Record Constructors come in three flavors

The Canonical, or Long constructor is the implicitly generated constructor. You can explicitly declare your own, which means the implicit one won't get generated. If you do declare your own, you must make sure fields all get assigned a value.

The Custom constructor is just an overloaded constructor. It must explicitly call the canonical constructor as it's first statement.

The Compact, or Short constructor is a special kind of constructor, used only on records. It's a succinct way of explicitly declaring a canonical constructor.

{LP} LearnProgramming
.academy

The compact constructor

You can't have both a compact constructor and an explicit canonical constructor.

This constructor is declared with no parentheses, so no arguments.

It has access to all the arguments of the canonical constructor. Don't confuse the arguments with the instance fields!

You can't do assignments to the instance fields in this constructor.

The implicit canonical constructor's assignments occur after the execution of this code.

The Java Class File Disassembler

This is **javap**.

It lists class members, by default just public and protected members in the class file.

This helps us 'see' implicit code in the compiled class file.