

## TRABALHO FINAL – 2022/2

### OBJETIVOS

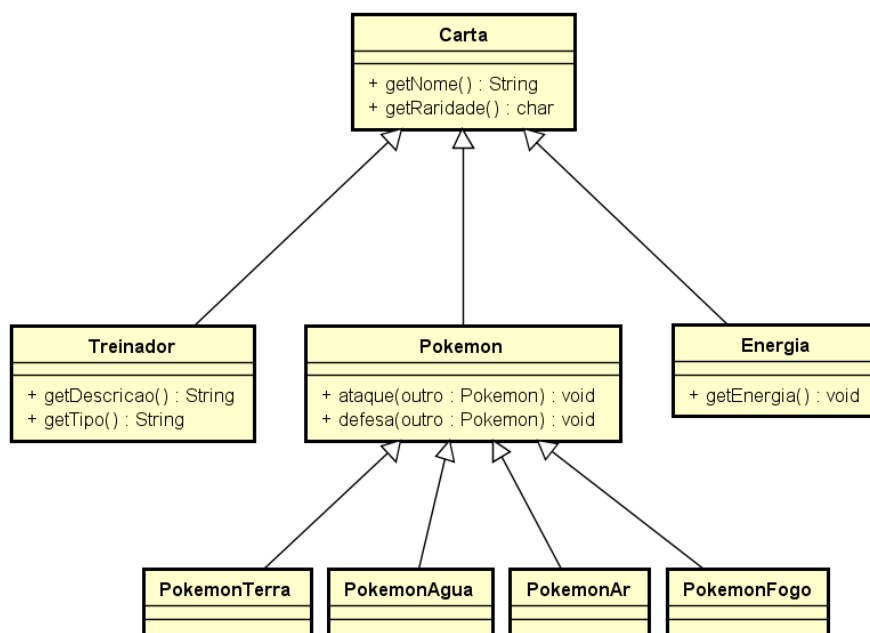
O objetivo deste trabalho é demonstrar domínio dos conceitos básicos de programação orientada a objetos em especial herança e polimorfismo, coleções, conhecimentos básicos de organização de código e criação de interface com o usuário usando JavaFX.

### TEMA DO TRABALHO

O trabalho deverá implementar um jogo ao estilo de “Jogo de Cartas Colecionável” como por exemplo “Pokemon” ou “Magic”. O tema e as regras do jogo podem ser definidos livremente desde que sejam respeitadas as seguintes restrições:

#### I. Restrições de implementação:

- Devem existir diferentes tipos de cartas que **justificadamente possam ser representadas por uma hierarquia de herança**. O diagrama abaixo é uma sugestão inicial de como poderia ser estruturada a hierarquia de classes.



- A interface com o usuário deve ser construída usando JavaFX.
- As classes devem ser coesas e ter suas responsabilidades claramente definidas. O conjunto de classes que modelam a interface com o usuário deve-se limitar a entrada e saída de dados. É obrigatório que toda a lógica de funcionamento do jogo esteja implementada em um conjunto de classes independente da interface com o usuário e que estas classes possam ser testadas de forma independente.

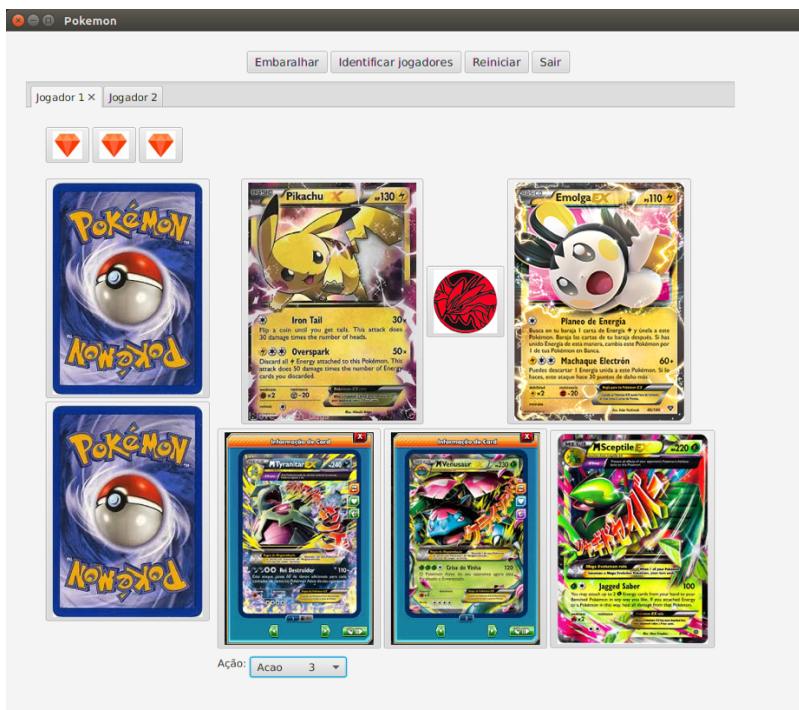
Baseado no material gentilmente cedido pelo Prof. Dr. Bernardo Copstein

