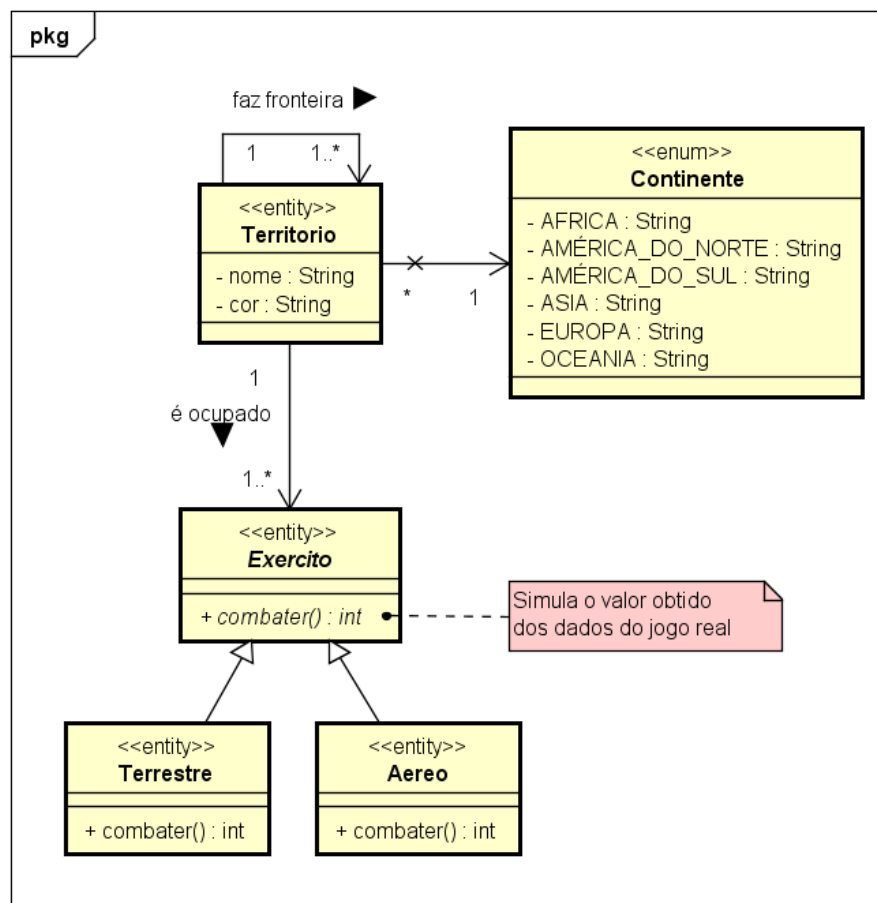
Figura 1: Mapa ilustrativo dos territórios

A implementação do mapa da Figura 1 deve ser realizada utilizando a matriz ilustrada na Figura 2. A partir dessa matriz pode-se verificar todas as fronteiras válidas entre os territórios. São válidas somente as fronteiras na horizontal e na vertical da matriz. Exemplo: o Brasil faz fronteiras com Colômbia e Argentina, mas não faz fronteira com o Chile. Observe que os territórios da última coluna fazem fronteira horizontal com os territórios da primeira coluna. Exemplos: Vladivostok faz fronteira horizontal com Alasca; Sibéria só tem fronteira horizontal com Omsk; Chile faz fronteira horizontal com Colômbia e Japão.

	1	2	3	4	5	6	7	8
1	Alasca	Vancouver	Groenlândia	Inglaterra	Itália	Suécia		Vladivostok
2		Califórnia	Otawa		Alemanha	Moscou	Omsk	Sibéria
3		México	Nova York	Nigéria	Egito	Oriente Médio	Índia	China
4	Chile	Colômbia		Congo	Sudão	Sumatra	Bornéu	Japão
5	Argentina	Brasil		África do Sul	Madagascar		Austrália	Nova Guiné

Figura 2: Matriz de fronteiras válidas entre os territórios

A Figura 3 mostra o diagrama de classes que define as classes básicas que devem ser implementadas.



powered by Astah

Figura 3: Diagrama de classes básico

A classe `Exercito` é uma classe abstrata, portanto, não poderão ser instanciados objetos do tipo `Exercito`, mas sim objetos do tipo `Terrestre` e do tipo `Aereo` que representam, respectivamente, os exércitos terrestres e exércitos aéreos. Um exército, seja terrestre ou aéreo, pode realizar combates para conquistar um território por meio do método `combater()`, que retorna um valor aleatório inteiro, simulando os valores obtidos dos dados do jogo real. O vencedor da disputa tem o atributo `cor` da classe `Territorio` alterado para a cor que identifica o jogador, significando que aquele território passa a pertencer a ele.

