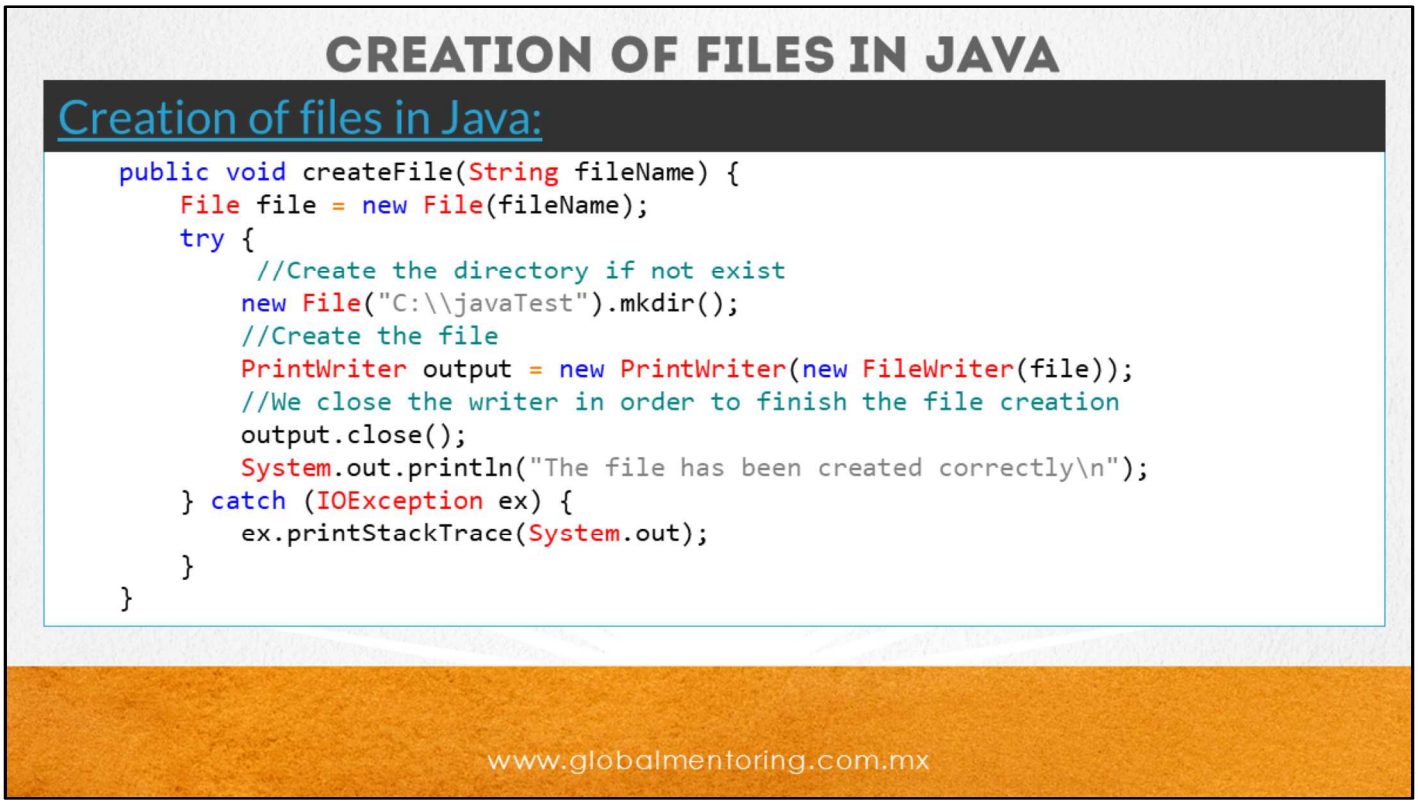
Hello, Ubaldo Acosta greets you. Welcome again. I hope you're ready to start with this lesson.

We are going to study the topic of file management in Java.

Are you ready? OK let's go!