## II. Restrições do jogo:

- Devem existir pelo menos 5 tipos de cartas diferentes.
- O objetivo do jogo deve ser eliminar todas as cartas do adversário.
- A cada ataque de um oponente a carta do adversário perde “vidas”, sendo eliminada da mesa quando não restarem mais vidas.
- A cada jogada um jogador pode colocar ou retirar uma carta da mesa ou acionar algum efeito da carta.
- Na vez de cada jogador, o “computador” deve anunciar de quem é a vez e só depois de confirmada a presença do jogador, deve-se mostrar a “sua tela” (ou mão).
- O “computador” deve ser capaz de detectar o final do jogo.

## III. Requisitos da interface com o usuário:

- Devem ser usadas imagens na representação das cartas que são apresentadas em cada jogada.
- Os componentes da interface devem permitir reiniciar o jogo, colocar os nomes dos dois jogadores, verificar o número de “vidas”, embaralhar e sacar as cartas (por exemplo), além de outras funcionalidades necessárias para o jogo.
- A imagem abaixo apresenta um exemplo de como a interface poderia se parecer:



É fornecido junto com a definição do trabalho um exemplo de implementação de um jogo de cartas simples utilizando o JavaFX. O grupo pode utilizar livremente o código como base para a implementação deste trabalho. Algumas observações sobre o código:

- O arquivo ZIP fornecido no Material de Apoio é um projeto que utiliza o sistema Maven para realizar o build do aplicativo.
- Caso seja do interesse, o projeto já foi configurado como um projeto da IDE Eclipse. Mas o grupo pode utilizar o projeto em qualquer IDE que dê suporte ao Maven ou esse pode ser utilizado via linha de comando.
- A documentação oficial do Maven pode ser obtida em <https://maven.apache.org/>
- A documentação de como utilizar Maven com Visual Studio Code pode ser obtida em <https://code.visualstudio.com/docs/java/java-build>
- Um tutorial de como utilizar o Maven com o Eclipse pode ser obtido em <https://www.digitalocean.com/community/tutorials/maven-eclipse-ide>
- Para disparar o processo de build e execução do projeto utilize: **mvn clean compile javafx:run**

## ENTREGA E APRESENTAÇÃO

- Os trabalhos podem ser realizados individualmente ou em grupos de até quatro alunos.
- A implementação deve seguir as orientações dadas em aula quanto a convenções Java para nomes de identificadores e estrutura das classes.
- Não serão aceitos trabalhos com erros de compilação. Programas que não compilarem corretamente terão nota **ZERO**.
- Uma versão em formato “zip” deve ser entregue via sistema Moodle contendo a implementação feita (todas as pastas e os arquivos .java). Este arquivo deve ter o nome e sobrenome do(s) aluno(s), da seguinte forma: nome\_ultimosobrenome-nome\_ultimosobrenome.zip. Deve ser feito o upload deste arquivo na tarefa indicada para isto no Moodle até a data e horário especificados.
- Trabalhos entregues, mas não apresentados, terão sua nota anulada pelo professor. Durante a apresentação será avaliado o domínio da resolução do problema, podendo inclusive ser possível invalidar o trabalho quando constatada a falta de conhecimento sobre o código implementado.
- A cópia parcial ou completa do trabalho terá como consequência a atribuição de nota **ZERO** ao trabalho dos alunos envolvidos.
- O Fórum da Disciplina pode ser utilizado para discutir qualquer tema relacionado ao trabalho: sugestões, críticas, estratégias. Mas não é permitido postar código.
- A data e horário limite para entrega do trabalho é **28/11/2022** as **21:00**.
- A data de apresentação do trabalho é **28/11/2022**.