When the program runs, here is a step-by-step explanation of what happens:

1. Memory Allocation:

Static Heap: This area of memory stores static variables and methods. In this program, there are no static variables or methods.

Stack: The stack memory stores local variables and method call information. It is used for method invocations and managing local variables. It keeps track of the execution flow. In this program, the stack will store variables related to the **main()** method.

Dynamic Heap: Objects are created in the dynamic heap. In this program, **Guitar** objects **obj1** and **obj2** will be created in the dynamic heap.

1. Object Creation:

The **Guitar** class is defined with private fields (**serialNumber**, **price**, **builder**, **model**, **backWood**, **topWood**) and public getter and setter methods.

In the **Tester** class, two **Guitar** objects are created: **obj1** using the default constructor and **obj2** using the parameterized constructor.

1. State of Objects:

**obj1**: The state of **obj1** is uninitialized because it is created using the default constructor. Its fields will have default values (**null** for **serialNumber** and **builder**, **0** for **price**, and **backWood**, **topWood**, **model**).

**obj2**: The state of **obj2** is initialized because it is created using the parameterized constructor with values passed for all fields.

1. Accessing Fields in Tester.java:

In the **Tester** class, all the fields of **obj1** and **obj2** are accessed and modified using setter and getter methods.

1. Current Object at "obj2.createSound();":

When the program reaches the line **obj2**.**createSound();,** **obj2** is the current object. The **createSound()** method is called on **obj2**.

1. Using "this" Keyword in main Method:

In the **main** method, the keyword **this** cannot be used to access the fields of **obj2** because this refers to the current object in the context of an instance method. However, the **main** method is a static method, so **this** is not available in it.

Overall, the program creates two **Guitar** objects, accesses their fields, and calls the **createSound**() method on **obj2**. The state of **obj1** is uninitialized, while **obj2** is initialized with the values provided in the constructor call.