CLASSES: OBJECT VS STATIC (VARIABLES AND METHODS)

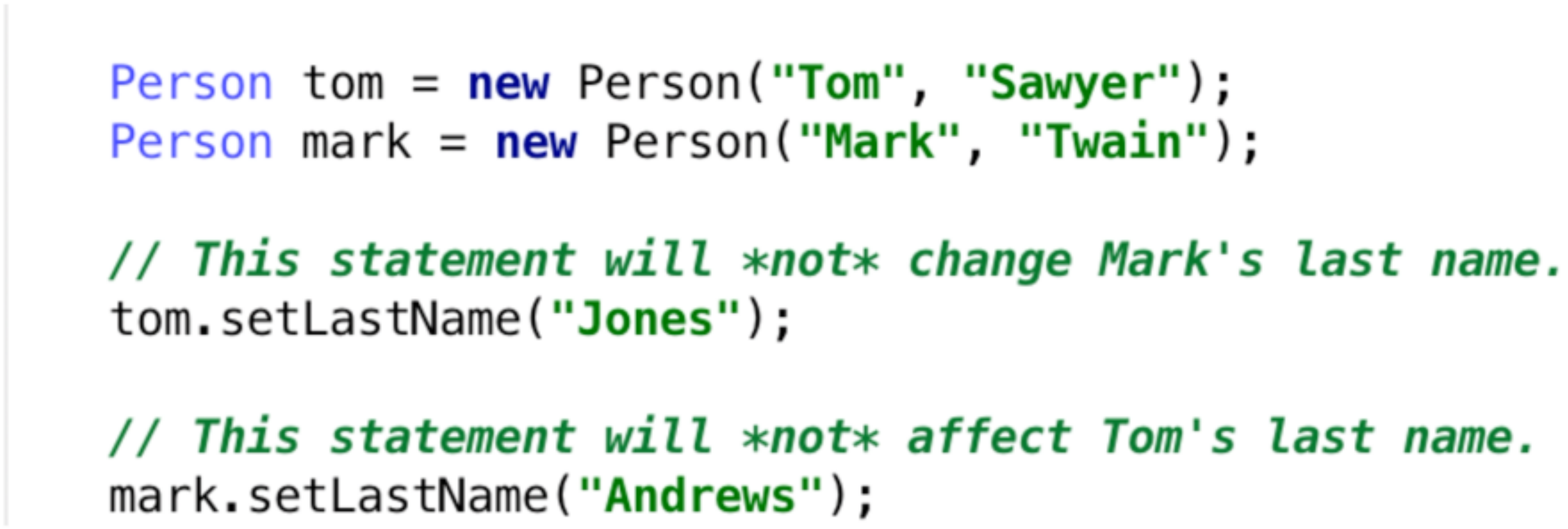
"OBJECTS ARE INSTANCES OF A CLASS"

"OBJECTS CONTAIN DATA (MEMBER VARIABLES) AS WELL AS FUNCTIONS (METHODS OR MEMBER FUNCTIONS)"

ITS IMPORTANT TO REALIZE THAT EACH OBJECT HAS ITS OWN VERSION OF THE METHODS AND OF THE MEMBER VARIABLES

THEREFORE, SUCH FUNCTIONS AND VARIABLES ARE CALLED OBJECT VARIABLES AND METHODS

Class instantiation, each object has its own version of the member variables



```java
Person tom = new Person("Tom", "Sawyer");
Person mark = new Person("Mark", "Twain");

// This statement will *not* change Mark's last name.
tom.setLastName("Jones");

// This statement will *not* affect Tom's last name.
mark.setLastName("Andrews");
```

"OBJECTS ARE INSTANCES OF A CLASS"

