
Guitar
- serialNumber
- price
- builder
- model
- backWood
- topWood
+ Guitar()
+ Guitar(serialNumber: String, price: int, builder: String, model: String, backWood: String, topWood: String)
+ getSerialNumber(): String
+ getPrice(): int
+ getBuilder(): String
+ getModel(): String
+ getBackWood(): String
+ getTopWood(): String
+ setSerialNumber(serialNumber: String): void
+ setPrice(price: int): void
+ createSound(): void
Tester
+ main(args: String[]): void

When the program runs, the following memory map can be observed:

Static Heap:

It contains the static variables and static methods of the classes used in the program. In this case, since there are no static variables or methods declared, the static heap is empty.

Stack:

It is used to store local variables, method parameters, and method calls. Each method call creates a new stack frame.

In the method main, the following variables are stored on the stack:

args - the command-line arguments array

obj1 - a reference variable for the Guitar object created with the default constructor

obj2 - a reference variable for the Guitar object created with the parameterized constructor

Dynamic Heap:

It is used to store dynamically allocated objects.

In this program, two Guitar objects (obj1 and obj2) are created on the dynamic heap.

Objects in the program:

The objects in the program are instances of the Guitar class.

obj1 and obj2 are the two objects created in the program.

State of obj1 and obj2:

Initially, obj1 is in its default state with all fields set to empty values (null for strings and 0 for the price field).

obj2 is created with specific values passed to the constructor, so it has the provided values for each field.

Accessing fields of obj1 in the class Tester.java:

In the class Tester.java, all fields of obj1 are accessed and modified using setter methods (setSerialNumber, setPrice, etc.). This allows accessing and modifying the fields in a controlled manner, ensuring proper encapsulation.

Current object when reaching "obj2.createSound();":

The current object when the program reaches the line obj2.createSound(); is obj2, as the method createSound is invoked on obj2.

Using "this" keyword to access fields of obj2 in the method main:

In the method main, the keyword "this" cannot be used to access fields of obj2 because main is a static method and "this" refers to the current object (which doesn't exist in a static context). Instead, you directly access the fields of obj2 using the reference variable obj2.