



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

INSTITUTO METRÓPOLE DIGITAL

DIM0507 - TESTE DE SOFTWARE I - T01

**ANCHEL VITOR VARELA DA SILVA
ANDRÉ FRANKLIN DE OLIVEIRA LIMA
JOÃO VICTOR DA SILVA SATURNINO**

Batalha - Testes de Software

**NATAL - RN
2024**

Definição das classes de personagem

Ambas as classes podem gastar até 20 pontos em seus atributos. Porém, **todos** devem possuir no mínimo 3 pontos.

Guerreiro

- Resistência \geq Ataque
- Ataque \geq Resistência
- Defesa $<$ Ataque
- Defesa $<$ Resistência
- Velocidade $<$ Ataque
- Velocidade $<$ Resistência

1. Partições

- (1) Resistência \geq Ataque
- (2) Resistência $<$ Ataque
- (3) Ataque \geq Resistência
- (4) Ataque $<$ Resistência
- (5) Defesa \geq Ataque
- (6) Defesa $<$ Ataque
- (7) Defesa \geq Resistência
- (8) Defesa $<$ Resistência
- (9) Velocidade \geq Ataque
- (10) Velocidade $<$ Ataque
- (11) Velocidade \geq Resistência
- (12) Velocidade $<$ Resistência
- (13) Pontos Gastos em atributo ≥ 3
- (14) Pontos Gastos em atributo < 3

Casos de teste

Considere a entrada representada por números entre chaves e separados por vírgula, onde seus respectivos valores são: {Resistência, Ataque, Defesa, Velocidade, Pontos Gastos}

ID TESTE	Entrada	Saída esperada	Partições cobertas
CT01	{6, 6, 4, 4, 20}	Guerreiro Válido	1, 3, 6, 8, 10, 12, 13
CT02	{6, 7, 4, 3, 20}	Guerreiro inválido	2, 3, 6, 8, 10, 12, 13
CT03	{7, 6, 3, 4, 20}	Guerreiro inválido	1, 4, 6, 8, 10, 12, 13
CT04	{5, 5, 6, 4, 20}	Guerreiro inválido	1, 3, 5, 7, 10, 12, 13
CT05	{5, 5, 4, 6, 20}	Guerreiro inválido	1, 3, 6, 8, 9, 11, 13

CT06	{7, 7, 5, 1, 20}	Guerreiro inválido	1, 3, 6, 8, 10, 12, 14
------	------------------	--------------------	------------------------

Análise de valor limite

Entrada	Saída esperada	Partições cobertas
{7, 7, 3, 3, 20}	Guerreiro Válido	1(max), 3(max), 6(min), 8(min), 10(min), 12(min), 13(min)
{6, 7, 4, 3, 20}	Guerreiro Inválido	2(min-), 3(min), 6(min), 8(min), 10(min), 12(min), 13(min)
{7, 6, 4, 3, 20}	Guerreiro Inválido	1(max), 4(min-), 6(min), 8(min), 10(min), 12(min), 13(min)
{5, 5, 5, 4, 19}	Guerreiro Inválido	1(max-), 3(max-), 5(max+), 7(max+), 10(max-), 12(max-), 13(min+)
{5, 5, 4, 5, 19}	Guerreiro Inválido	1(max-), 3(max-), 6(max), 8(max), 9(max+), 11(max+), 13(min+)
{7, 7, 3, 2, 19}	Guerreiro Inválido	1(max), 3(max), 6(min), 8(min), 10(min), 12(min), 14(min-)

Tabela de decisão

- **Condições**

- Resistência >= Ataque [T, F]
- Ataque >= Resistência [T, F]
- Defesa < Ataque [T, F]
- Defesa < Resistência [T, F]
- Velocidade < Ataque [T, F]
- Velocidade < Resistência [T, F]

Condição

Resistência >= Ataque	T	T	T	T	T	T	T	T	T	T	F
Ataque <= Resistência	T	T	T	T	T	T	T	T	T	F	-
Defesa < Ataque	T	T	T	T	T	T	T	T	F	-	-
Defesa < Resistência	T	T	T	T	F	F	F	F	-	-	-
Velocidade < Ataque	T	T	F	F	-	-	-	-	-	-	-
Velocidade < Resistência	T	F	-	-	-	-	-	-	-	-	-
Ação											
É Guerreiro	x										
Não é Guerreiro		x	x	x	x	x	x	x	x	x	x