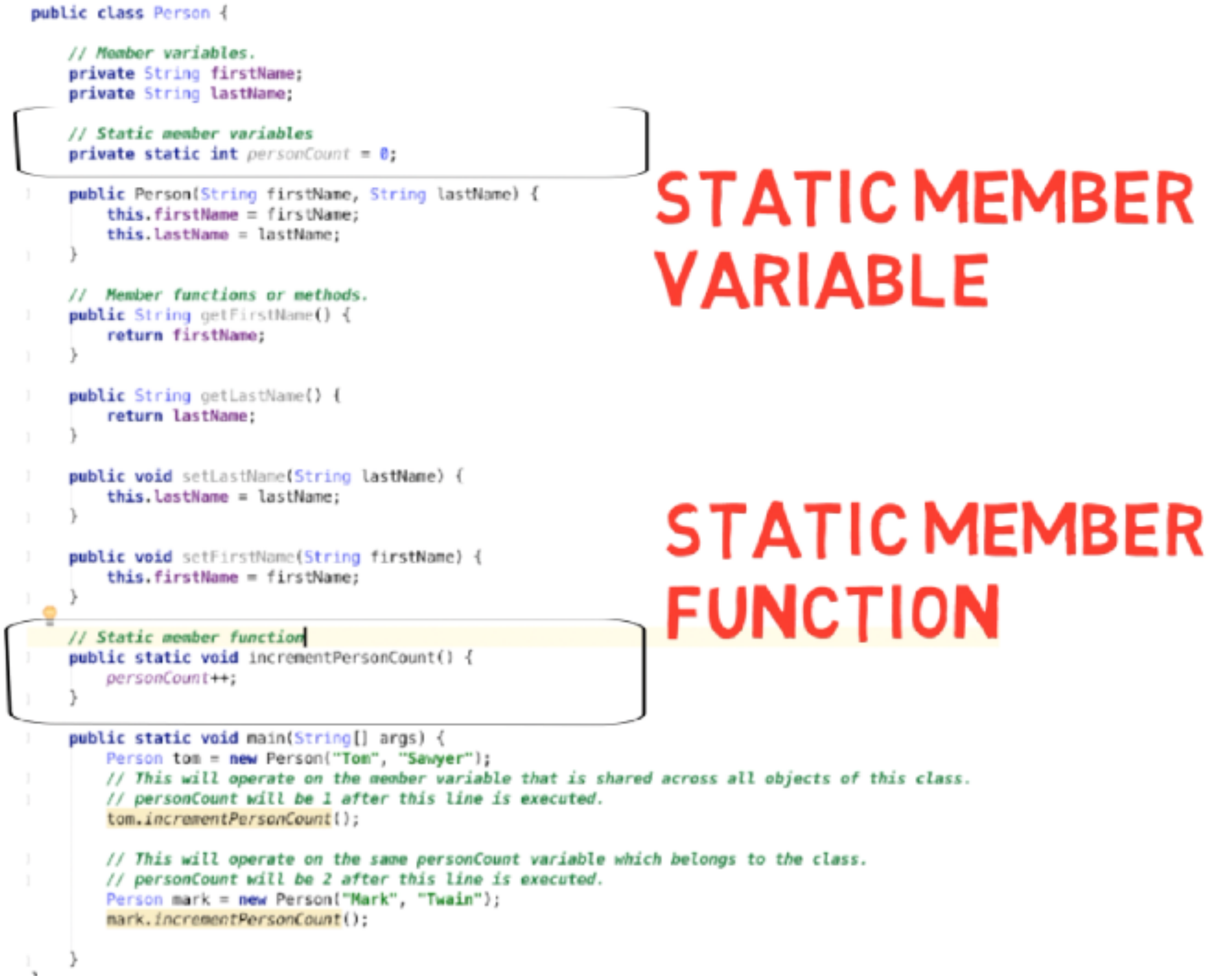
"OBJECTS CONTAIN DATA (MEMBER VARIABLES)
AS WELL AS FUNCTIONS (METHODS OR MEMBER FUNCTIONS)"

ITS IMPORTANT TO REALIZE THAT EACH OBJECT HAS ITS OWN VERSION OF THE METHODS AND OF THE MEMBER VARIABLES

THEREFORE, SUCH FUNCTIONS AND VARIABLES ARE CALLED OBJECT VARIABLES AND METHODS

THERE ARE ALSO, DIFFERENT TYPES OF VARIABLES AND FUNCTIONS THAT CAN BE CREATED ONCE-PER-CLASS I.E. SHARED BY ALL OBJECTS (INSTANCES) OF THAT CLASS

SUCH VARIABLES AND FUNCTIONS ARE CALLED CLASS VARIABLES OR METHODS, OR AS "STATIC" VARIABLES OR METHODS

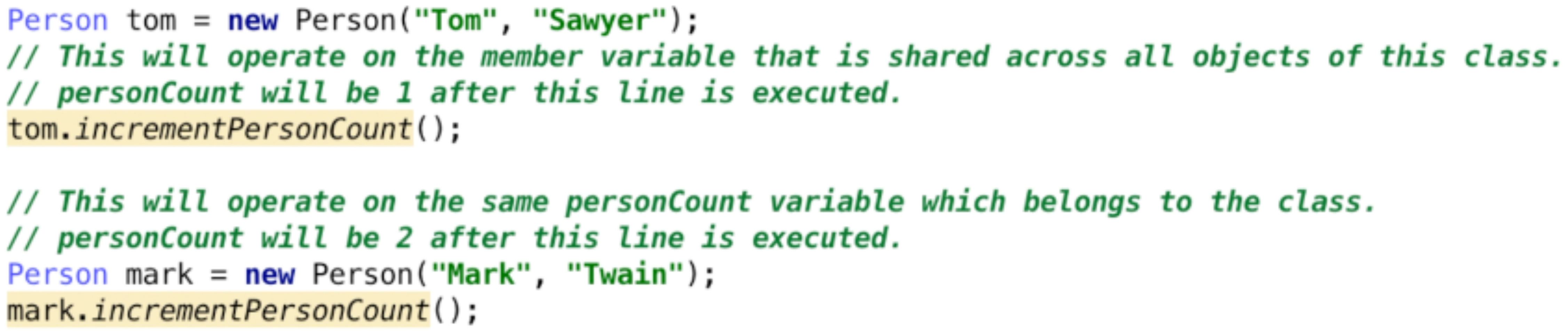


```java
public class Person {

    // Member variables.
    private String firstName;
    private String lastName;

    // Static member variables
    private static int personCount = 0;

    public Person(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    //  Member functions or methods.
    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    // Static member function
    public static void incrementPersonCount() {
        personCount++;
    }

    public static void main(String[] args) {
        Person tom = new Person("Tom", "Sawyer");
        // This will operate on the member variable that is shared across all objects of this class.
        // personCount will be 1 after this line is executed.
        tom.incrementPersonCount();

        // This will operate on the same personCount variable which belongs to the class.
        // personCount will be 2 after this line is executed.
        Person mark = new Person("Mark", "Twain");
        mark.incrementPersonCount();

    }
```

STATIC MEMBER VARIABLE

STATIC MEMBER FUNCTION

Updating static variables updates the value for all objects



```
Person tom = new Person("Tom", "Sawyer");
// This will operate on the member variable that is shared across all objects of this class.
// personCount will be 1 after this line is executed.
tom.incrementPersonCount();

// This will operate on the same personCount variable which belongs to the class.
// personCount will be 2 after this line is executed.
Person mark = new Person("Mark", "Twain");
mark.incrementPersonCount();
```