Programação Orientada a Objetos

Em um projeto usando o Maven e considerando a imagem abaixo:

1. **(10%)** Criar a classe **Character** em um pacote chamado model. Defina os possíveis atributos que essa classe deve ter a partir dos personagens do jogo.
2. **(5%)** Implementar os métodos getters e setters da classe **Character**.
3. **(5%)** Criar uma classe executável chamada **MyApp** na pasta raiz da pasta java.
4. **(5%)** Instanciar os seguintes Characters:
   - Sonic.
   - Tails.
   - Monkey.
5. **(10%)** Criar a classe **Environment** em um pacote chamado model. Defina os possíveis atributos para esta classe. Perceba que os personagens estão todos situados em um ambiente (Hill Top Zone). O ambiente possui uma altura e uma largura fixa e os personagens estão em uma determinada posição no plano cartesiano.
6. **(5%)** Criar os getters e setters para a classe **Environment**.

7. **(5%)** Instanciar o objeto para o ambiente **Hill Top Zone**.
8. **(5%)** Adicionar os personagens **Tails**, **Sonic** e **Monkey**, no ambiente **Hill Top Zone**.
9. **(5%)** Perceba que o ambiente pode possuir outros tipos de objetos que não são personagens (e.g., caixas e rings). Para isso, crie uma classe Object e defina seus atributos.
10. **(5%)** Criar os getters e setters para a classe **Object**.
11. **(5%)** Istancie:
    - 3 rings.
    - 1 box.
12. **(5%)** Adicione esses objetos ao ambiente.
13. **(10%)** Os personagens podem se mover pelo ambiente. A movimentação é baseado na mudança de sua posição cartesiana considerando uma direção adotada. Faça um método em **Character**, que dado a entrada de uma direção (FRENTE, TRÁS, CIMA, BAIXO) mude o posicionamento do eixo correto em uma unidade.
14. **(10%)** Todos os personagens podem se mover, mas o comportamento de alguns personagens não são possíveis em alguns outros personagens, por serem particulares. Por exemplo, o personagem Tails pode voar, enquanto o Monkey pode escalar árvores. Já o personagem Sonic só pode pular, assim. Crie as classes específicas para cada personagem sobrecarregando os métodos pertinentes.
15. **(10%)** Adicione um método no ambiente que faça a identificação que um objeto colidiu com outro observando as seguintes condições:
    - Se o Sonic colidir com o Tails, nada acontece.
    - Se o Sonic colidir com um Ring, o Sonic coleta uma moeda.
    - Se o Tais colidir com um Ring, nada acontece.
    - Se o Monkey colidir com o Sonic e o Sonic estiver pulando, o Monkey morre. Caso o contrário, o Sonic morre e uma vida é tirada do sonic.

    Demonstre cada um dos casos com chamadas a este método.