プログラミング（１／３） 実験指導書
ファイル・ディレクトリ処理

1 目的

コンピュータのファイルシステムには，ファイル（file）やディレクトリ（directory, 別名：フォルダ（folder））というものがある．ファイルとはハードディスク等に保存されたデータを一まとまりにしたものである．ファイルを多数保存した場合，目的のファイルを探すのが煩雑になる．そのため，机の引き出しの中を区切るようにしてファイルを整理すると，管理が楽になる．ハードディスク等にファイル入れておくために区切った小部屋をディレクトリという．

また，ディレクトリの中にはファイルが保存されたり，別のディレクトリ（サブディレクトリ）が作成されたりする．そしてまた，そのサブディレクトリの中に他のファイルやディレクトリが保存／作成されたりする．このようなファイルシステムを階層型ファイルシステムと言う．

本実験はプログラムからファイルシステムを処理する基本を習得することを目的とする．同時にコマンドプロンプト（command prompt）との関係に関して学ぶ．ただし，言語は Java を使用し，標準ライブラリを用いる．

2 基本的事項

2.1 コマンドプロンプト

CUI（character user interface）と呼ばれる入力インターフェースで，コマンドで命令を入力して処理を実行する Windows のアプリケーションである．Java のプログラムを実行する場合，コマンドプロンプト等から実行するが，カレントディレクトリ（current directory），相対パス（relative path），絶対パス（absolute path）を理解する必要がある．また，コマンドプロンプトから実行するアプリケーションはプロセスとして一まとまりとして扱われる．プロセス（process）は，英語で「処理」や「過程」といった意味を持つ単語だが，コンピュータの世界ではプログラム実行中のインスタンス（実体）の意味で使われる．

2.2 カレントディレクトリ

あるプロセスが現在関連付けられている（現在の位置である）ディレクトリのことである．

2.3 相対パス

現在の位置としているカレントディレクトリを起点に，指し示したいディレクトリの相対位置を記述する方法で，途中にあるディレクトリを区切り記号で繋い

で並べる．区切り記号は Windows では「¥」（日本では円記号，日本以外ではバックスラッシュ），UNIX 系 OS などでは「/」（スラッシュ）を用いる．

2.4 絶対パス

ファイルやフォルダの所在を示すパスの表記法の一つで，階層型ファイルシステムの頂点（最上位階層）から目的のファイルやディレクトリまでの道筋を省略なくすべて記述する方式．頂点は UNIX 系 OS ではルートディレクトリ（「/」であらわされる）を，Windows ではドライブ名（「C:¥」など）である．

2.5 コマンドプロンプトの使用方法

スタートメニューの検索で「cmd」と入力する．検索されたコマンドプロンプトをクリックし，実行する．標準では"C:¥Users¥「本人のユーザー名」>"がプロンプトになり，カレントディレクトリを含んだ表記となっている．プロンプトは CUI アプリケーションの入力を促すメッセージ部である．Windows では">"が使用される．

このプロンプトからは内部コマンドと外部コマンドを実行することができる．外部コマンドとは，実行できる形態のアプリケーションのファイルのことである．ファイルシステムの何処かに実態のファイルがある．内部コマンドとはコマンドプロンプトにある標準のコマンドのことである．いつでも実行可能である．java, javac は外部コマンドである．そのためインストールが必要となる．外部コマンドのインストールはファイルシステムへのファイルの保存である．また便利に使用するために PATH （パス）を通す作業が必要となる．コマンドプロンプトはカレントディレクトリ（現在の位置）と関連付けられており，外部コマンドを探す順番が決まっている．先ず，カレントディレクトリから外部コマンドを探す．次に，環境変数 PATH のセミコロンで分けられた文字列から順にディレクトリを探す．PATH を通す作業とは，外部コマンドを簡単に実行する際に必要となる．絶対パスで直接外部コマンドを実行する場合には必要ないことになる．

以下に，内部コマンドの例を挙げる．

2.5.1 cd

cd とは chage directory の略でカレントディレクトリを移動する．ディレクトリは絶対パス／相対パスどちらでも指定できる．

2.5.2 dir

dir とは directory の略でファイルとサブディレクトリの一覧を表示する．

2.5.3 md（mkdir）

md とは make directory の略でディレクトリを作成する.

2.5.4 cls

cls とは clear screen の略でコマンドプロンプトの内容をすべて消去（クリア）する.

