



Curso de Ciência da Computação

Atividades Avaliativas 4 componentes por grupo

Objetivo:

Realizar o desenvolvimento das soluções para os problemas e sistemas propostos utilizando o paradigma orientado a objetos.

Conceitos básicos:

- Classes e Objetos, Conjunto de Atributos, Comportamentos, Associação e Navegabilidade, Coleções, Herança e assuntos relacionados, *Greenfoot*

O que deve ser entregue:

- Para ambas as atividades deve-se realizar a gravação de um único vídeo (máximo 10 minutos) apresentado a implementação e execução. **Cada integrante** do grupo **deverá apresentar** alguma parte.
- Arquivos contendo os projetos e algoritmos em Java.

Avaliação:

A avaliação será composta por duas partes:

- Avaliação do material pedido no item “O que deve ser entregue:”
- Arguição INDIVIDUAL** dos componentes do grupo, no dia da entrega da atividade.
 - a. **Cada integrante** do grupo **deverá apresentar** alguma parte.
- A NOTA SERÁ INDIVIDUAL**, de acordo com os critérios apresentados pelo professor aos alunos da disciplina.

Importante:

- Data da arguição da atividade: **24 de Novembro de 2023.**
- Pode utilizar a classe InOut (InOut.java) para obter os dados via Janela de Diálogo.
- Sinta-se livre para propor melhorias, alterações, novas funcionalidades.** Converse com o professor para definir melhor o escopo da implementação.
- Valor da Atividade Avaliativa: 3,0 pontos.**

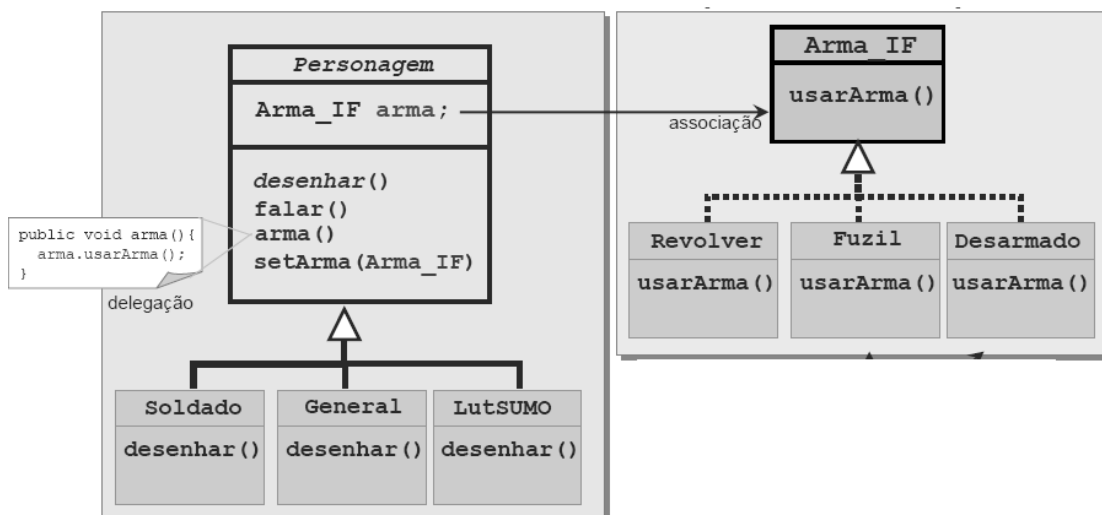
Atividade 01 – Contexto “Jogo com Greenfoot”.

Pesquisa, implementação e apresentação de um projeto de autoria própria/livre, a ser acompanhado pelo professor, usando o “*Greenfoot*”. Se for um caso de evolução de projeto já existente, deve-se ficar claro qual foi a contribuição realizada. Converse com o professor para definir melhor o escopo da implementação.

Atividade 02 – Contexto “Jogo Modelado”.

Crie um projeto de nome “*JogoModelado*”. Crie inicialmente dois pacotes, um para as armas e outro para os personagens. Estes pacotes deverão armazenar as classes e interfaces relativas ao seu contexto. Mais pacotes podem ser criados de acordo com a necessidade.

O diagrama de classes mostrado abaixo é uma base inicial para a atividade. Cada grupo deve incrementá-lo de acordo com as necessidades e as regras definidas para o jogo.



As seguintes funcionalidades no sistema do jogo modelado acima devem ser incorporadas:

- a. Inserir uma nova arma para os personagens: Faca.
- b. Inserir um novo personagem Mago que consegue soltar magia como arma para se defender.
- c. Inserir no jogo um novo personagem: DragaoAlado – verifique se todos os comportamentos definidos em Personagem são comuns ao dragão (um dragão deve falar?). Defina uma nova arma (a seu gosto) para o dragão (fogo, por exemplo)
- d. Inserir um novo método no jogo: correr(). Este método deve permitir que um personagem possa correr em caso de perigo. Verifique se todos os personagens podem correr (ou seja, este comportamento é comum a todos?). Ex.: um dragão sai na carreira?
- e. Inserir um novo método no jogo: voar(). Este método deve permitir que um personagem (em vez de correr) possa voar em caso de perigo. Verifique se todos os personagens são capazes de voar (ou seja, este comportamento é comum a todos?). Ex.: um lutador de sumô pode voar? E o dragão?

Crie uma classe que irá representar o jogo, contendo o método *main*. Execute o programa fazendo testes (atribuindo e trocando armas em cada personagem, por exemplo) e aplicando as regras de batalha, ou seja, faça a simulação de um jogo ou combate entre os personagens utilizando as regras definida pelo grupo.