Observação: o atributo `cor` da classe `Territorio` pode ser alterada para uma variável de instância do tipo `Enum`.

## Definições e regras gerais

O jogo possui três momentos:

- 1) **Inicialização:** quando os territórios são distribuídos entre os jogadores de forma aleatória (identifica-se o dono do território pelo seu atributo `cor`);
- 2) **Preparação:** quando os jogadores posicionam seus exércitos terrestres e aéreos nos territórios recebidos no momento anterior;
- 3) **Execução:** quando as rodadas de ataque e defesa entre os jogadores estão em andamento até que um deles vença o jogo.
  - 3.1) Em cada rodada, o jogador pode optar pela combinação de ataques terrestres e aéreos,



respeitando as fronteiras definidas no mapa ilustrado da Figura 2.

Para simplificar, os participantes do jogo podem ser o usuário e a máquina como adversária. A máquina pode ter suas jogadas programadas de modo aleatório, ou então pode-se implementar as jogadas da máquina com um certo grau de “inteligência” observando determinadas condições antes de executar uma ação.

O número de exércitos (terrestres e aéreos) que podem participar de uma disputa por um território é limitado a no máximo 3, tanto para o ataque quanto para a defesa. O ataque aéreo só é possível se o adversário possuir no mínimo um exército aéreo em seu território.

No momento da disputa, o método `combater()` de cada exército é chamado e retorna um valor inteiro aleatório que correspondem às faces do dado do jogo real de tabuleiro. Assim, o retorno do método `combater()` entre exércitos terrestres equivale à comparação dos valores obtidos pelos dados vermelhos de ataque contra os dados amarelos de defesa do jogo de tabuleiro. Já o retorno do método `combater()` entre exércitos aéreos representa as faces dos dados azuis de bombardeio.

Compara-se os valores obtidos por jogador e vence aquele que obtiver um valor maior. No caso de empate, a defesa vence.

## Funcionamento e regras do jogo

### 1. Distribuição dos territórios

- Os territórios podem ser distribuídos de modo aleatório entre os jogadores.
- Cada território passa a ter um exército terrestre de ocupação e um exército aéreo.
- Após a distribuição dos territórios, os jogadores poderão alocar seus exércitos conforme a quantidade de contingente disponível.

### 2. Distribuição dos exércitos terrestres

- A cada rodada, o jogador tem direito a receber uma quantidade de exércitos terrestres que corresponde à metade inteira do total de territórios.
- O jogador poderá distribuir os exércitos terrestres em seus territórios conforme sua estratégia.

### 3. Distribuição dos exércitos aéreos

- A cada rodada, o jogador tem direito a receber uma quantidade de exércitos aéreos que corresponde a um terço do total de territórios, considerando o quociente inteiro.
- O jogador poderá distribuir os exércitos aéreos em seus territórios conforme sua estratégia.

### 4. Rodadas

- Após a distribuição inicial dos territórios, as rodadas são iniciadas com a distribuição dos exércitos terrestres e aéreos, de acordo com a quantidade de territórios pertencentes a cada jogador.
- Dentro da rodada, o jogador deve optar por:
  - Combate terrestre,
  - Combate aéreo,
  - Remanejamento de exércitos.
- Ao remanejar exércitos, passa-se a vez para o próximo jogador.

### 5. Combate terrestre

- O jogador escolhe um território para atacar, utilizando no máximo 3 exércitos oriundos de um território adjacente que lhe pertence. Não importa se no seu território há contingente com valor maior, o número máximo de exércitos que se pode utilizar no ataque é 3.



