**Officer class**

/\*\*

\*This class creates an officer object and provides methods for accessing the information.

\*@author Paul Armstrong 3537294

\*\*/

public class Officer{

/\*\*

\*The name of the officer.

\*\*/

private String name;

/\*\*

\*The rank of the officer.

\*\*/

private String rank;

/\*\*

\*The strength value of the officer.

\*\*/

private int strength;

/\*\*

\*Constructs a new officer object by taking in a name, rank, and strength value.

\*@param nameIn The name of the officer.

\*@param rankIn The rank of the officer.

\*@param strengthIn The strength value of the officer.

\*\*/

public Officer(String nameIn, String rankIn, int strengthIn){

name = nameIn;

rank = rankIn;

strength = strengthIn;

}

/\*\*

\*Accessor to get the strength of the officer alone.

\*@return The strength value fo the officer.

\*\*/

public int getStrength(){

return strength;

}

/\*\*

\*Accessor to get all of the officers recorded info.

@return The officers name, rank and strength value.

\*\*/

public String toString(){

return rank + " " + name + "(" + strength + ")";

}

}

**Starship class**

/\*\*

\*This class creates a Starship object and provides methods to retrieve and edit its information.

\*@author Paul Armstrong 3537294

\*\*/

public class Starship{

/\*\*

\*The name of the ship.

\*\*/

private String shipName;

/\*\*

\*The number of HP the ship has.

\*\*/

private int health;

/\*\*

\*The capitain of the ship.

\*\*/

private Officer captain;

/\*\*

\*The first officer of the ship.

\*\*/

private Officer firstOfficer;

/\*\*

\*The chief engineer of ship.

\*\*/

private Officer chiefEng;

/\*\*

\*Constructs a new starship object by taking in a name, health value and 3 officers.

\*@param nameIn The ships name.

\*@param healthIn The ships HP.

\*@param captainIn The capitain of the ship.

\*@param firstOfficer The first officer of the ship.

\*@param chiefEng The chief engineer of the ship.

\*\*/

public Starship(String nameIn, int healthIn, Officer captainIn, Officer firstOfficerIn, Officer chiefEngIn){

shipName = nameIn;

health = healthIn;

captain = captainIn;

firstOfficer = firstOfficerIn;

chiefEng = chiefEngIn;

}

/\*\*

\*Accessor method for the ships data.

\*@return Ships name, HP and crew.

\*\*/

public String toString(){

int strength = (captain.getStrength() \* 3) + firstOfficer.getStrength() + (chiefEng.getStrength() \* 2);

return shipName + "(" + health + "HP): " + captain + ", " + firstOfficer + ", " + chiefEng + ", " + strength;

}

/\*\*

\*Accessor to determine teh damage delt to the ship.

\*@param attackpoints The strength of the attcking ship.

\*@return Ships name, current HP value and damaged received.

\*\*/

public String getsAttacked(int attackpoints){

int strength = (captain.getStrength() \* 3) + firstOfficer.getStrength() + (chiefEng.getStrength() \* 2);

if((strength/2) >= attackpoints){

health -= 0;

}

else{

health -= (attackpoints - (strength/2));

}

return shipName + "(" + health + "HP): " + "received " + (strength - (attackpoints/2)) + " damage";

}

/\*\*

\*Accessor to determine whether or not a ship has been destroyed.

\*@return If the ship is destroyed.

\*\*/

public boolean isDestroyed(){

boolean destroyed;

if(health <= 0){

destroyed = true;

}

else{

destroyed = false;

}

return destroyed;

}

/\*\*

\*Accessor method to simulate a battle and then print out the results.

\*@param underAttack The ship under attack.

\*@return The outcome of the battle.

\*\*/

public String attackShip(Starship underAttack){

int strength = (captain.getStrength() \* 3) + firstOfficer.getStrength() + (chiefEng.getStrength() \* 2);

return underAttack.getsAttacked(strength);

}

}

**Astromachy class**

public class Astromachy {

public static void main(String[] args){

System.out.println("Ensigns Paul Armstrong reporting to the Captain: ");

System.out.println();

//Train officers

Officer picard = new Officer("Jean-Luc Picard", "Captain", 4);

Officer riker = new Officer("William Riker", "Commander", 3);

Officer data = new Officer("Data", "Lt. Commander", 5);

Officer tomalak = new Officer("Tomalak", "Commander", 5);

Officer sela = new Officer("Sela", "Lt. Commander", 3);

Officer terrh = new Officer("Terrh Letanr", "Ensign", 4);

//Build and man starships

Starship ship1 = new Starship("Enterprise E", 50, picard, riker, data);

Starship ship2 = new Starship("Tal", 48, tomalak, sela, terrh);

//Battle

System.out.println("----------B A T T L E 1 ------------");

System.out.println(ship1);

System.out.println("VS");

System.out.println(ship2);

Starship winner;

/\*

\* Lt.Commander Panos' orders:

\* Write the code for the first battle here.

\* Ensure the reference variable winner

\* points to the winning Starship object

\* after the battle is over

\*/

winner = ship1;

while((ship1.isDestroyed() == false) && (ship2.isDestroyed() == false)){

ship1.attackShip(ship2);

System.out.println(ship1.attackShip(ship2));

if(!ship2.isDestroyed()){

ship2.attackShip(ship1);

}

System.out.println(ship2.attackShip(ship1));

}

if(ship1.isDestroyed()){

winner = ship2;

}

System.out.println("----------W I N N E R 1 ------------");

System.out.println(winner.toString());

/\*

\* Lt.Commander Panos' orders:

\* Create three new Officers with names and strengths of your choice.

\* Create a new Starship with name and healthpoints of your choice and

\* assign the three new Officers you created on it.

\* Simulate a second battle against the winner of the first battle.

\* The previous winner will fire first and retain any damage it

\* had collected from the previous battle.

\* As before, print similar messages for the report

\*/

Officer Brynn = new Officer("Brynn", "Captain", 4);

Officer Kassidy = new Officer("Kassidy", "Cheif Engineer", 6);

Officer Benedict = new Officer("Benedict", "Commander", 10);

Starship ship3 = new Starship("Unicorn Palace", 70, Brynn, Kassidy, Benedict);

System.out.println("----------B A T T L E 2 ------------");

System.out.println(winner);

System.out.println("VS");

System.out.println(ship3);

while((winner.isDestroyed() == false) && (ship3.isDestroyed() == false)){

winner.attackShip(ship3);

System.out.println(winner.attackShip(ship3));

if(!ship3.isDestroyed()){

ship3.attackShip(winner);

System.out.println(ship3.attackShip(winner));

}

}

if(winner.isDestroyed()){

winner = ship3;

}

System.out.println("----------W I N N E R 2 ------------");

System.out.println(winner.toString());

}

}

**Output**

