package Swiggy\_Project;  
  
public class Player {  
 private int health;  
 private int strength;  
 private int attack;  
  
 public Player(int health, int strength, int attack) {  
 this.health = health;  
 this.strength = strength;  
 this.attack = attack;  
 }  
  
 public int getHealth() {  
 return health;  
 }  
  
 public void setHealth(int health) {  
 this.health = health;  
 }  
  
 public int getStrength() {  
 return strength;  
 }  
  
 public int getAttack() {  
 return attack;  
 }  
  
 public boolean isAlive() {  
 return this.health > 0;  
 }  
  
 public void defend(int damage) {  
 this.health -= damage;  
 if (this.health < 0) {  
 this.health = 0;  
 }  
 }  
  
 @Override  
 public String toString() {  
 return "Player{" +  
 "health=" + health +  
 ", strength=" + strength +  
 ", attack=" + attack +  
 '}';  
 }  
}

package Swiggy\_Project;  
  
import java.util.Random;  
  
public class Dice {  
 private static final int *SIDES* = 6;  
 private Random random;  
  
 public Dice() {  
 this.random = new Random();  
 }  
  
 public int roll() {  
 return random.nextInt(*SIDES*) + 1;  
 }  
}

package Swiggy\_Project;  
  
public class Arena {  
 private Player player1;  
 private Player player2;  
 private Dice dice;  
  
 public Arena(Player player1, Player player2) {  
 this.player1 = player1;  
 this.player2 = player2;  
 this.dice = new Dice();  
 }  
  
 public void startFight() {  
 Player attacker = player1.getHealth() <= player2.getHealth() ? player1 : player2;  
 Player defender = attacker == player1 ? player2 : player1;  
  
 while (player1.isAlive() && player2.isAlive()) {  
 executeTurn(attacker, defender);  
 if (defender.isAlive()) {  
 Player temp = attacker;  
 attacker = defender;  
 defender = temp;  
 }  
 }  
  
 System.*out*.println("Fight over!");  
 System.*out*.println(player1);  
 System.*out*.println(player2);  
 }  
  
 private void executeTurn(Player attacker, Player defender) {  
 int attackRoll = dice.roll();  
 int defenseRoll = dice.roll();  
  
 int attackDamage = attacker.getAttack() \* attackRoll;  
 int defenseDamage = defender.getStrength() \* defenseRoll;  
  
 int damageToDefender = Math.*max*(0, attackDamage - defenseDamage);  
  
 defender.defend(damageToDefender);  
  
 System.*out*.println(attacker + " attacked with roll " + attackRoll + " causing damage " + damageToDefender);  
 System.*out*.println(defender + " defended with roll " + defenseRoll + " and now has health " + defender.getHealth());  
 }  
}

package Swiggy\_Project;  
  
public class main {  
 public static void main(String[] args) {  
 Player playerA = new Player(50, 5, 10);  
 Player playerB = new Player(100, 10, 5);  
  
 Arena arena = new Arena(playerA, playerB);  
 arena.startFight();  
 }  
}

package Swiggy\_Project;  
  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
  
public class ArenaTest {  
 @Test  
 public void testPlayerInitialization() {  
 Player player = new Player(100, 10, 15);  
 *assertEquals*(100, player.getHealth());  
 *assertEquals*(10, player.getStrength());  
 *assertEquals*(15, player.getAttack());  
 }  
  
 @Test  
 public void testDiceRoll() {  
 Dice dice = new Dice();  
 for (int i = 0; i < 100; i++) {  
 int roll = dice.roll();  
 *assertTrue*(roll >= 1 && roll <= 6);  
 }  
 }  
  
 @Test  
 public void testPlayerDefend() {  
 Player player = new Player(100, 10, 15);  
 player.defend(30);  
 *assertEquals*(70, player.getHealth());  
  
 player.defend(80);  
 *assertEquals*(0, player.getHealth());  
 }  
  
 @Test  
 public void testFight() {  
 Player playerA = new Player(50, 5, 10);  
 Player playerB = new Player(100, 10, 5);  
  
 Arena arena = new Arena(playerA, playerB);  
 arena.startFight();  
  
 *assertTrue*(!playerA.isAlive() || !playerB.isAlive());  
 }  
}