- Só é permitido atacar um território adversário que faça fronteira com um território pertencente ao jogador, ou que esteja ligado por uma linha pontilhada no mapa da Figura 2 (como a Terra é redonda, pode-se atacar Vladivostok a partir do Alasca e vice-versa).
  - Não se pode deslocar todo o contingente do exército para o ataque: deve permanecer pelo menos um exército no território para sua ocupação. Portanto, para efetuar um ataque terrestre é necessário que haja pelo menos dois exércitos: um será utilizado para o ataque e o outro permanece ocupando o território.
    - Exemplo: supondo que o jogador possua o México e Nova York, e deseja conquistar a Califórnia a partir do México. Ele poderá utilizar somente os exércitos do México para a disputa, desde que haja exércitos suficientes no México.
- O método `combater()` é chamado o número de vezes correspondente a quantidade de exércitos em batalha, respeitando o limite máximo de 3 exércitos para ataque e 3 para defesa.
- No combate terrestre, esse método deve retornar um valor entre 1 e 6.
  - O turno é iniciado por aquele que ataca. Os valores obtidos pelas chamadas do método feitas pelos exércitos atacantes e atacados são ordenados e comparados entre si tal como no jogo de tabuleiro.
  - Somente exércitos terrestres participam do combate terrestre. Exércitos aéreos não podem defender ataques terrestres, nem serem utilizados como exército de ocupação do território.
  - Quando o jogador perde um território que ainda possui exércitos aéreos, esses exércitos passam a ser do oponente.
- Se, após uma batalha (um ou mais confronto de dados), o atacante destruir todos os exércitos do território do defensor, terá então conquistado o território e deverá deslocar seus exércitos atacantes para o território conquistado, onde eles permanecerão. Este deslocamento obedece à seguinte regra: o número de exércitos a ser deslocado neste instante é igual, no máximo, ao número de exércitos que participaram do último ataque.
- Após conquistar um território e deslocar os exércitos permitidos, o atacante, se quiser, poderá realizar novo ataque a partir do território conquistado, ainda na mesma jogada. Somente então é que vai efetuar novo deslocamento dos exércitos atacantes para o território atacado.
- Obs.: Durante os ataques, só é permitido deslocar os exércitos atacantes para os territórios atacados, e não para outro território qualquer.
- **Exemplo:**
  - O jogador deseja conquistar o Chile que está ocupado por 4 exércitos terrestres. Considere que o jogador utilize o limite máximo de 3 exércitos para o ataque vindos da Argentina. O método `combater()` de cada exército do jogador é chamado. Por sua vez, o oponente pode chamar o método `combater()` de 3 exércitos terrestres seus, porque possui 4 exércitos. Suponha que são obtidos os seguintes dados:

Turno 1			
Jogador (3 exércitos)	Adversário (4 exércitos)	Comparação	Resultado
<code>combater()</code> retornou 5	<code>combater()</code> retornou 3	6 = 6	Jogador perde 1 exército da Argentina Adversário perde 2 exércitos do Chile
<code>combater()</code> retornou 6	<code>combater()</code> retornou 6	5 > 3	
<code>combater()</code> retornou 2	<code>combater()</code> retornou 1	2 > 1	

- Agora o jogador possui saldo de 2 exércitos no ataque e o adversário possui 2 exércitos no seu território. Se o jogador desejar prosseguir com o combate terrestre, suponha que obtém-se os seguintes dados:



Turno 2			
Jogador (2 exércitos)	Adversário (2 exércitos)	Comparação	Resultado
<code>combater()</code> retornou 1 <code>combater()</code> retornou 4	<code>combater()</code> retornou 1 <code>combater()</code> retornou 2	$4 > 2$ $1 = 1$	Adversário perde 1 exército do Chile Jogador perde um exército da Argentina

- Agora o jogador possui saldo de 1 exército no ataque e o adversário possui 1 exército na defesa. Se o jogador desejar prosseguir com o ataque, suponha que obtém-se os seguintes dados:

Turno 3			
Jogador (1 exército)	Adversário (1 exército)	Comparação	Resultado
<code>combater()</code> retornou 3	<code>combater()</code> retornou 2	$3 > 2$	Adversário perde um exército do Chile

- Como o adversário possuía somente um exército terrestre e o perdeu, então o jogador venceu e conquistou o Chile (a cor desse território muda indicando que agora ele pertence ao vencedor da batalha). Os exércitos aéreos que podem existir no território passam a ser do vencedor.
- Como o jogador usou inicialmente três exércitos no ataque, ele poderá deslocar no máximo três exércitos para ocupar o território recém-conquistado.
- O jogador pode optar por atacar um novo território adjacente ou remanejar seus exércitos.
  - Se optar por atacar um novo território, novas chamadas do método `combater()` serão comparadas.
  - Se optar por remanejar os exércitos, os deslocamentos serão feitos de acordo com a estratégia do jogador e é passada a vez.

