

Fundamentos de Java

3-9: Abstração

Projeto:

Objetivos da Lição – Usar abstração

- Adicionar um parâmetro a um construtor
- Adicionar um efeito aleatório a um ator
- Acessar um método em uma subclasse de mundo

Instruções:

Abra o cenário da lição 8 (JF_V02_S03_L08PrjStudent).

Execute as seguintes tarefas:

1. Salve o cenário como JF_V02_S03_L09PrjStudent
2. Adicione uma propriedade à classe Bloco denominada velocidade de giro que armazena um valor inteiro
3. Crie um construtor na classe Bloco com a assinatura pública Block(int maxturnspeed)
4. Adicione código ao construtor Bloco que defina o campo de velocidade de giro como um valor aleatório entre – maxturnspeed e até +maxturnspeed.
5. Modifique o construtor Bloco de modo que se a velocidade de giro for definida como 0, ela seja alterada para 1.
6. Agora você modificará a classe RobotWorld. Atualize a linha - addObject(new Block(), 427, 145); para addObject(new Block(2), 427, 145);
7. Modifique a instrução turn(2) no método act da classe **block** para usar a variável turnspeed, em vez do valor 2.
8. Adicione uma propriedade à classe RobotWorld denominada int currentMaxTurnSpeed = 2;
9. Adicione uma propriedade à classe RobotWorld denominada int currentLevel = 1;
10. Crie um método no RobotWorld chamado setUpLevel com a assinatura de public void setUpLevel()
11. Programe o seguinte pseudocódigo no RobotWorld no método setUpLevel. Isso adicionará outro bloco giratório com uma velocidade aleatória ao mundo toda vez que aumentarmos um nível.
 - Se o nível for equivalente a 2
 - Adicione 1 a currentMaxTurnSpeed
 - Adicione um novo bloco na velocidade máxima do nível
 - Adicione 5 pedaços de pizza ao mundo

```

Se o nível for equivalente a 3
    Adicione 1 a currentMaxTurnSpeed
    Adicione um novo bloco na velocidade máxima do nível
    Adicione 5 pedaços de pizza ao mundo
Else if level equals 4
    //Elabore suas próprias informações aqui

```

12. Crie um método no RobotWorld denominado increaseLevel com a assinatura de public void increaseLevel()

13. Modifique o método increaseLevel de modo que ele incremente o campo currentLevel em 1.

14. Adicione uma chamada ao método setUpLevel dentro de increaseLevel.

15. Adicione o código a seguir ao método da classe Robô detectHome quando o robô chegar em casa.

```

RobotWorld myworld = (RobotWorld)getWorld();
myworld.increaseLevel();

```

*Nota. getWorld() retorna uma referência a World Class Type, mas adicionamos nossos próprios métodos públicos e para acessá-los precisamos obter uma referência a RobotWorld. Então, abstraímos nosso Mundo como um tipo de RobotWorld. Em seguida, isso nos dará acesso às suas propriedades públicas e métodos.

16. Crie uma propriedade no Robô denominada temporizador do tipo int.

17. Em Robô, crie o valor de uma constante para o valor máximo do cronômetro. Ele deve estar no formato private final int MAXTIMER com o valor 1000.

18. No Robô, modifique o construtor para definir o cronômetro com o valor de maxtimer. (Aumente o valor de MAXTIMER caso queira definir mais tempo)

19. Crie um método denominado updateTimer() no Robô que diminua o valor do cronômetro em uma unidade. Adicione a chamada do método updateTimer() ao método act do Robô.

20. Modifique o updateTimer de modo que ele chame o método World showText() para exibir o tempo restante (70, 580).

21. Atualize updateTimer de modo que, se o cronômetro estiver em 0, usemos removeLife() para perder uma vida.

22. Crie um método public void denominado resetPosition que defina a localização do robô de volta à posição inicial.

23. Adicione uma chamada de método resetPosition ao método removeLife() para que o Robô sempre retorne à posição inicial quando perder uma vida.

24. Remova a chamada setLocation dos métodos detectBlockCollision() e detectWallCollision() porque agora ela será tratada pela chamada removeLife().

25. Adicione uma chamada de método resetPosition ao método detectHome(), em vez de usar setLocation().

26. Crie um método public void resetTimer() no Robô que defina o cronômetro com o valor MAXTIMER.

27. Adicione uma chamada ao método `resetTimer()` como a última linha da instrução `if` no método `updateTimer()`.
28. Modifique `detectHome` para que ele chame `resetTimer()` depois de ter passado de nível quando chegarmos à Casa.
29. Use o pseudocódigo a seguir:
 Set timer = timer + 200
 Modifique `eatPizza` de modo que, toda vez que uma pizza for comida, adicionemos 200 ao temporizador.
30. Compile seu cenário.
31. Salve o cenário.

Opções possíveis para ampliar o projeto

- Adicione um ator de movimento aleatório que redefinirá o robô quando você colidir
- Adicione outros obstáculos
- Adicione seus próprios personagens
- Faça seus próprios arquivos de som
- Adicione mais paredes

por exemplo

