public class Spaceship {

private String shipName;

private int passengerCount;

private int maxPassengers;

public Spaceship(String shipName, int maxPassengers) {

this.shipName = shipName;

this.maxPassengers = (maxPassengers >= 0)?maxPassengers:100;

/\* Code above is equal to:

if (maxPassengers >= 0)

this.maxPassengers = maxPassengers;

else

this.maxPassengers = 100; \*/

}

public void setShipName(String shipName) {

this.shipName = shipName;

}

public String getShipName() {

return this.shipName;

}

public int getMaxPassengers() {

return this.maxPassengers;

}

public void board(int boardingPassengers) {

if ( this.passengerCount + boardingPassengers <= maxPassengers) {

this.passengerCount += boardingPassengers;

}

}

public void leave(int leavingPassengers) {

if ( this.passengerCount - leavingPassengers >= 0 ) {

this.passengerCount -= leavingPassengers;

}

}

public int getPassengerCount() {

return this.passengerCount;

}

public void transferTo(Spaceship newShip) {

if (newShip == null)

return;

if (this.getPassengerCount() + newShip.getPassengerCount() <= newShip.getMaxPassengers()) {

newShip.board(this.getPassengerCount());

this.leave(this.getPassengerCount());

}

}

// Method toString() will automatically be called when the object is called

// inside a System.out.println() instruction.

public String toString() {

String out = "Ship name: " + this.getShipName() + "\n" +

"Passenger count: " + this.getPassengerCount() + "\n" +

"Max passengers: " + this.getMaxPassengers();

return out;

}

}

Informática II – Prepa Tec Campus Eugenio Garza Lagüera  
Actividad 6: Spaceship

Sección I.: De acuerdo con la clase “Spaceship”, responde los siguientes ejercicios. Considera que cada problema es independiente del anterior.

**Problema 1:**

Spaceship s1 = **new** Spaceship("Battlecruiser", 1000);

s1.board(250);

s1.leave(100);

s1.board(5);

System.**out**.println(s1);

**Problema 2:**

Spaceship Hyperion = **new** Spaceship("", 2500);

Hyperion.setShipName("Hyperion");

Hyperion.board(450);

Hyperion.board(3655);

Spaceship TheMilano = **new** Spaceship("The Milano", 10);

TheMilano.setShipName("Milano");

Hyperion = TheMilano;

TheMilano.board(1);

Hyperion.board(4);

System.**out**.println(TheMilano);

System.**out**.println(Hyperion);

**Problema 3:**

Spaceship Normandy, Andromeda;

Normandy = **new** Spaceship("Normandy", 30);

Andromeda = **new** Spaceship("Andromeda", 500);

Andromeda = Normandy;

Normandy.setShipName("SSV Normandy SR-1");

Normandy.board(10);

Andromeda.board(10);

Andromeda.setShipName("Andromeda ISS");

Andromeda.board(20);

System.**out**.println(Andromeda);

System.**out**.println(Normandy);

**Problema 4**

Spaceship Voyager1, Apollo;

Voyager1 = **new** Spaceship("Voyager 1", 5);

Apollo = **new** Spaceship("Apollo 11", 10);

Voyager1.board(3);

Voyager1.transferTo(Apollo);

Voyager1.board(5);

Voyager1.transferTo(Apollo);

Voyager1.board(2);

Voyager1.transferTo(Apollo);

System.**out**.println(Voyager1);

System.**out**.println(Apollo);

Sección II: Codifica los siguientes problemas:

Problema 1. Escribe un método de instancia público para la clase Spaceship llamado equals que compare el objeto que lo invoca contra un objeto recibido como parámetro de entrada. Si los dos objetos son idénticos (el contenido de sus variables de instancia es igual), deberá retornar TRUE. Si el contenido de sus variables de instancia es diferente, retornará FALSE. Tip: Revisa el método transferTo() de la clase Spaceship.

Spaceship nave1 = new Spaceship(“Nave1”, 10);  
Spaceship nave2 = new Spaceship(“Nave2”, 20);  
boolean flag = nave1.equals(nave2); //false

public boolean equals(Spaceship otherShip) {

}

Problema 2. Escribe un método público para la clase Spaceship llamado clone, que devuelva un nuevo objeto de la clase Spaceship con el mismo contenido que el objeto que la llama. Debe tener la siguiente firma:

Spaceship nave1 = new Spaceship(“Nave1”, 10);  
Spaceship nave2 = nave1.clone();

public Spaceship clone( ) {

}