## 6. Combate aéreo

- Os ataques aéreos servem somente para enfraquecer o exército adversário, tornando mais fácil um possível ataque terrestre.
- Para se efetuar um ataque aéreo, o território adversário deve ter no mínimo 1 exército aéreo e 4 terrestres.
- O jogador escolhe um território para atacar (bombardear), enviando uma quantidade de exércitos aéreos limitado a 3.
- Os territórios que podem ser atacados pelos exércitos aéreos são aqueles que fazem fronteira com algum território do jogador. Entretanto, o jogador pode reunir exércitos aéreos de outros territórios (compondo um ataque de no máximo três exércitos aéreos) desde que esses exércitos estejam no mesmo continente ou um continente adjacente.
  - Exemplo: supondo que o jogador possua exércitos aéreos espalhados pela Índia, Egito e Moscou e deseja conquistar a Sibéria pela China, mas o adversário possui 20 exércitos terrestres lá. Desde que o adversário também tenha no mínimo 1 exército aéreo na Sibéria, o jogador poderá enfraquecer o exército terrestre da Sibéria enviando exércitos aéreos seja da Índia, Egito ou Moscou, para posteriormente realizar o combate terrestre pela China.
- No combate aéreo, o método `combater()` deve retornar um valor entre 0 e 3.
- A cada turno, o método `combater()` é chamado primeiro por quem defende. Na defesa, somente os exércitos aéreos são abatidos. O método `combater()` é chamado por um de seus exércitos aéreos cujo retorno define quantos exércitos aéreos do ataque devem ser abatidos.
- Em seguida, o método `combater()` é chamado por cada exército aéreo em ataque. O valor retornado pelo método determina o número de exércitos oponentes que serão abatidos:



- quando o retorno é igual 0, nada acontece (nenhum exército oponente é abatido);
- quando o retorno é igual a 1, o território atacado perde 1 exército aéreo e 1 terrestre;
- quando o retorno é igual a 2, o território atacado perde 1 exército aéreo e 2 terrestres;
- quando o retorno é igual a 3, o território atacado perde 1 exército aéreo e 3 terrestres;

○ **Exemplo:**

- O jogador deseja conquistar Moscou que possui 10 exércitos terrestres e 4 exércitos aéreos. Considere que o jogador utilize o máximo de 3 exércitos aéreos para tentar enfraquecer os exércitos do adversário, vindos da Alemanha, Suécia e Omsk. O método `combater()` de um dos exércitos aéreos do território atacado é chamado para iniciar a defesa. Suponha que se obtêm os seguintes dados:

Turno 1		Resultado
Adversário (10 terrestres e 4 aéreos)	<code>combater()</code> retornou 1	Jogador perde 1 exército aéreo da Alemanha, que não poderá ser mais usado. O jogador ainda poderá usar os exércitos aéreos da Suécia e Omsk.*
Jogador (2 aéreos)	<code>combater()</code> retornou 1 <code>combater()</code> retornou 0	Adversário perde 1 exército aéreo e 1 terrestre de Moscou.

\*A ordem de retirada dos exércitos aéreos do jogador é aleatória.

- O jogador havia iniciado o ataque com 3 exércitos aéreos. Como no combate aéreo é o adversário que inicia o turno, o jogador ficou com 2 exércitos aéreos para atacar na sua vez. O jogador obteve o valor 1 de retorno do método `combater()` para um dos exércitos e 0 para o outro, o que significa que devem ser retirados 1 exército aéreo e 1 exército terrestre do território adversário, que agora tem saldo de 9 exércitos terrestres e 3 exércitos aéreos. Se o jogador optar por continuar ataque aéreo, suponha que são obtidos os seguintes dados:

Turno 2		Resultado
Adversário (9 terrestres e 3 aéreos)	<code>combater()</code> retornou 0	Nada acontece.
Jogador (2 aéreos)	<code>combater()</code> retornou 2 <code>combater()</code> retornou 2	Adversário perde 2 exércitos aéreos e 4 terrestres de Moscou.

