UNIVERSIDADE FEDERAL DO CEARÁ – UFC SOBRAL
TÉCNICAS DE PROGRAMAÇÃO – PROF. FISCHER

Prova 1: O Regaste dos Alunos de Técnicas de Programação na Ilha de Java 🤖

A Prova 1 consiste na **implementação** das funcionalidades abaixo descritas e **explicação** do código implementado. O trabalho a ser desenvolvido é um jogo que utiliza o código do Robô Andador visto em sala.

Figura 1: Ilha de Java
Fonte: https://www.thecrazytourist.com/25-best-things-java-indonesia/

1. Contexto

Os alunos de Técnicas de Programação (2023-1) empolgados com os estudos de programação resolveram fazer uma viagem para a Ilha de Java[1]. ✈️
Porém, eles se perderam na Ilha😮. Agora, precisam ser resgatados. A sorte dos alunos é que na última aula (25/04) o Robô Andador conseguiu se movimentar👋.
Desta forma, a Prova 1 será uma implementação de um jogo relacionado ao resgate dos alunos de Técnicas de Programação na Ilha de Java.

[1] https://pt.wikipedia.org/wiki/Java

2. Plano

O plano do jogo é constituído de coordenadas X e Y. As células do plano podem ter um robô (qualquer tipo), aluno ou bug. As células podem estar vazias, com aluno, com bug ou algum robô. Nunca um aluno e um bug podem estar na mesma célula (melhor evitar, vai que vira regra!). Os robôs podem em algum momento ficar na mesma célula. A Figura 2 apresenta uma ilustração do plano (exemplo: uma rodada qualquer do jogo), neste caso o plano tem as seguintes dimensões X=8 e Y=8, como os alunos perdidos (5 alunos), bugs (7 bugs) e os personagens robôs (Andador, Peão, Torre, Bispo, Cavalo, Rei e Rainha - sempre um de cada robô). **O jogo deverá ser desenvolvido sem interface gráfica**, a Figura 2 é apenas uma demonstração do jogo, não é para implementar a tela neste trabalho, (neste trabalho!🤐). Toda ação do jogador deverá ser feita mediante o terminal. No início do jogo todos os robôs começam na posição X=1 e Y=1 do plano. O plano poderá ser de qualquer tamanho (X e Y), indicado pelo jogador, inclusive X e Y podem ser diferentes.

Obs: Esse jogo não é o tradicional jogo de Xadrez, invejosos dirão que parece o Xadrez mas, não tem nada igual. Não faça a besteira de tentar procurar um código do Xadrez na internet, para facilitar sua vida, vai somente atrasar e não vai ajudar em nada.

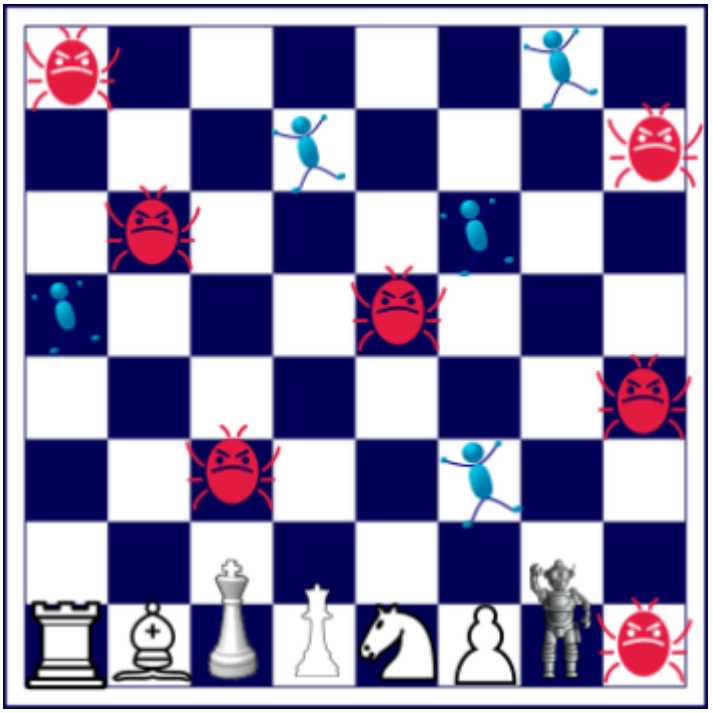

Figura 2: exemplo ilustrativo do jogo

3. **Robô**

Para otimizar a tarefa do Robô Andador foram convidados alguns robôs para serem personagens do jogo. Sendo que cada robô se movimenta de forma diferente no plano. Os robôs são listados a seguir. Cada robô tem duas opções de movimento:
Movimento **avançar** e movimento **retroceder**.