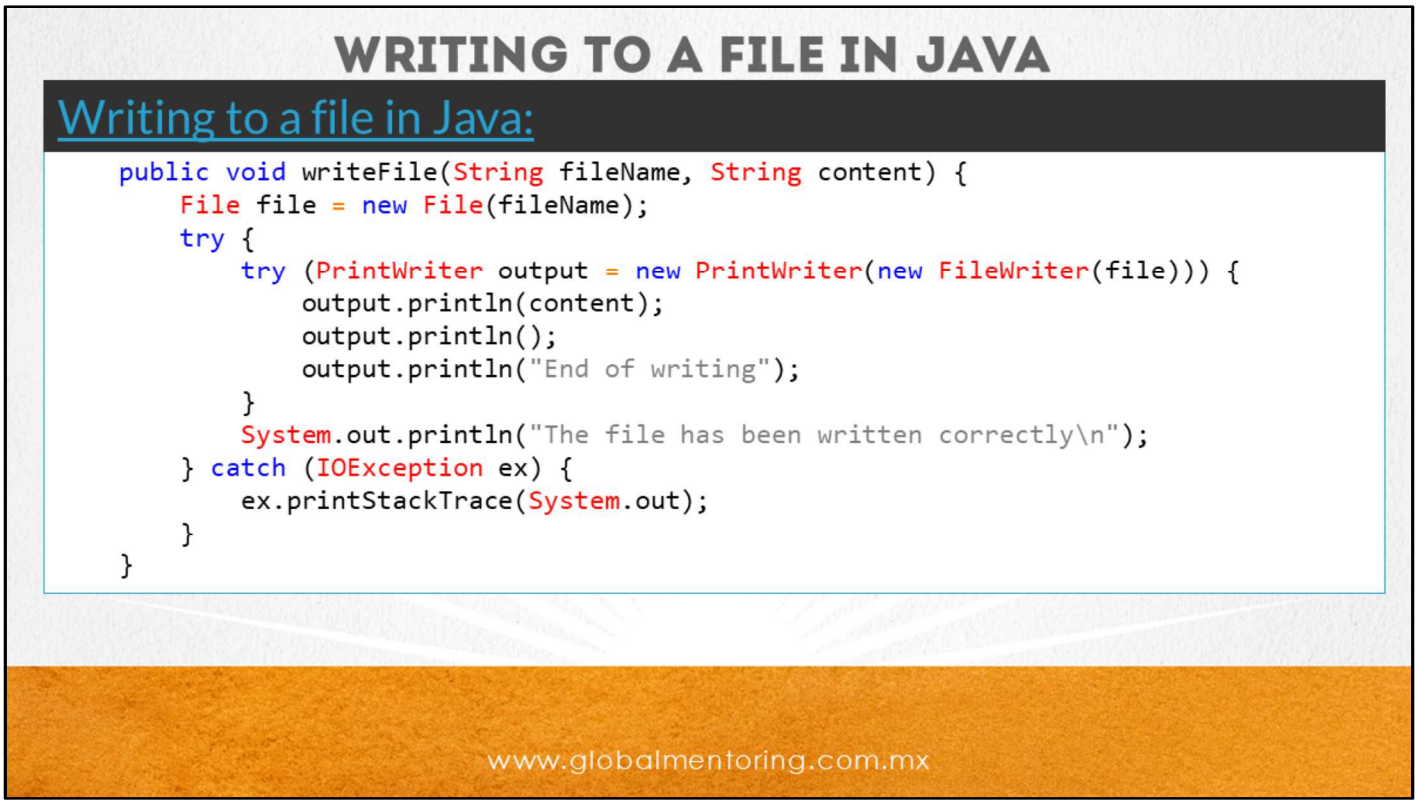


In this lesson we will study the management of files in Java. The File, FileWriter and PrintWriter classes are used to create a file. It is convenient to verify that we have permissions to write the file, for example, we can create a folder and assign writing permissions, in this way we will not have problems to create, read or modify the created file.

If the folder has not been created yet, you can use the mkdir() method of the File class in order to create one.

The FileWriter class is used to write information to a file, and as we can see in the code, we first create a file with the File class, however this does not physically create the file, for this we need the FileWriter class and finally the PrintWriter class that is the one that will create the file on the hard disk. We see that the test file is created inside a folder which must already exist so that it does not cause file creation errors.

Once the file is created, we can read and write about it, then we will see how to perform these processes.



Let's now study how to write information to a file. The FileWriter class is used to write characters to a file.

The PrintWriter class is used to print complete lines, with the print () and println () methods. The disadvantage of this method is that it loses the information that we have previously stored in the file, therefore it does not tell us to add new information, but write from scratch to the file.

Here, we are using the concept of try-with-resources when opening the PrintWriter object. In this case we are opening the file with the FileWriter class, after that the PrintWriter class, and automatically the handling of try-with-resources will close the resource once the code block of the try is finished, in this way we will make sure that no resources remain open due to code error or failure in the execution of the program.

Later we will see an example of how to carry out this process.

Let's see now how to read information from a file. The File, FileReader and BufferedReader classes are used to read a file

The code is very similar to reading information from the standard console.

The FileReader class is used to read characters.

The BufferedReader class is used to read complete lines.

So depending on what we need to read, it will be the kind of class we can use. Later we will see an example of reading information from a file.

Let's see now how to append information to a file already created.

The difference with the previous exercise is the call to the constructor of the FileWriter class, now we provide a true flag, which means that the information is going to be added to the existing one, and with this, what is already in the box is not lost. file, but it append the information provided.

Later we will see an exercise to attach information to a file.