- Agora o adversário possui 1 exército aéreo e 5 terrestres. Se jogador optar novamente pelo ataque aéreo, suponha que são obtidos os seguintes dados:

Turno 3		Resultado
Adversário (5 terrestres e 1 aéreo)	<code>combater()</code> retornou 1	Jogador perde 1 exército aéreo de Omsk, que não poderá ser mais usado. Ainda restou o exército aéreo da Suécia.*
Jogador (1 aéreo)	<code>combater()</code> retornou 0	Nada acontece

\*A ordem de retirada dos exércitos aéreos do jogador é aleatória.

- O jogador pode optar por mais um ataque aéreo ou continuar com o ataque terrestre ou remanejar seus exércitos. Se optar pelo ataque aéreo, suponha que são obtidos os seguintes dados:





Turno 4		Resultado
Adversário (5 terrestres e 1 aéreo)	<code>combater()</code> retornou 0	Nada acontece
Jogador (1 aéreo)	<code>combater()</code> retornou 3	Adversário perde 1 exército aéreo e 3 terrestres de Moscou. Sem exército aéreo, Moscou só pode ser atacado por exércitos terrestres.

- Como o adversário detinha somente 1 exército aéreo e o jogador obteve 3 pelo método `combater()`, o adversário agora não possui mais exércitos aéreos e ficou com saldo de 2 exércitos terrestres. Com o adversário sem exércitos aéreos, o jogador não pode continuar com o ataque aéreo. Portanto, o jogador deve optar por um ataque terrestre ou o remanejamento dos exércitos.
- Só é permitido prosseguir com o combate aéreo se houver 1 exército aéreo e 4 exércitos terrestres, no mínimo.
- Se optar por remanejar os exércitos, os deslocamentos serão feitos de acordo com a estratégia do jogador e é passada a vez.

## 7. Remanejamento de exércitos

- Segue as regras definidas no item 2.
- Ao remanejar os exércitos, o jogador passa a vez.

## 8. Vencedor do jogo

- Vence o jogador que conquistar dois continentes na sua totalidade.

## Pontos importantes a serem considerados durante a realização do trabalho:

- Originalidade da implementação das classes concretas e classes de projeto.
- Implementação das regras de forma correta
- Implementação dos conceitos de Programação Orientada a Objetos
  - Encapsulamento
  - Herança
  - Polimorfismo
  - Implemente o trabalho o mais orientado a objetos possível.
- Organização das classes em pacotes (seguindo a convenção de Java)
- Tratamento de exceções
- Implementação ou uso de classes de interfaces (opcional)
- Implementação ou uso de classes genéricas (opcional)
- Documentação do sistema
- Usabilidade do sistema:
  - Clareza na execução das jogadas: qual jogador (usuário ou sistema) está na vez de jogar; qual o estado dos jogadores em cada time; etc.;
- Organização dos menus (caso a interação do usuário seja em modo texto)
- Organização da interface gráfica (caso a interação do usuário seja por interface gráfica – opcional)





### **Bônus**

- Implementação de um padrão de projeto catalogado (Padrões GoF, padrões GRASP entre outros)
- Além da implementação, o padrão deve constar na documentação (diagrama de classes de projeto).

## **Apêndice 1**

### **Documentação do sistema**

Além do código da implementação e apresentação do mesmo, deve ser entregue um documento especificando:

1. **Passo-a-passo** para a compilação e execução do programa.
2. **Funcionamento geral** do programa
  - a) Como as entradas devem ser fornecidas
  - b) Qual o processamento realizado a partir das entradas
  - c) Qual(is) a(s) saída(s) esperada(s).
3. **Conceitos de Orientação a Objetos**

Escolha classes utilizadas no projeto para exemplificar como foram implementados os seguintes conceitos de orientação a objetos:

  - Encapsulamento
  - Herança
  - Polimorfismo
4. **Decisões de projeto**
  - a) Definição de novas classes

Explique o funcionamento geral do sistema (jogo) quanto:

    - Classes controladoras implementadas
    - Classes de interação com o usuário (menus de opções)
    - Tratamento de exceções (se houver)
    - Utilização de *Generics* (se houver)
    - Organização das classes em pacotes (coesão e acoplamento)
    - Implementação de padrões de projeto (se houver)
  - b) Diagrama de classes de projeto (opcional)

Apresente o diagrama de classes de projeto do sistema.