```
/*Askhsh 2)a: The contract of the ADT Trampala(Τραμπάλα)*/

public class Trampala {
  //The class Trampala has a set of three values. The value K represents
  //the maximum weight each new Trampala the user creates can hold. The
  //values leftS and rightS,represent the two sides of every new
  //Trampala.
  private ...;

  //////Constructors//////
  //Constructor: Trampala
  //Precondition:
  //The weight that is defined by the user must not be less than
  //zero. If it is an exception will handle it.
  //Postconditions:
  //Constructs a valid Trampala instance with the given weight
  //and with the default situation of a Trampala as empty.
   public Trampala(int newWeight);

   //////Transformers//////
   //Method:lLoad(Puts objects at the left side of the seesaw)
   //Preconditions:
   //The weight of the object must not be a negative number(exception)
   //Postconditions:
   public double lLoad(double newLoad);

   //Method:lUnload(Takes an object of the seesaw from the left side)
   //Preconditions:
   //The seesaw must not be already emtpy
   //Postconditions
   public double lUnload(double oLoad);

    //Method:rLoad(Puts objects at the right side of the seesaw)
   //Preconditions:
   //The weight of the object must not be a negative number(exception)
   //Postconditions:
   public double rLoad(double newLoad);

   //Method:rUnload(Takes an object of the seesaw from the right side)
   //Preconditions:
   //The seesaw must not be already emtpy
   //Postconditions
   public double rUnload(double oLoad);

   //////Accesors//////
   //Method:loadExistence(Checks if there is an object on the seesaw or
   //not).
   //Precondition:
   //Postconditon:
   //Returns -1 if the seesaw is empty, 1 if it has an object only
   //on one of the sides(right or left) and 0 if it has an object on both
   //sides.
   public int loadExistence();

   //Method:tCondition(Checks if the seesaw is broken or not)
   //Preconditions:
   //Postcondition:for each possible condition it returns the
corresponding message
```

```
    //and if the seesaw breaks an exception handles it.
    public String tCondition();

    //Method:totalWeight
    //Preconditions:
    //Postcontitions:returns the total weight of the onjects on the seesaw
    public double totalWeight();

    //////Observers//////
    //Method:Balance
    //Preconditions:
    //Postcoditions:If the weight of the right side is equal to the weight
    //of the left side the method returns true.
    public boolean Balance();
}
```