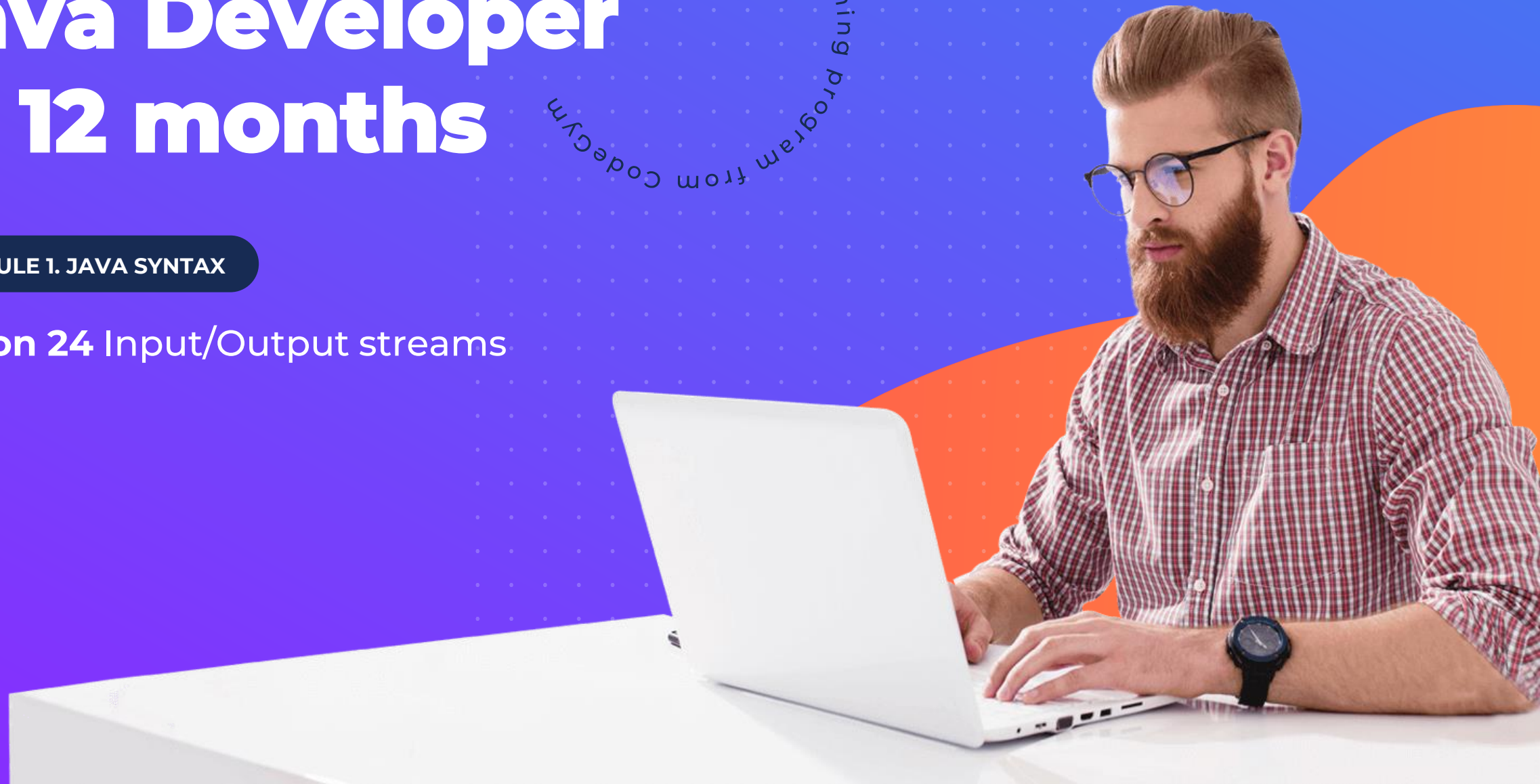
CODEGYM

Java Developer in 12 months

Mentor-supported training program from CodeGym

MODULE 1. JAVA SYNTAX

Lesson 24 Input/Output streams



CODEGYM

Lesson plan

- ByteArrayInputStream
- ByteArrayOutputStream
- Decorator

CODEGYM

ByteArrayInputStream

The ByteArrayInputStream class can be used to read an input array (of bytes).

It means than the class ByteArrayInputStream can turn array of bytes into InputStream.

