Laser Tag Lab

Learning Target: I will write a class that uses object oriented program to solve a problem.

Part 1.

Write a class, LaserTagPlayer.java, that meets the following specifications.

* Instance variables
  + private String name
  + private int laserPower
  + private int shieldStrength
* Constructor that matches parameters
* setters and getters for all instance variables
* toString()

Part 2.

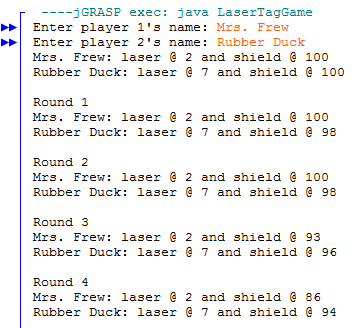
* Constructor that takes just the name
  + set the laserPower to a random number 1 – 10, and shieldStrength to 100
* public boolean isShieldDepleted() – if the shield strength is less than or equal to zero, return true. Otherwise, return false.
* public void attemptToTag(LaserTagPlayer that) – this LaserTagPlayer has a 50% chance of successfully tagging that LaserTagPlayer. If successful, reduce that LaserTagPlayer’s shield strength by this LaserTagPlayer’s laser power value.

Part 3.

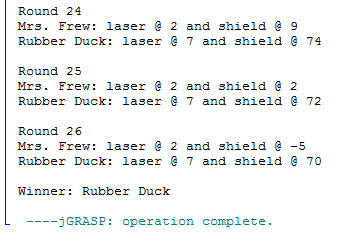
Make a new java file, LaserTagGame.java, with just a main method.

* Ask for two names using the Scanner.
* Make two LaserTagPlayer objects using the constructor that takes just a name.
* Next, station the participants against each other in an activity. The participants should take turns attempting to tag each other.
* The program should end when one participant’s shields are depleted, displaying the name of the participant whose shield strength happened to end in the positive range on this particular iteration of the activity as the winner.

See next page for example run.



…



|  |  |  |
| --- | --- | --- |
| Part 1 Rubric | Part 2 Rubric | Part 3 Rubric |
| * Indenting * Comment header * File is named LaserTagPlayer.java * Private instance variables * Constructor * Setters * Getters * Additional methods * Main: 2 pts | * Default constructor: 2 pts * isShieldDepleted: 1 pts * attemptToTag: 2 pts | * Indenting * Comment header * File is named LaserTagGame.java * Only has 1 method, main() * Uses a Scanner * Constructs 2 LaserTagPlayer objects * Calls attemptToTag * Calls isShieldDepleted * Loops until winner * Prints out winner |