// Copyright 2003 Bill Campbell and Ethan Bolker

// Demonstrate the interplay between static members (fields and methods)
// and instance (non-static) members.

> java StaticDemo
0: counter = 1; objectField = 0
StaticDemo.classMethod() = 100
classMethod() = 100
1: counter = 2; objectField = 1
StaticDemo.classMethod() = 101
classMethod() = 101
2: counter = 3; objectField = 2
StaticDemo.classMethod() = 103
classMethod() = 103
3: counter = 4; objectField = 3
StaticDemo.classMethod() = 106
classMethod() = 106
4: counter = 5; objectField = 4
StaticDemo.classMethod() = 110
classMethod() = 110

A class variable is associated with the (one) class.

public class StaticDemo
{
    // Declare three (static) class variables
    // A class variable is associated with the (one) class.
    private static int counter = 0;
    private static int classVar = 0;
    private static Terminal terminal = new Terminal();

    int objectField = 0; // an instance variable; one per object

    // The constructor keeps track of how many StaticDemo objects
    // have been constructed.
    public StaticDemo(int objectFieldValue)
    {
        objectField = objectFieldValue; // set the instance variable
        counter++; // increment counter (counting the StaticDemos made)
    }

    // TODO: Define an accessor method for classVar.

    public void instanceMethod()
    {
        // Instance methods can refer to both instance variables
        // and class variables.

        terminal.println("counter = "+counter+
            "; objectField = "+objectField);

        classVar = classVar + objectField;
    }

    public static int classMethod()
    {
        // Class methods may refer only to class variables
        // (and other class methods), as well as to
        // local variables.

        // What happens if we comment out the next line?
        int counter = 100;

        return counter + classVar;
    }

    public static void main(String[] args)
    {
        for (int i = 0; i < 5; i++) {
            StaticDemo sd = new StaticDemo(i);
            terminal.print(i+": ");
            sd.instanceMethod();
        }

        // classMethod() is equivalent to
        // StaticDemo.classMethod()
        terminal.println("StaticDemo.classMethod() = "+
            StaticDemo.classMethod());
        terminal.println("classMethod() = "+classMethod());
    }
}