Class ByteArrayInputStream is subclass of the class InputStream. That is why you can use ByteArrayInputStream as InputStream

CODEGYM

Creating of ByteArrayInputStream

If we want to use ByteArrayInputStream, first of all, we need to create instance of ByteArrayInputStream. To read the InputStream we need to pass the array of bytes to the constructor.

```
byte[] bytes = ... // array of bytes

ByteArrayInputStream byteArrayInputStream = new ByteArrayInputStream (byte) ;
```

CODEGYM

Methods of the ByteArrayInputStream

Methods	Description
int read()	Reads the next byte of data from this input stream.
int available()	Returns the number of remaining bytes that can be read (or skipped) from this input stream.
void reset()	Resets the buffer to the marked position. The marked position is 0 unless another position is marked or a different offset is specified in the constructor.
long skip(long n)	Skips n bytes of input from this input stream. Returns the number of bytes skipped (it may be less than n if we reach the end of the input stream).
void close()	Doesn't do anything.
void mark(int readAheadLimit)	Sets the mark field equal to the current position. If the reset method is called, then subsequent reading will start from that position. The readAheadLimit parameter is not used and does not affect the behavior of the method.

CODEGYM

ByteArrayOutputStream

The **ByteArrayOutputStream** class allows you to capture the data written to the stream into a byte array.

You can call the toByteArray() method and get all the written data in a byte array when the process of writing data into ByteArrayOutputStream is complete.

ByteArrayOutputStream can be useful in situations where there is a component that outputs its data into OutputStream, but the data is needed as an array of bytes.

CODEGYM

Creating of ByteArrayOutputStream

In order to use ByteArrayOutputStream, first of all, you need to create an instance of it.

Here's how you can do it:

```
ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream () ;
```

CODEGYM

Methods of the ByteArrayOutputStream

Methods	Description
int write(int b)	Is used to write one byte to ByteArrayOutputStream. The write() method of the ByteArrayOutputStream object accepts an int value containing one byte or a set of bytes for writing. Only the first byte of the int value is written. The rest is ignored.
int write(byte[] b, int off, int len)	Write bytes in the amount of len, starting at offset off, from the byte array to ByteArrayOutputStream
toByteArray()	Allows to get all bytes written as an array of bytes when writing is completed
void close()	Closes the stream

CODEGYM

Combined streams

You can integrate streams to achieve more complex input and output operations. For example, reading one byte at a time is very slow, so you can read a large block of data from disk at one time and then get bytes from the block of data.

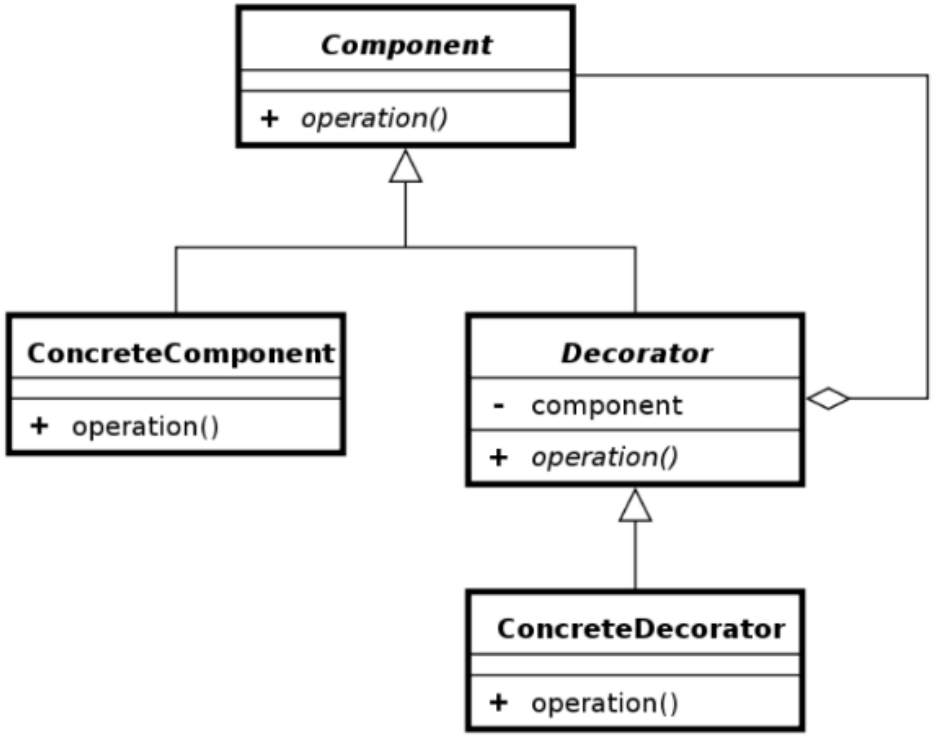
To achieve buffering, InputStream can be wrapped in BufferedInputStream.

```
InputStream input = new BufferedInputStream (new FileInputStream("c: \\ data\\input-file.txt") ) ;
```

CODEGYM

Decorator

A decorator is a structural pattern that allows you dynamically add new behaviors to objects by wrapping them in wrapper objects. Decorators provide flexible alternative to subclasses for extending functionality.



CODEGYM

Homework

MODULE 1. JAVA SYNTAX

Complete Level 25

CODEGYM

Answers to questions

MODULE 1. JAVA SYNTAX