2.5.5 type

テキストファイルの内容を出力する.

2.5.6 exit

コマンドプロンプトを終了させる.

2.5.7 .

「.」はカレントディレクトリを意味する．半角ドットである.

2.5.8 ..

「..」は親（上位）ディレクトリを意味する．半角ドットを２個連ねている.

2.5.9 テキストファイルの作成（応用）

「type　nul　>　text.txt」は type コマンドの応用でテキストファイルを作成する．これはリダイレクトを使用し標準出力をファイルへ変更している．リダイレクトや標準出力などの説明は割愛する.

3　実験で使用するプログラミング文法（Java）

3.1　拡張 for 制御文

拡張 for 制御文とは，コレクション（ある型の集合）や配列から要素を取り出しながら何らかの処理をする繰り返しである．以下の様な記述となる．普通の for 制御文と違いカウンターが必要ない.

```
for　(型 変数名: コレクション) {
        命令文; …
}
```

3.2 列挙型

enum と言うキーワードが導入され, 列挙型をオブジェクトとして利用できる. 列挙型とは static（静的）な変数をまとめて型付けをしたものである．以下の様な記述となる.

```
public enum 型 {
        変数名１, 変数名２,…
}
```

型.変数名１で参照することができる.

3.3 可変長引数

引数の個数が可変なメソッドを定義できる.

```
public void method1 (型... 変数名) {
        …
}
```

メソットの定義において引数の型の後に半角ドットを３個連ねて記述することで，そのメソッドを可変長引数とすることができる.

4 実験で使用する標準ライブラリ

4.1 Path

ファイルシステムのファイルを特定するために使用可能なオブジェクトである．通常，それはシステムに依存する絶対パス／相対パスである.

4.2 StandardCopyOption（列挙型）

標準コピー・オプションの定義である.

4.3 Files

このクラスは，ファイル，ディレクトリ，またはその他の種類のファイルを操作する static メソッドだけである.

4.4 BasicFileAttributes

ファイル・システム内のファイルに関連付けられた基本属性です.

4.5 LocalDateTime

タイムゾーンのない日付/時間である.

4.6 ZoneId

タイムゾーンIDを表現する.

4.7 Class

Classクラスのオブジェクトは，実行中のJavaアプリケーションのクラスを表したものである.

5 実験方法

5.1 ファイルコピー

```
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;

public class E1_1 {
	public static void main(String[] args) {
		String source = args[0];
		String target = args[1];
		Path sourceFile = Paths.get(source);
		Path targetFile = Paths.get(target);
		try {
			Path copyFile = copy(sourceFile, targetFile);
			System.out.println("コピーできました");
		} catch (IOException e) {
			e.printStackTrace();
		}
	}
	public static Path copy (Path sourceFile, Path targetFile) throws IOException {
		Path path = Files.copy(sourceFile, targetFile);
		return targetFile;
	}
}
```

5.2　ディレクトリ作成

実験にて配布する別紙を参照

5.3　ディレクトリ・ファイル表示

実験にて配布する別紙を参照

5.4　属性

実験にて配布する別紙を参照

6　実験結果

※本人の結果を記述する

7　検討事項

7.1　ファイル数　（最低基準）

実験にて配布する別紙を参照

7.2　ファイル容量　（標準基準）

実験にて配布する別紙を参照

7.3　ディレクトリ・ファイル階層表示（A 基準）

実験にて配布する別紙を参照

7.4　ディレクトリ間コピー　（S 基準）

実験にて配布する別紙を参照

8　参考文献

1）柴田 芳樹. Java 2 Standard Edition 5.0 Tiger: 拡張された言語仕様について. ピアソンエデュケーション, 2005.

2）オラクル. Java(tm) Platform, Standard Edition 8 API 仕様. https://docs.oracle.com/javase/jp/8/docs/api/, (参照 2019-3-26).

3）柴田 芳樹. Java プログラマーなら習得しておきたい Java SE 8 実践プログラミング. 株式会社インプレス, 2014.

9　実験報告書に関して

この実験指導書の各項（本項を除く）を本人の理解を捕捉し，本人の言葉に直して提出すること．ただし，第５章は，本人のプログラムを含む．第６章は第５章の実行結果を記述する．第７章はプログラムと実行結果を記述すること．また，本人が参考にした書籍や Web ページは第８章に記述する．