Assassino

- Ataque >= Velocidade
- Velocidade >= Ataque
- Resistência <= Ataque
- Resistência <= Velocidade
- Defesa <= Ataque
- Defesa <= Velocidade

1. Partições

- (1) Ataque >= Velocidade
- (2) Ataque < Velocidade
- (3) Velocidade >= Ataque
- (4) Velocidade < Ataque
- (5) Resistência > Ataque
- (6) Resistência <= Ataque
- (7) Resistência > Velocidade
- (8) Resistência <= Velocidade
- (9) Defesa > Ataque
- (10) Defesa <= Ataque
- (11) Defesa > Velocidade
- (12) Defesa <= Velocidade
- (13) Pontos Gastos em atributos >= 3
- (14) Pontos Gastos em atributos < 3

Casos de teste

Considere a entrada representada por números entre chaves e separados por vírgula, onde seus respectivos valores são: {Resistência, Ataque, Defesa, Velocidade, Pontos Gastos}

ID TESTE	Entrada	Saída esperada	Partições cobertas
CT07	{4, 6, 4, 6, 20}	Assassino Válido	1, 3, 6, 8, 10, 12, 13
CT08	{5, 5, 4, 6, 20}	Assassino Inválido	2, 3, 6, 8, 10, 12, 13
CT09	{5, 6, 4, 5, 20}	Assassino Inválido	1, 4, 6, 8, 10, 12, 13
CT10	{6, 5, 4, 5, 20}	Assassino Inválido	1, 3, 5, 7, 10, 12, 13
CT11	{4, 5, 6, 5, 20}	Assassino Inválido	1, 3, 6, 8, 9, 11, 13
CT12	{6, 6, 2, 6, 20}	Assassino Inválido	1, 3, 6, 8, 10, 12, 14

Análise de valor limite

Entrada	Saída esperada	Partições cobertas
{5, 5, 5, 5, 20}	Assassino Válido	1(max), 3(max), 6(max), 8(max), 10(max), 12(max), 13(min+)
{4, 5, 4, 7, 20}	Assassino Inválido	2(min-), 3(min+), 6(min+), 8(min+), 10(min+), 12(min+), 13(min+)
{4, 7, 4, 5, 20}	Assassino Inválido	1(min+), 4(min-), 6(min+), 8(min+), 10(min+), 12(min+), 13(min+)
{6, 5, 4, 5, 20}	Assassino Inválido	1(max), 3(max), 5(max+), 7(max+), 10(min+), 12(min+), 13(min+)
{4, 5, 6, 5, 20}	Assassino Inválido	1(max), 3(max), 6(min+), 8(min+), 9(max+), 11(max+), 13(min+)
{6, 6, 2, 6, 20}	Assassino Inválido	1(max), 3(max), 6(max), 8(max), 10(max-), 12(max-), 14(min-)

Tabela de decisão

- Condições

- Ataque \geq Velocidade [T, F]
- Velocidade \geq Ataque [T, F]
- Resistência \leq Ataque [T, F]
- Resistência \leq Velocidade [T, F]
- Defesa \leq Ataque [T, F]
- Defesa \leq Velocidade [T, F]

Condição											
Ataque \geq Velocidade	T	T	T	T	T	T	T	T	T	T	F
Velocidade \geq Ataque	T	T	T	T	T	T	T	T	T	F	-
Resistência \geq Ataque	T	T	T	T	T	T	T	T	F	-	-
Resistência \leq Velocidade	T	T	T	T	F	F	F	F	-	-	-
Defesa \leq Ataque	T	T	F	F	-	-	-	-	-	-	-
Defesa \leq Velocidade	T	F	-	-	-	-	-	-	-	-	-
Ação											
É Assassino	x										
Não é Assassino		x	x	x	x	x	x	x	x	x	x

Mecânica de Batalha

1. Ordem dos turnos

- Personagem com maior **Velocidade** ataca primeiro.
- Em caso de empate, a ordem de ataque é aleatória.

