

Practices for Lesson 14: Java File NIO2

Chapter 14

Practices for Lesson 14: Overview

Practice Overview

In these practices, explore various new features in Java 8 that relate to streams.

Practice 14-1: Working with Files

Overview

In this practice, read text files using new features in Java 8 and the `lines` method.

Assumptions

You have completed the lecture portion of this lesson and the previous practice. A text excerpt from the play Hamlet has been provided you as a test file in the root directory of the project. The contents of the files are as follows.

```
Enter Rosencrantz and Guildenstern.

Pol. Fare you well, my lord.
Ham. These tedious old fools!
Pol. You go to seek the Lord Hamlet. There he is.
Ros. [to Polonius] God save you, sir!

Exit [Polonius].

Guil. My honour'd lord!
Ros. My most dear lord!
Ham. My excellent good friends! How dost thou, Guildenstern? Ah,
    Rosencrantz! Good lads, how do ye both?
Ros. As the indifferent children of the earth.
Guil. Happy in that we are not over-happy.
    On Fortune's cap we are not the very button.
Ham. Nor the soles of her shoe?
Ros. Neither, my lord.
Ham. Then you live about her waist, or in the middle of her
    favours?
Guil. Faith, her privates we.
Ham. In the secret parts of Fortune? O! most true! she is a
    strumpet. What news?
Ros. None, my lord, but that the world's grown honest.
Ham. Then is doomsday near! But your news is not true. Let me
    question more in particular. What have you, my good friends,
    deserved at the hands of Fortune that she sends you to prison
    hither?
Guil. Prison, my lord?
Ham. Denmark's a prison.
Ros. Then is the world one.
Ham. A goodly one; in which there are many confines, wards, and
    dungeons, Denmark being one o' th' worst.
Ros. We think not so, my lord.
Ham. Why, then 'tis none to you; for there is nothing either good
    or bad but thinking makes it so. To me it is a prison.
Ros. Why, then your ambition makes it one. 'Tis too narrow for
your
mind.
Ham. O God, I could be bounded in a nutshell and count myself a
    king of infinite space, were it not that I have bad dreams.
Guil. Which dreams indeed are ambition; for the very substance of
    the ambitious is merely the shadow of a dream.
Ham. A dream itself is but a shadow.
```

Tasks

1. Open the `LambdaFiles14-01Prac` project.
 - Select `File > Open Project`.
 - Browse to `/home/oracle/labs/14-NIO.2/practices/practice1`.
 - Select `LambdaFiles14-01Prac` and click `Open Project`.
2. Edit the `P01BufferedReader` class to perform the steps that follow.
3. Using a `BufferedReader` and a stream, read in and print out the `hamlet.txt` file.
4. The output should look like the original text provided above.
5. Edit the `P02NioRead` class to perform the steps that follow.
6. Using the `Path`, `File`, and `Files` classes and a stream to read and print the contents of the `hamlet.txt` file.
7. The output should look like the original text provided above.
8. Edit the `P03NioReadAll` class to perform the steps that follow.
9. Using the NIO features and streams, read the contents of the `hamlet.txt` file into an `ArrayList`.
10. Filter and print out the lines for `Rosencrantz` for example: `String.contains("Ros.")`. The output should look similar to the following:

```
=== Rosencrantz ===
Ros. [to Polonius] God save you, sir!
Ros. My most dear lord!
Ros. As the indifferent children of the earth.
Ros. Neither, my lord.
Ros. None, my lord, but that the world's grown honest.
Ros. Then is the world one.
Ros. We think not so, my lord.
Ros. Why, then your ambition makes it one. 'Tis too narrow for
your
```

11. Filter and print out the lines for `Guilkenstern` (`"Guil."`). The output should look similar to the following:

```
=== Guilkenstern ===
Guil. My honour'd lord!
Guil. Happy in that we are not over-happy.
Guil. Faith, her privates we.
Guil. Prison, my lord?
Guil. Which dreams indeed are ambition; for the very substance of
```

12. Edit the `P04NioReadAll` class to perform the steps that follow.
13. Using the NIO features and streams, read the contents of the `hamlet.txt` file into an `ArrayList`.

15. Filter and print out the word "lord". Print a count of the number of times the word occurs. The output should look similar to the following:

```
=== Lord Count ===  
lord.  
lord!  
lord!  
lord.  
lord,  
lord?  
lord.  
Word count: 7
```

16. Filter and print out the word "prison". Print a count of the number of times the word occurs. The output should look similar to the following:

```
=== Prison Count ===  
prison  
prison.  
prison.  
Word count: 3
```

Practice 14-2: Working with Directories

Overview

In this practice, list directories and files using new features found in Java 8.

Assumptions

You have completed the lecture portion of this lesson and the previous practice.

Tasks

1. Open the `LambdaDirectory14-02Prac` project.
 - Select `File > Open Project`.
 - Browse to `/home/oracle/labs/14-NIO.2/practices/practice2`.
 - Select `LambdaDirectory14-02Prac` and click `Open Project`.
2. Edit the `DirList` class to perform the steps that follow.
3. Read all the files in the current directory using the `list` method.
4. Print the results. The output should look similar to the following:

```
=== Dir list ===
./build
./hamlet.txt
./nbproject
./src
./manifest.mf
./build.xml
```

5. Edit the `DirWalk` class to perform the steps that follow.
6. Use the `Files.walk` method to read the directory tree for the project.
7. Print the results. The output should look similar to the following:

```
=== Dir walk ===
.
./build
./build/classes
./build/classes/.netbeans_automatic_build
./build/classes/.netbeans_update_resources
./build/classes/com
./build/classes/com/example
./build/classes/com/example/lambda
./build/classes/com/example/lambda/DirFind.class
./build/classes/com/example/lambda/DirList.class
./build/classes/com/example/lambda/DirWalk.class
./build/classes/com/example/lambda/Main.class
./build.xml
./hamlet.txt
./manifest.mf
./nbproject
./nbproject/build-impl.xml
./nbproject/genfiles.properties
./nbproject/private
./nbproject/private/private.properties
```

```
./nbproject/private/private.xml
./nbproject/project.properties
./nbproject/project.xml
./src
./src/com
./src/com/example
./src/com/example/lambda
./src/com/example/lambda/DirFind.java
./src/com/example/lambda/DirList.java
./src/com/example/lambda/DirWalk.java
./src/com/example/lambda/Main.java
```

8. Next, walk the directory tree and filter the results so that only paths containing "build" are displayed.
9. The output should look similar to the following:

```
=== Dir build ===
./build
./build/classes
./build/classes/.netbeans_automatic_build
./build/classes/.netbeans_update_resources
./build/classes/com
./build/classes/com/example
./build/classes/com/example/lambda
./build/classes/com/example/lambda/DirFind.class
./build/classes/com/example/lambda/DirList.class
./build/classes/com/example/lambda/DirWalk.class
./build/classes/com/example/lambda/Main.class
./build.xml
./nbproject/build-impl.xml
```

10. Edit the DirFind class to perform the steps that follow.
11. Use the Files.find method to search the directory structure for entries that are directories.
12. Print the results. The output should look similar to the following:

```
=== Find all dirs ===
.
./build
./build/classes
./build/classes/com
./build/classes/com/example
./build/classes/com/example/lambda
./nbproject
./nbproject/private
./src
./src/com
./src/com/example
./src/com/example/lambda
```

