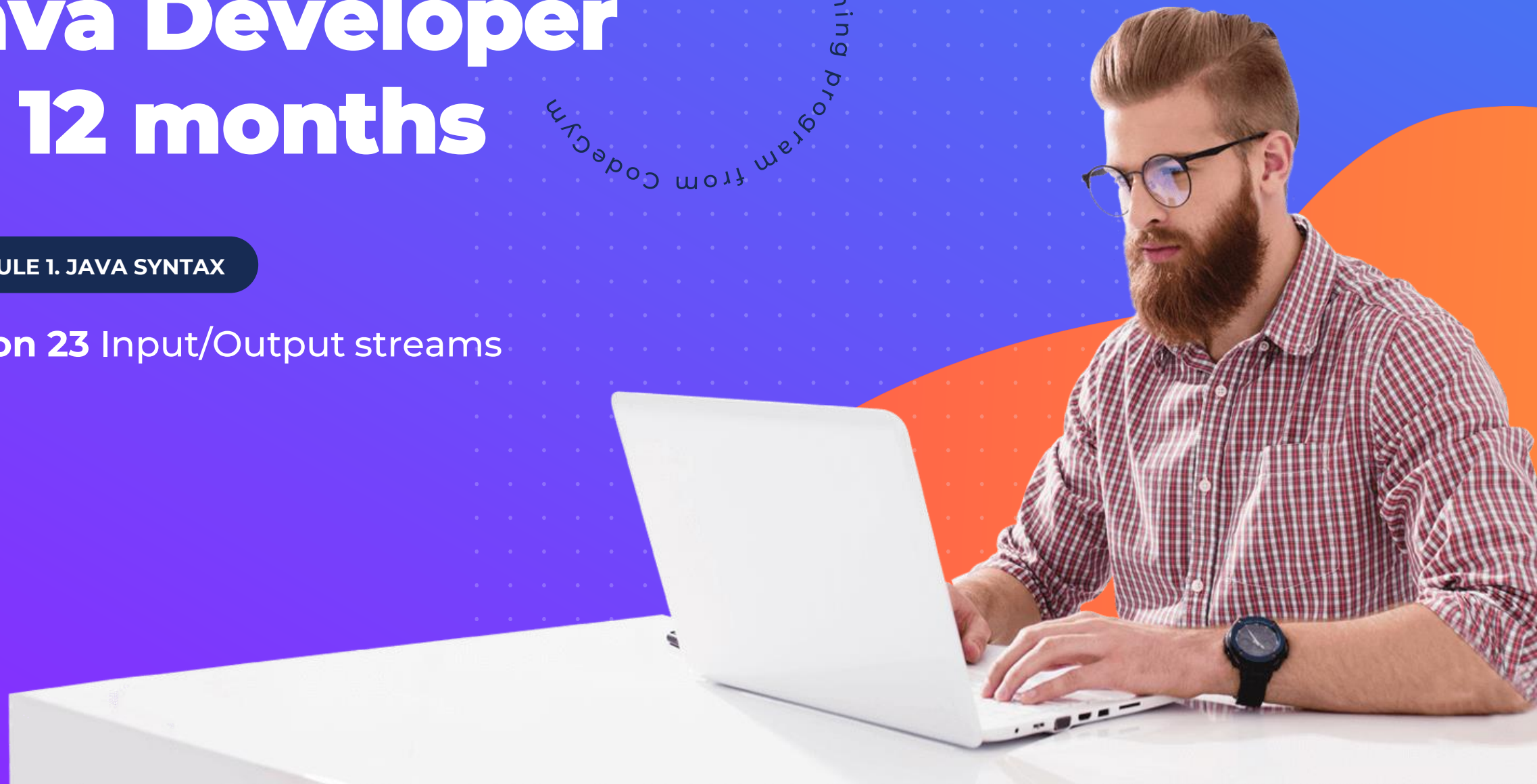
CODEGYM

Java Developer in 12 months

Mentor-supported training program from CodeGym

MODULE 1. JAVA SYNTAX

Lesson 23 Input/Output streams



CODEGYM

Lesson plan

- Streams for data input
- Chained streams
- Output streams
- BufferedWriter

CODEGYM

Streams for data input

In programming, the term "stream" is used to described the process of data exchange. It refers specifically to the flow of data.

Streams are a versatile tool. They allow the program to receive data from anywhere (incoming streams) and send data anywhere (outgoing streams).

