

Mainstream Java

Reading Files with Java.nio

Information Hiding and Resources

- Why we have `edu.duke.*` and `FileResource`
 - Insulate programmers from complexity
 - Even without `edu.duke.*` this happens!
 - Empowering essentials: problem-solving!
- Duplicating `FileResource` and `URLResource`
 - Seven classes and three packages
 - Exceptions and different idioms

Reading Files Simply

```
public void runHello(){  
    FileResource hello = new FileResource("hello_unicode.txt");  
    for(String line : hello.lines()){  
        System.out.println(line);  
    }  
}
```

- Using edu.duke.FileResource
 - Hello Around the World!

Reading Files Simply

```
public void runHello(){  
    FileResource hello = new FileResource("hello_unicode.txt");  
    for(String line : hello.lines()){  
        System.out.println(line);  
    }  
}
```

- Using edu.duke.FileResource
 - Hello Around the World!
 - Create FileResource

Reading Files Simply

```
public void runHello(){  
    FileResource hello = new FileResource("hello_unicode.txt");  
    for(String line : hello.lines()){  
        System.out.println(line);  
    }  
}
```

- Using edu.duke.FileResource
 - Hello Around the World!
 - Create FileResource
 - Loop over each line of file

Reading Files with Raw Java

```
public void readAndPrint() throws IOException{
    Path p = Paths.get("hello_unicode.txt");
    BufferedReader reader = Files.newBufferedReader(p);
    while (true) {
        String line = reader.readLine();
        if (line == null) {
            break;
        }
        System.out.println(line);
    }
}
```

- Using seven classes and three packages

Reading Files with Raw Java

```
public void readAndPrint() throws IOException{  
    Path p = Paths.get("hello_unicode.txt");  
    BufferedReader reader = Files.newBufferedReader(p);  
    while (true) {  
        String line = reader.readLine();  
        if (line == null) {  
            break;  
        }  
        System.out.println(line);  
    }  
}
```

- Using seven classes and three packages
 - Create Path resource

Reading Files with Raw Java

```
public void readAndPrint() throws IOException{
    Path p = Paths.get("hello_unicode.txt");
    BufferedReader reader = Files.newBufferedReader(p);
    while (true) {
        String line = reader.readLine();
        if (line == null) {
            break;
        }
        System.out.println(line);
    }
}
```

- Using seven classes and three packages
 - Create Path resource
 - Create BufferedReader from Path using Files

Reading Files with Raw Java

```
public void readAndPrint() throws IOException{
    Path p = Paths.get("hello_unicode.txt");
    BufferedReader reader = Files.newBufferedReader(p);
    while (true) {
        String line = reader.readLine();
        if (line == null) {
            break;
        }
        System.out.println(line);
    }
}
```

- Using seven classes and three packages
 - Create Path resource
 - Create BufferedReader from Path using Files

Reading Files with Raw Java

```
public void readAndPrint() throws IOException{
    Path p = Paths.get("hello_unicode.txt");
    BufferedReader reader = Files.newBufferedReader(p);
    while (true) {
        String line = reader.readLine();
        if (line == null) {
            break;
        }
        System.out.println(line);
    }
}
```

- Using seven classes and three packages
 - Create Path resource
 - Create BufferedReader from Path using Files
 - Loop over each line of file

Java Package Details

- `java.nio.file` package
 - `Paths`, `Files`, `Path`
 - Note that `Files` and `Paths` use static methods to get or create resources
 - Many uses throw exceptions, deal with those
- `java.io` package
 - `BufferedReader` and `IOException`
 - Lots of details for all situations

URLs and java.net

```
URL source = new URL("http://.../hello_unicode.txt");
BufferedReader reader =
    new BufferedReader(
        new InputStreamReader(source.openStream()));
while (true) {
    String line = reader.readLine();
    // as before
```

- Reading from URLs similar to URLResource

URLs and java.net

```
URL source = new URL("http://.../hello_unicode.txt");
BufferedReader reader =
    new BufferedReader(
        new InputStreamReader(source.openStream()));
while (true) {
    String line = reader.readLine();
    // as before
```

- Reading from URLs similar to URLResource
 - Create a URL object

URLs and java.net

```
URL source = new URL("http://.../hello_unicode.txt");
BufferedReader reader =
    new BufferedReader(
        new InputStreamReader(source.openStream()));
while (true) {
    String line = reader.readLine();
    // as before
```

- Reading from URLs similar to URLResource
 - Create a URL object
 - Use the URL object and java.io classes

URLs and java.net

```
URL source = new URL("http://.../hello_unicode.txt");
BufferedReader reader =
    new BufferedReader(
        new InputStreamReader(source.openStream()));
while (true) {
    String line = reader.readLine();
    // as before
```

- Reading from URLs similar to URLResource
 - Create a URL object
 - Use the URL object and java.io classes
 - Hierarchy of Reader classes

URLs and java.net

```
URL source = new URL("http://.../hello_unicode.txt");
BufferedReader reader =
    new BufferedReader(
        new InputStreamReader(source.openStream()));
while (true) {
    String line = reader.readLine();
    // as before
```

- Reading from URLs similar to URLResource
 - Create a URL object
 - Use the URL object and java.io classes
 - Hierarchy of Reader classes

URLs and java.net

```
URL source = new URL("http://.../hello_unicode.txt");
BufferedReader reader =
    new BufferedReader(
        new InputStreamReader(source.openStream()));
while (true) {
    String line = reader.readLine();
    // as before
```

- Reading from URLs similar to URLResource
 - Create a URL object
 - Use the URL object and java.io classes
 - Hierarchy of Reader classes
 - Read all lines