- Tabela de decisão

- Considere “Velocidade 1” a velocidade definida do **Personagem 1**
- Considere “Velocidade 2” a velocidade definida do **Personagem 2**

ID_DEC	Condição				
DEC01	Velocidade 1 \geq Velocidade 2	T	T	F	F
DEC02	Velocidade 2 \geq Velocidade 1	T	F	T	F

ID_ACAO	Ação				
AC01	Personagem 1 inicia		x		
AC02	Personagem 2 inicia			x	
AC03	Aleatório	x			x

2. Evasão

- Número aleatório gerado \leq chance de evasão do personagem atacado, o ataque é evitado.
- Tabela de decisão
 - Considere “Num. Aleatório” o número aleatório gerado para o cálculo da evasão.
 - Considere “Chance de evasão” a porcentagem de evasão do atacado.

ID_DEC	Condição				
DEC03	Num. Aleatório \geq Chance de evasão	T	T	F	F
DEC04	Chance de evasão \geq Num. Aleatório	T	F	T	F
ID_ACAO	Ação				
AC04	Ataque evitado	x		x	x
AC05	Ataque recebido		x		

3. Dano crítico

- Número aleatório gerado ≤ 10 , o ataque é considerado um golpe crítico.
- Tabela de decisão
 - Considere “Num. Aleatório” o número aleatório gerado para o golpe crítico.

ID_DEC	Condição				
DEC05	Num. Aleatório ≥ 10	T	T	F	F
DEC06	$10 \geq$ Num. Aleatório	T	F	T	F
ID_ACAO	Ação				
AC06	Golpe crítico	X	X		
AC07	Golpe normal			X	X

4. Dano infringido

- O dano infringido não pode ser menor do que 1. Caso o cálculo do dano resultar em um número menor do que 1, o dano a ser considerado é 1.
- Tabela de decisão
 - Considere “Dano” o valor calculado do dano infringido ao oponente..

ID_DEC	Condição				
DEC07	Dano ≥ 1	T	T	F	F
DEC08	Dano ≤ 1	T	F	T	F
ID_ACAO	Ação				
AC08	Dano normal	X	X		
AC09	Dano arredondado para 1			X	X

5. Fim da batalha

- Se um dos personagens estiver com HP ≤ 0 , a batalha termina.
- Ao fim da batalha, o personagem com HP > 0 é o vencedor.

• Tabela de decisão

ID_DEC	Condição				
DEC09	HP Personagem 1 ≤ 0	T	T	F	F
DEC10	HP Personagem 2 ≤ 0	T	F	T	F
ID_ACAO	Ação				
AC10	Personagem 1 vence			X	
AC11	Personagem 2 vence		X		
AC12	Empate	X			
AC13	Novo turno				X

Critério MC/DC

O **if** mais complexo do projeto é o da função “checarRegradeClasse()”, que realiza a verificação dos atributos de classe no momento da criação para validar a criação do Guerreiro ou Assassino baseado nas restrições de cada classe. O **if** escolhido nesse caso será o do Guerreiro por possuir melhores restrições.

Todas as condições devem ser falsas para que um guerreiro válido seja criado. Quando uma delas for verdadeira, uma exceção é lançada informando que existem parâmetros inválidos para a criação de um guerreiro. **Nesta tabela, entenda a saída esperada “False” como um guerreiro que teve todas as condições de criação atendidas, e “True” como um guerreiro inválido em que a exceção foi lançada.**

ID_MCDC	Resistencia < Ataque	Ataque < Resistência	Defesa >= Ataque	Defesa >= Resustência	Velocidade >= Ataque	Vekicudade >= Resistência	Saída esperada
MCDC01	F	F	F	F	F	F	F
MCDC02	T	F	F	F	F	F	T
MCDC03	F	T	F	F	F	F	T
MCDC04	F	F	T	F	F	F	T

