



GRAND CIRCUS

CODING • BOOTCAMPS

JAVA BOOTCAMP

TEXT AND BINARY FILES

TEXT AND BINARY FILES

TOPICS - WORKING WITH FILE I/O

TEXT AND BINARY FILES

A PACKAGE FOR WORKING WITH DIRECTORIES AND FILES

```
java.nio.*
```

```
java.io.*
```

IMPORTANT FILE I/O TYPES FILE IO CLASSES AND INTERFACES PATH PATHS FILE FILES

TEXT AND BINARY FILES

METHODS OF THE *PATH* INTERFACE

- `getFileName()`
- `getParent()`
- `getRoot()`
- `toAbsolutePath()`

TEXT AND BINARY FILES

METHODS OF THE *FILES* CLASS

- exists(Path)
- isReadable(Path)
- isWritable(Path)
- isDirectory(Path)
- size(Path)
- createFile(Path)
- createDirectory(Path)
- delete(Path)

TEXT AND BINARY FILES

CODE THAT CREATES A DIRECTORY IF IT DOESN'T ALREADY EXIST

```
String dirString = "c:/murach/java/files";  
Path dirPath = Paths.get(dirString);  
if (Files.notExists(dirPath)) {  
    Files.createDirectories(dirPath);  
}
```

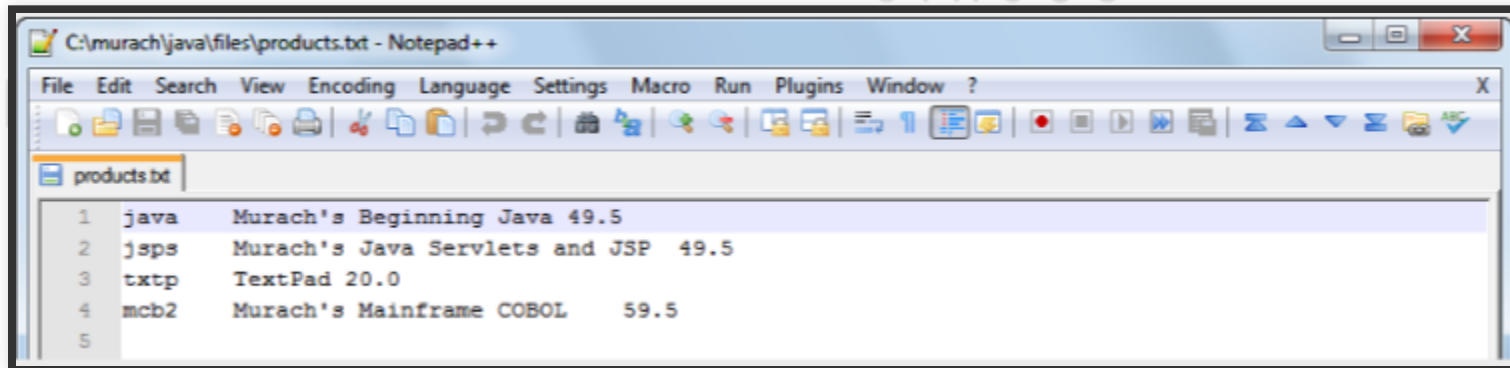
TEXT AND BINARY FILES

CODE THAT CREATES A FILE IF IT DOESN'T ALREADY EXIST

```
String fileString = "products.txt";  
Path filePath = Paths.get(dirString, fileString);  
if (Files.notExists(filePath)) {  
    Files.createFile(filePath);  
}
```

TEXT AND BINARY FILES

A TEXT FILE THAT'S OPENED BY A TEXT EDITOR



[illegible]

TEXT AND BINARY FILES

WRITE DATA TO THE FILE

```
File productsFile = new File ("products.txt");

try {
    PrintWriter out = new PrintWriter(new FileOutputStream(
        productsFile,true)); // append = true
    out.println("Hello World!");
    out.close();
}
catch (IOException e)
{
    System.out.println(e);
}
```

TEXT AND BINARY FILES

READ DATA FROM THE FILE

```
File productsFile = new File ("products.txt");  
try {  
    FileReader r = new FileReader(productsFile);  
    BufferedReader in = new BufferedReader(r);  
    ...  
}
```

TEXT AND BINARY FILES

READ DATA FROM THE FILE(CONTINUE)

```
String line = in.readLine();
while (line != null)
{
    System.out.println(line);
    line = in.readLine();
}
in.close();
} catch (IOException e) {
    System.out.println(e);
}
```

TEXT AND BINARY FILES

- IOException
- EOFException
- FileNotFoundException

RECAP

RECAP

WHAT YOU SHOULD KNOW AT THIS POINT

- Know File I/O Java classes
- Know the functions of the different File I/O classes
- Using File I/O classes to get information about files and directories
- Know file types and the differences between them
- How to write and read from files