**日報**

2023年08月28日

氏名：小松大二

|  |
| --- |
| ■目標  ・Problem15の問題を進める |
| ■作業内容  ・Problem15の問題を進める  ＊佐藤さんから頂いたpdfファイルを読んで、各クラスなどの役割を理解する  ＊問題2「ファイル内の文章について、「。」の後は改行して出力する」  →　完了 |
| * 習得したこと   ・pdfファイルを見て処理の流れを把握することができる  ・NestSearchではdata\in以下のファイルが全て探索され、SimpleRuleで、探索されたファイルの中から、ファイル名に「Problem15」が含まれるものを選び、SimpleManagerで保持されることが理解できる  ・SimpleManagerで保持されたファイルをgetRebuildRuleメソッドで「data\in → data\out」に変更し、読み込むパスと書き込むパスの組み合わせを決め、最後にdata\inからdata\outファイルを出力していることが理解できる  ・BufferedReaderとBufferedWriterでファイルの読み取りと書き込みができることを理解し、その二つとreplaceAllメソッドを使って、replaceAll("。", "。\n");と書くことで、ファイル内の文章の「。」の後に改行を挿入したものを出力することができる  ・メソッド名();で、メソッドを呼びだすことができることを改めて理解する |
| * 感想   今日もProblem15の課題を進めました。  佐藤さんから各クラスなどの役割と処理イメージをまとめたpdfファイルを頂いたので、それをじっくり読んで各ファイルについて理解することから、今日は取り掛かりました。pdfファイルを読んで、難しい内容だったため、処理の流れを完全に理解したとは言い切れませんが、なんとなくのイメージで把握することができました。それに合わせて各クラスの役割についても、pdfファイルを読む前よりも明瞭にイメージすることができ、NestSearchではdata\in以下のファイルが全て探索され、SimpleRuleで、探索されたファイルの中から、ファイル名に「Problem15」が含まれるものを選び、SimpleManagerで保持される、そして、SimpleManagerで保持されたファイルをgetRebuildRuleメソッドで「data\in → data\out」に変更し、読み込むパスと書き込むパスの組み合わせを決め、最後にdata\inからファイルを読み込んでdata\outに出力しているという流れも理解することができました。  処理の流れと各クラスの役割について、先週よりも理解できたので、このまま先週解けなかった問題2に取り組みました。SampleManagerのgetRebuildRuleにて編成処理を行っていることが理解できたので、この部分に処理を書き足せば「。」の後に改行して出力することができるのではないかと考えました。処理を書いていくために、まずはSampleManagerのgetRebuildRuleの部分を読んでみると、先週自分が使ったreplaceAllメソッドが使われていることが確認できました。ここでそのメソッドが使われているのならば、自分が先週書いたものをここに書けば望んだ結果が得られると思い、実際に書いてみました。  Data\inをdata\outに変更する処理の書き方を真似しながら書いてみたのですが、エラーが出てしまい、問題2を実装することができませんでした。なぜエラーが出てしまったのか悩みましたが、恐らく、ここではあくまでパスの変更を行っているだけで、ファイルの中身の変更を行っているわけではないのでエラーが出てしまったのかなと考えました。やはり問題2を解決するためには、先週書いたような、ファイルの読み取り、内容の変更、ファイルへの書き込みを行うためのメソッドを作成する必要があるのだと改めて思いました。  この考えのもと、先週自分が書いたメソッドをもう一度見直し、改めて書いてみて、ファイルの読み取り（BufferedReader）、内容の変更（replaceAllメソッド）、ファイルへの書き込み（writeメソッド）の必要なメソッドがあることを確認して、プログラムを実行してみたのですが、それでも動きませんでした。プログラムを実行した際にエラーが起きているわけでもなく、何が原因なのか分からずとても苦戦しました。エラーは起きていないけど、何かを書いていない、もしくは誤字などがあるせいでメソッドが正常に機能していないのではと考え、改めてプログラムを読み直した結果、まずそもそもmainメソッド内で、改行の指示を出したnewLineメソッドを呼び出せていませんでした。mainメソッド内にnewLine();と書き、再度プログラムを実行したところ、無事に「。」の後に改行を挿入することができ、問題2を解決することができました。  あまりに初歩的なミスで、自分で気づいたとき呆れて笑ってしまいました。このミスが起きてしまった原因として、今までmainメソッド内で別のメソッドを呼び出すということを、自分の手でやってこなかったため、メソッドを呼びだす文は最初からあるものだと考えてしまっていたことにあります。今回のことで、改めて「メソッド名();」でメソッドを呼びだすことができることを覚えたので、次からは忘れないようにしていきたいです。 |