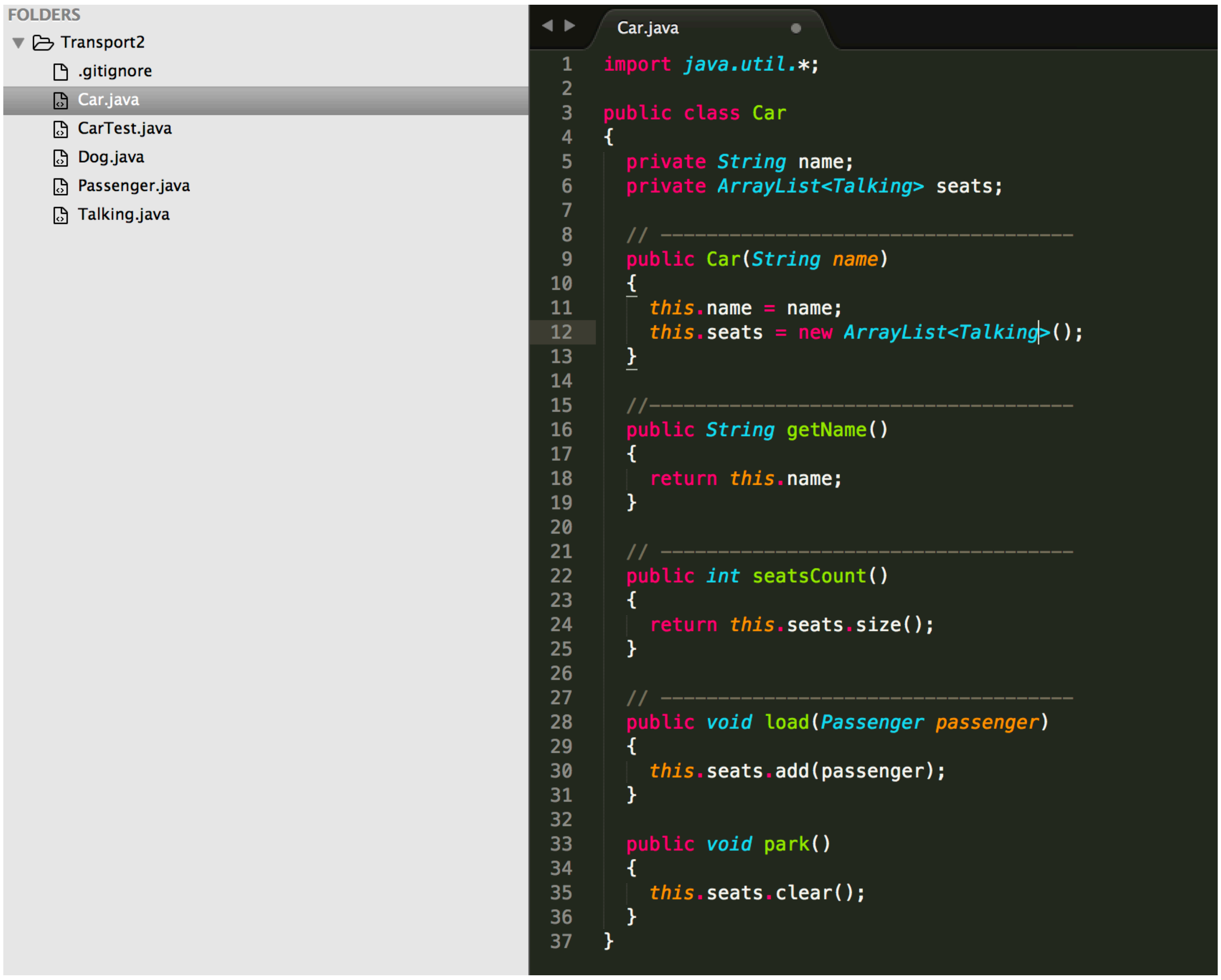
I.T.f

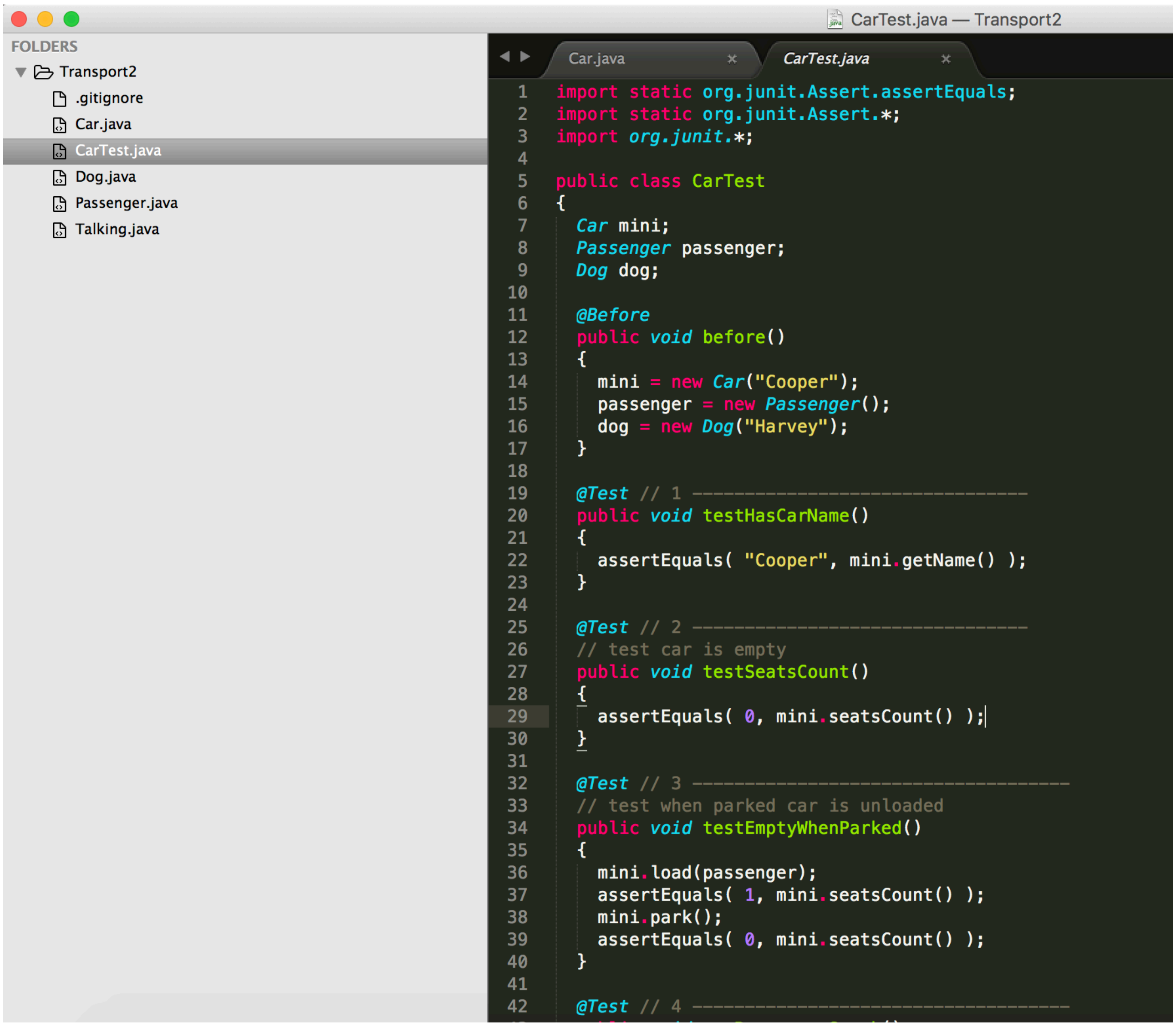
Polymorphism Example

- Carry out a Polymorphism task(T .1) or demonstrate use of Polymorphism in a program you have written.