MCDC05	F	F	F	T	F	F	T
MCDC06	F	F	F	F	T	F	T
MCDC07	F	F	F	F	F	T	T

Os casos MCDC apresentados na tabela acima cobrem todas as possibilidades que compõem o if mais complexo, composto por “OU’s”. Uma “limitação” específica da linguagem Java não permitiu a cobertura de 100% nos testes do guerreiro. Especificamente, a terceira e quarta condições sempre retornarão o mesmo resultado, pois como o ataque é sempre igual à resistência, quando a defesa for maior ou igual à resistência, logo também será maior ou igual que o ataque, sendo assim:

`isDefesaMaiorOuIgualQueAtaque() || isDefesaMaiorOuIgualQueResistencia()`

Quando a defesa for maior ou igual que resistência, ela será maior ou igual que Ataque, assim a segunda instrução nunca será verificada por ser aplicado um “don’t care” para a linguagem, já que a primeira condição é suficiente para retornar true. O mesmo se aplica para as condições 5 e 6.

Matriz de Rastreabilidade

Partição/critério	Casos de Teste Relacionados
Resistência >= Ataque (Guerreiro)	CT01, CT03, CT04, CT05, CT06
Resistência < Ataque (Guerreiro)	CT02
Ataque >= Resistência (Guerreiro)	CT01, CT02, CT04, CT05, CT06
Ataque < Resistência (Guerreiro)	CT03
Defesa >= Ataque (Guerreiro)	CT04
Defesa < Ataque (Guerreiro)	CT01, CT02, CT03, CT05, CT06
Defesa >= Resistência (Guerreiro)	CT04
Defesa < Resistência (Guerreiro)	CT01, CT02, CT03, CT05, CT06
Velocidade >= Ataque (Guerreiro)	CT05
Velocidade < Ataque (Guerreiro)	CT01, CT02, CT03, CT04, CT06
Velocidade >= Resistência (Guerreiro)	CT05
Velocidade < Resistência (Guerreiro)	CT01, CT02, CT03, CT04, CT06
Pontos Gastos em Atributo >= 3 (Guerreiro)	CT01, CT02, CT03, CT04, CT05

Pontos Gastos em Atributo < 3 (Guerreiro)	CT06
Ataque >= Velocidade (Assassino)	CT07, CT09, CT10, CT11, CT12
Ataque < Velocidade (Assassino)	CT08
Velocidade >= Ataque (Assassino)	CT07, CT08, CT10, CT11, CT12
Velocidade < Ataque (Assassino)	CT09
Resistência > Ataque (Assassino)	CT10
Resistência <= Ataque (Assassino)	CT07, CT08, CT09, CT11, CT12
Resistência > Velocidade (Assassino)	CT10
Resistência <= Velocidade (Assassino)	CT07, CT08, CT09, CT11, CT12
Defesa > Ataque (Assassino)	CT11
Defesa <= Ataque (Assassino)	CT07, CT08, CT09, CT10, CT12
Pontos gastos em atributos >= 3	CT07, CT08, CT09, CT10, CT11
Pontos gastos em atributos < 3	CT12
Critério de ordem dos turnos Velocidade 1 >= Velocidade 2	DEC01, AC01
Critério de ordem dos turnos Velocidade 2 >= Velocidade 1	DEC02, AC02
Critério de ordem dos turnos Velocidade 1 == Velocidade 2	DEC01, DEC02, AC03
Critério de evasão Num. Aleatório >= Chance de evasão	DEC03, AC04
Critério de evasão Chance de evasão >= Num. Aleatório	DEC04, AC05
Critério de evasão Num. Aleatório == Chance de evasão	DEC03, DEC04, AC04
Critério de dano crítico Num. Aleatório > 10	DEC05, AC06
Critério de dano crítico 10 >= Num. Aleatório	DEC06, AC07
Critério de dano infringido Dano > 1	DEC07, AC08

Critério de dano infringido Dano <= 1	DEC08, AC09
Critério de fim da batalha HP Personagem 1 <= 0	DEC09, AC11, AC12
Critério de fim da batalha HP Personagem 2 <= 0	DEC10, AC12, AC10
Critério de fim da batalha HP Personagem 1 == 0 && HP Personagem 2 == 0	DEC09, DEC10, AC13
if mais complexo da aplicação	MCDC01, MCDC02, MCDC03, MCDC04, MCDC05, MCDC06, MCDC07