They are divided into two types:

- Incoming stream (Input): is used to receive data
- Output stream (Output): is used to send data

CODEGYM

Byte streams and character streams

Bytes (and arrays of bytes) can be written to an **OutputStream**, and you can read bytes (and arrays of bytes) from an InputStream object.

The Reader class is an analogue of the InputStream class, but its **read()** method reads characters – char, not bytes.

The Writer class is an analogue of the OutputStream class, but with one difference: it works with characters (char) instead of bytes.

If we compare these four classes, we get the following picture:

Bytes (byte)	Characters (char)	
Reading data	InputStream	Reader
Writing data	OutputStream	Writer

CODEGYM

InputStream

Some methods from the InputStream class and all of its descendant classes

Methods	Description
int read()	Reads one byte from the stream
int read(byte[] buffer)	Reads the array of bytes from the stream
byte[] readAllBytes()	Reads all bytes from the stream
long skip(long n)	Skips n bytes in the stream (reads and throws out)
int available()	Checks how many bytes are left in the stream
void close()	Closes the stream

CODEGYM

InputStreamReader

The **InputStreamReader** class has the same methods as the Reader class, and they work in exactly the same way.

The main difference between the **InputStreamReader** and, for example, **FileReader** classes is where they read data from.

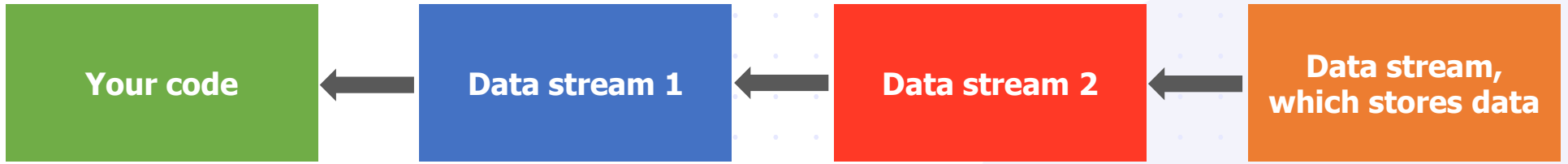
FileReader reads data from a file and **InputStreamReader** reads data from an **InputStream**.

CODEGYM

Chains of streams

Another interesting feature of streams is the ability to combine multiple streams into chains.

A stream can read data not only from the data source that stores it, but from another thread.



Reading from the console

An interesting fact: the Scanner class is nothing more than an incoming intermediate data stream that reads them from the System.in stream (which is also a data stream).

Here are two ways to read a line from the console:

Scanner class	BufferedReader and InputStreamReader classes
InputStream stream = System.*in*; Scanner console = new Scanner(stream); String line = console.nextLine();	InputStream stream = System.*in*; InputStreamReader reader = new InputStreamReader(stream); BufferedReader buff = new BufferedReader(reader); String line = buff.readLine();

CODEGYM

BufferedReader

The BufferedReader class, is a class that inherits from Reader and allows you to read characters.

However, the most interesting thing is that you also need to pass a stream to it as a data source, from which it can read characters - a stream that inherits from the Reader class.

Unlike the InputStreamReader, the BufferedReader class doesn't convert bytes to characters: it doesn't convert anything at all. BufferedReader buffers data.

CODEGYM

OutputStream

Some methods from the OutputStream class and all of its descendant classes

Methods	Description
void write(int b)	Writes one byte (not int) to stream.
void write(byte[] buffer)	Writes a byte array to stream
void write(byte[] buffer, off, len)	Writes part of a byte array to stream
void flush()	Writes to stream all of the data, which is available in the buffer
void close()	Closes the stream

CODEGYM

Writer

Some methods from the Writer class and all of its descendant classes:

Methods	Description
void write(int b)	Writes one character (not int) to stream.
void write(char[] buffer)	Writes an array of characters to the stream
void write(char[] buffer, off, len)	Writes part of a character array to the stream
void write(String str)	Writes a string to a stream
void write(String str, off, len)	Writes part of a string to the stream
void flush()	Writes to the stream all the data that is stored in the buffer
void close()	Closes the stream

CODEGYM

BufferedWriter

The BufferedWriter class in Java writes text to a character output stream by buffering the written characters to ensure efficient writing of characters, arrays (characters), and strings.

The BufferedWriter class has the following constructors:

```
BufferedWriter(Writer out)
BufferedWriter(Writer out, int size)
```

It takes the output stream to which it should write, as a parameter.
The size parameter specifies the size of the buffer.

CODEGYM

Homework

MODULE 1. JAVA SYNTAX

Complete Level 24

CODEGYM

Answers to questions

MODULE 1. JAVA SYNTAX