- **Andador:** Movimenta-se em qualquer número de células desejada:
  - *Avançar*: para cima
  - *Retroceder*: para baixo

- **Peão**: Movimenta-se apenas uma célula em cada jogada:
  - *Avançar*: para direita
  - *Retroceder*: para esquerda

- **Torre**: Movimenta-se até duas células em cada jogada
  - *Avançar*: para cima
  - *Retroceder*: para baixo

- **Bispo**: Movimenta-se pelas diagonais até duas células em cada jogada
  - *Avançar*: para cima
  - *Retroceder*: para baixo

- **Cavalo**: Movimenta-se pelas diagonais até duas células em cada jogada
  - *Avançar*: para direita
  - *Retroceder*: para esquerda

- **Rei**: Movimenta-se como o Cavalo, porém, até 4 células em cada jogada
  - *Avançar*: para direita
  - *Retroceder*: para esquerda

- **Rainha**: Movimenta-se como o Bispo, porém, até 4 células em cada jogada
  - *Avançar*: para cima
  - *Retroceder*: para baixo

Todo robô tem um nome e um identificador.

A cada aluno resgatado o robô ganha **10 pontos**. Para resgatar o aluno o robô tem que estar na mesma célula que o aluno se encontra. Porém, se o robô se mover para uma célula em que tenha um bug o robô perde **15 pontos**.

4. **Aluno e Bug**

Os alunos e os bugs são colocados no início do jogo no plano de forma aleatória. Para sortear coordenadas X e Y para inserir aleatoriamente alunos no plano a biblioteca

Random[2] poderá ser utilizada. O número de alunos e bugs devem ser informados pelo usuário no início do jogo. Assim, se o jogador escolher 10 alunos perdidos, o sistema deverá sortear 10 alunos para serem colocados no jogo. O número de alunos e bugs podem ser diferentes, mas sempre deverá ser menor que a metade do número de células do plano.

5. Dinâmica do Jogo

a) O jogo deverá iniciar pedindo para o jogador informar seu nome. O jogo salva a informação.
b) Posteriormente deve ser informado as dimensões (X e Y) do plano. O jogo cria o plano segundo as dimensões fornecidas.
c) O usuário deverá informar o número de alunos perdidos e bugs que o jogo terá. O jogo faz o sorteio das posições dos alunos perdidos e dos bugs sem revelar as posições escolhidas no plano.
d) As rodadas do jogo se iniciam. As rodadas ocorrem até que todos os alunos sejam resgatados ou o jogador escolha sair do jogo.
e) Em cada rodada o jogador deverá, para cada robô, indicar uma ação e o número de células que cada robô irá se movimentar no plano, o jogo deverá somente permitir movimentos válidos para cada robô (veja a Seção 3 Robô). Sua implementação deverá tratar as situações em que o jogador escolhe uma posição para o robô fora dos limites do plano. O jogo deverá mostrar em terminal o plano e onde cada robô ficou depois da rodada, bem como se um aluno foi resgatado ou se um bug atacou o robô. Ainda, as células já visitadas devem ser identificadas para facilitar o jogador. Se mais de um robô se mover para uma mesma célula na mesma rodada, os robôs ganham a pontuação se encontrarem o aluno, ou perdem se forem atacados por um bug.

f) O jogo deverá mostrar após cada rodada o número de alunos resgatados e número de bugs ocorridos.
g) O jogo deverá mostrar a pontuação de cada robô ao final de cada rodada, bem como o tabuleiro com as posições dos robôs e os alunos e bugs descobertos
h) No término do jogo um relatório deverá ser apresentado, composto das seguintes informações para cada robô (Andador, Peão, Torre, Bispo, Cavalo, Rei e Rainha), pontuação, todas as células visitadas (lista de (X,Y)), número de alunos resgatados e número de bugs encontrados. O melhor dos robôs (vencedor, maior pontuação) deverá ser apresentado juntamente com o nome do jogador.

6. Requisitos para a entrega da Prova 1

- Data: 27/05/2023
- Pontuação: 25 pontos
- Trabalho individual
- Poste seu código no GitHub
- Faça um vídeo de explicação e posto no YouTube como não listado

[2] https://docs.oracle.com/javase/7/docs/api/java/util/Random.html

- Coloque no classroom o link do seu projeto no GitHub e link do vídeo de explicação do seu trabalho
- Você deverá gravar um vídeo entre 13 a 15 minutos explicando o seu trabalho. As apresentações fora desse intervalo (13 a 15 minutos) não serão pontuadas
- Exemplos de ferramentas para gravação: ActivePresenter, câmera do celular e etc…
- No início do vídeo você deverá se apresentar (nome, matrícula e período em que você está cursando) “filmando seu rosto”
- Demais partes do vídeo podem ser apenas a apresentação da sua tela na qual você estará explicando o seu código

7. Requisitos que o seu trabalho deverá ter:

a) Encapsulamento
b) Herança
c) Polimorfismo
d) Classe abstrata
e) Construtores
f) Sobrecarga de métodos
g) Sobrescrita de métodos
h) Exceções (próximo conteúdo)
i) Interface (Orientação a Objeto) (próximo conteúdo)

Observação importante:Trabalhos que não serão pontuados:

1. Trabalho que não utilize o código iniciado em sala (Robô Andador)
2. Trabalho que não tenha o vídeo de apresentação
3. Trabalhos duplicados para mais de um aluno
4. Trabalhos que não sigam as instruções do jogo acima descritas
5. Trabalhos que não sigam o paradigma da Orientados a Objetos

Bom trabalho para todos!