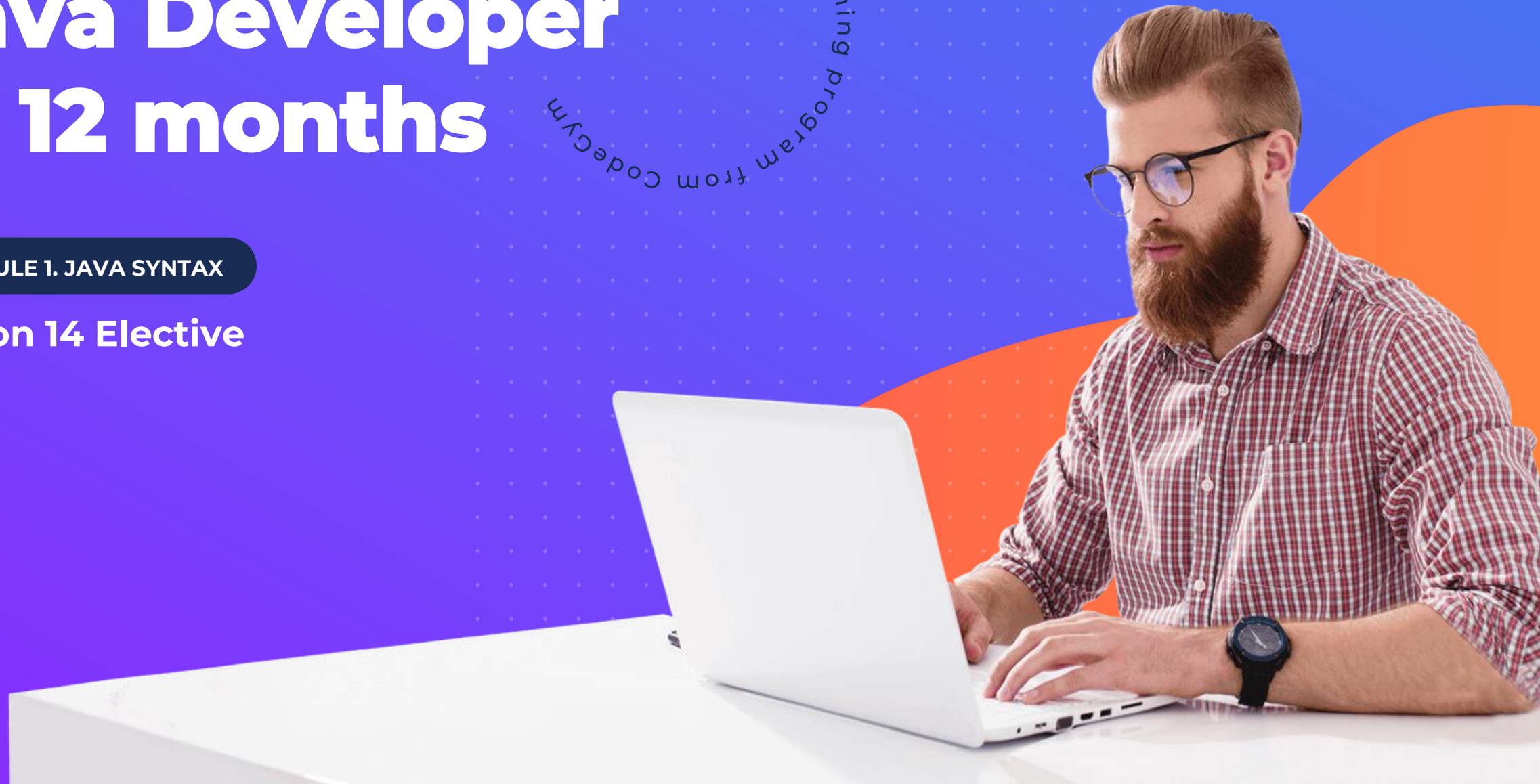
CODEGYM

Java Developer in 12 months

Mentor-supported training program from CodeGym

MODULE 1. JAVA SYNTAX

Lesson 14 Elective



CODEGYM

Lesson plan

- Object lifecycle
- Loading a class

CODEGYM

Object lifetime

Garbage Collector (GC)
In Java, the deletion of objects is completely automated. The Java machine itself handles the deletion of objects.

This process is called garbage collection, and the mechanism that collects garbage is called the garbage collector (GC).

Its job is to remove objects that are no longer used in the program. In other words, it frees up memory in the JVM for other objects.

An object exists from the moment it is created until it is destroyed by the GC. From the moment an object is created, it can be used in a program as long as there is at least one reference to the object, i.e. the object is accessible. That is, once there is not a single reference to an object, it still exists in memory for some time, but is not accessible.