Polymorphism is the ability of an object to take on many forms. The most common use of polymorphism in OOP occurs when a parent class reference is used to refer to a child class object

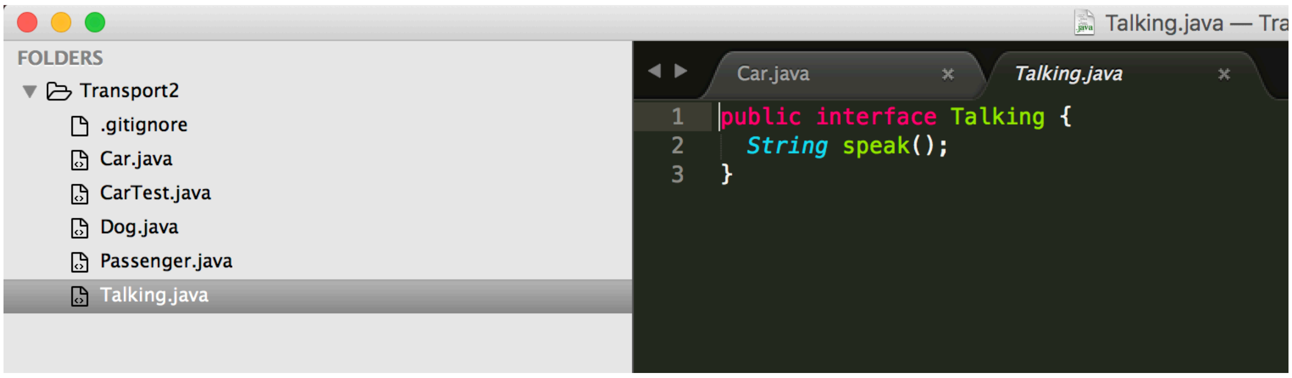


```java
import java.util.*;

public class Car
{
  private String name;
  private ArrayList<Talking> seats;

  // ------------------------------------
  public Car(String name)
  {
    this.name = name;
    this.seats = new ArrayList<Talking>();
  }

  //-------------------------------------
  public String getName()
  {
    return this.name;
  }

  // ------------------------------------
  public int seatsCount()
  {
    return this.seats.size();
  }

  // ------------------------------------
  public void load(Passenger passenger)
  {
    this.seats.add(passenger);
  }

  public void park()
  {
    this.seats.clear();
  }
}
```

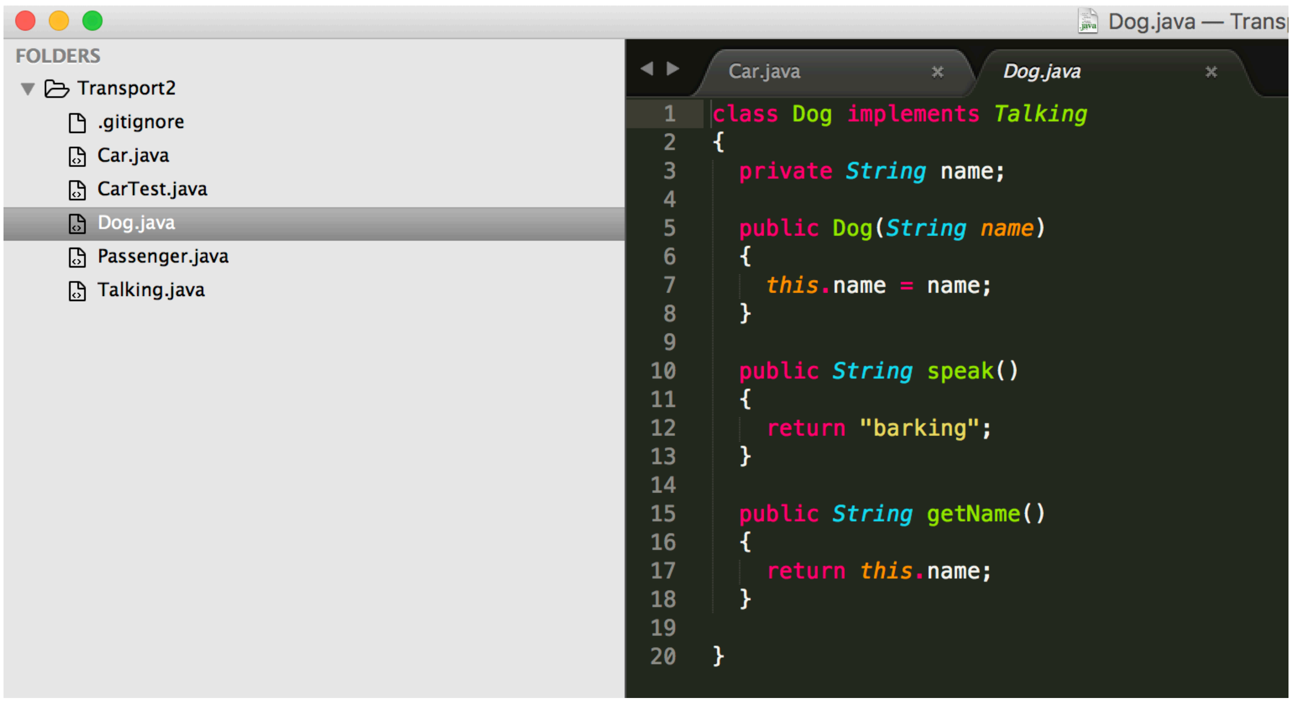


```java
import static org.junit.Assert.assertEquals;
import static org.junit.Assert.*;
import org.junit.*;

public class CarTest
{
  Car mini;
  Passenger passenger;
  Dog dog;

  @Before
  public void before()
  {
    mini = new Car("Cooper");