CODEGYM

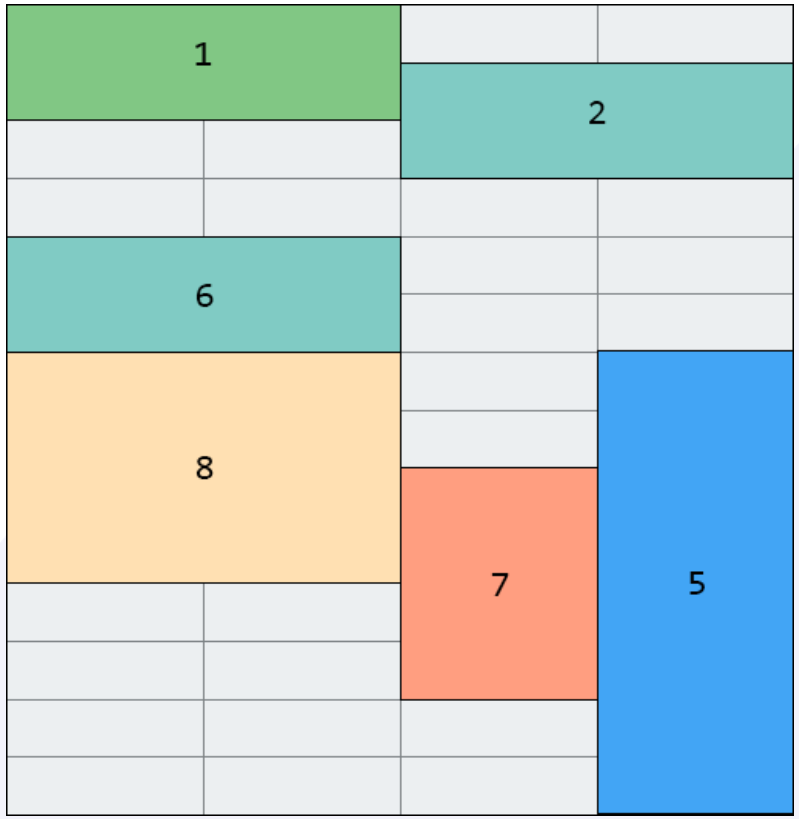
Object lifetime

Memory fragmentation

If you constantly create and delete objects, soon memory will be heavily fragmented: areas of occupied memory will be interspersed with areas of unoccupied memory.

Memory optimization (defragmentation)

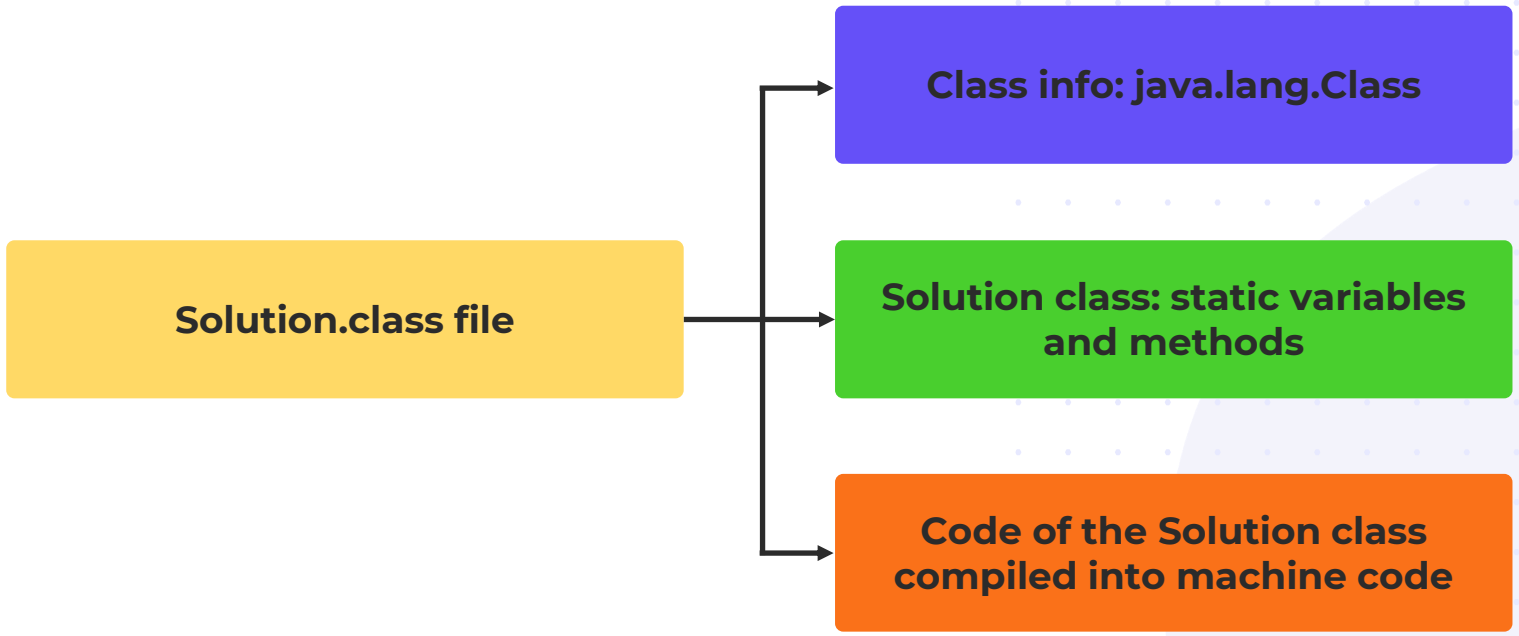
Memory is divided into two parts. All objects are created (and deleted) in just one half of memory. When it comes time to clean up the holes in memory, all objects in the first half are copied to the second half. But they are copied right next to each other so that there are no holes.



CODEGYM

Loading a class into memory

A class is an object. Actually, when a class is loaded into memory, three special "objects" are created:



CODEGYM

Getting objects

You can get the Class object for any type using a command like this:

```
Class name = ClassName.class;
```

Code	Note
`Class a = String.class;`	Get a Class object with information about the String class
`Class b = Object.class;`	Get a Class object with information about the Object class
`Class c = Integer.class;`	Get a Class object with information about the Integer class
`Class d = int.class;`	Get a Class object with information about the int type
`Class e = void.class;`	Get a Class object with information about the void type

You can also get a reference to a class description object from any object, since each object has the getClass() method, which it inherits from the Object class.

Code	Note
`Class a = "Hello".getClass();`	Same object as String.class
`Class b = new Integer().getClass();`	Same object as Integer.class
`Class c = Boolean.TRUE.getClass();`	Same object as Boolean.class

CODEGYM

Homework

MODULE 1. JAVA SYNTAX

Complete Level 15

CODEGYM

Answers to questions

MODULE 1. JAVA SYNTAX