    passenger = new Passenger();
    dog = new Dog("Harvey");
  }

  @Test // 1 --------------------------------
  public void testHasCarName()
  {
    assertEquals( "Cooper", mini.getName() );
  }

  @Test // 2 --------------------------------
  // test car is empty
  public void testSeatsCount()
  {
    assertEquals( 0, mini.seatsCount() );
  }

  @Test // 3 -------------------------------------
  // test when parked car is unloaded
  public void testEmptyWhenParked()
  {
    mini.load(passenger);
    assertEquals( 1, mini.seatsCount() );
    mini.park();
    assertEquals( 0, mini.seatsCount() );
  }

  @Test // 4 -------------------------------------
```

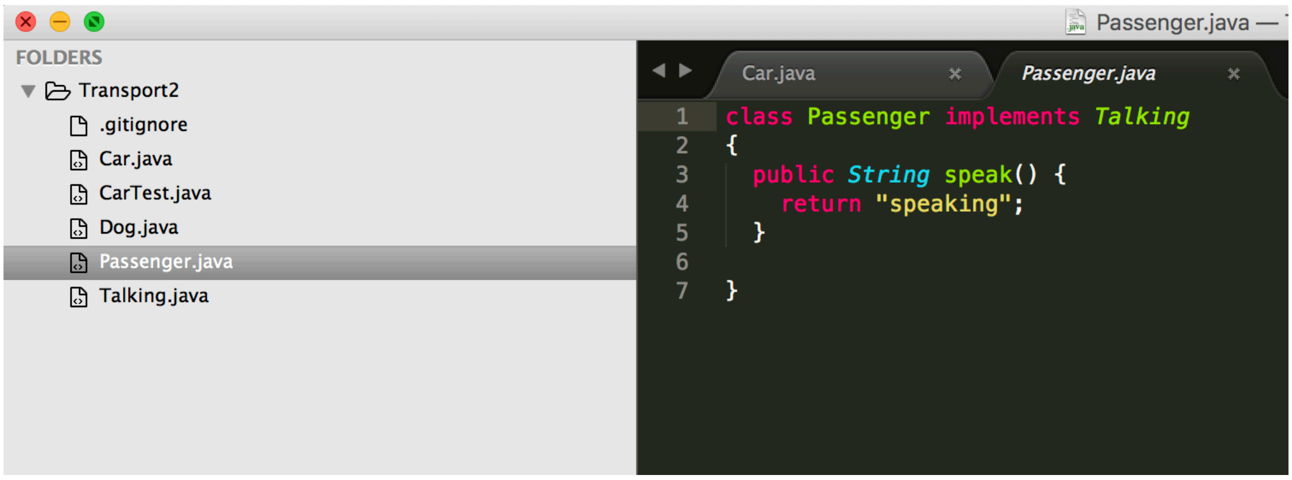
Tuesday, 21 February 2017



```java
public interface Talking {
  String speak();
}
```



```java
class Dog implements Talking
{
  private String name;

  public Dog(String name)
  {
    this.name = name;
  }

  public String speak()
  {
    return "barking";
  }

  public String getName()
  {
    return this.name;
  }

}
```



```java
class Passenger implements Talking
{
  public String speak() {
    return "speaking";
  }

}
```