

● J2Kad16D 「ファイル出力のきほん (FileWriter)」

(実践編 P.171 「File クラス」、P.153 「ファイルへの出力」)

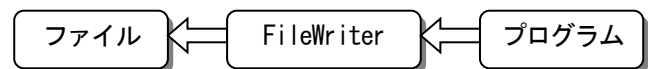
FileWriter を使ってファイル「test.txt」へ文字列「ECCCOMP」を出力する処理を作成せよ。なお、「test.txt」が存在する場合は削除してから（手作業で OK）実行すること。

ファイルの新規作成

```
Filefile = new File(ファイル名);
if (file.createNewFile()) {
    // 新規作成できたときの処理
}
```

ファイルへの出力

```
FileWriter fw = new FileWriter(file);
fw.write(出力する文字列);
fw.close();
```



課題完成時の画面

```
test.txt を作成しました！
test.txt に ECCCOMP を出力しました！
```

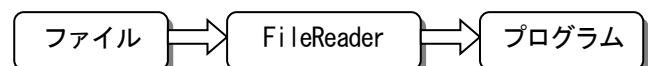
実際に文字列が出力されているかどうかは test.txt を開いて確認すること（IntelliJ のプロジェクトウィンドウから開くことも可能）。

● J2Kad16C 「ファイル入力のきほん (FileReader)」 (実践編 P.160 「ファイルからの入力」)

FileReader を使って「test.txt」へ出力した文字列を入力して表示する処理を作成せよ。

ファイルからの入力

```
FileReader fr = new FileReader(file);
int data;
while((data = fr.read()) != -1) System.out.println(data); // 読み込んだ値 (data) をそのまま表示
fr.close();
```



課題完成時の画面

```
test.txt から文字列を読み出します！
Eを読み出しました！
Cを読み出しました！
Cを読み出しました！
Cを読み出しました！
Oを読み出しました！
Mを読み出しました！
Pを読み出しました！
```

入力は1文字ずつ行われる。
なお、出力したデータをそのまま表示すると文字ではなく数値（文字コード）が表示される。

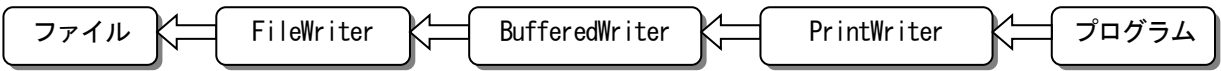
● J2Kad16B 「ストリームの連結 (BufferedWriter、BufferedReader)」

(実践編 P.156 「バッファ」、P.158 「ストリームの連結」)

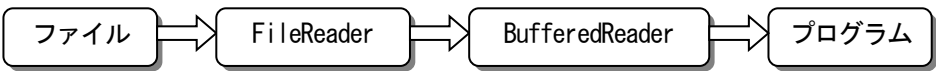
課題フォルダに Monster クラスが準備されている。以下の仕様でファイル入出力処理を作成せよ。なお FileWriter には BufferedWriter と PrintWriter を連結し、FileReader には BufferedReader を連結すること。

Monster
name
Monster ()
getName () : String
toString () : String

ファイル出力



ファイル入力



main メソッドの仕様

- ① ファイル「test.txt」がすでに存在しているときは削除する。
- ② ファイル「test.txt」を新規作成し、以下の処理を実行する。
- ③ Monster クラスを 5 体生成し、各モンスターの名前をファイルへ書き込む。なお、名前の後は改行すること。
- ④ ファイルから文字列を読み出し表示する (モンスターの名前が 5 行分表示される)。

課題完成時の画面

test.txt を削除しました！
test.txt を作成しました！
モンスター一覧を作成します！
トランセルを追加した！
ムックルを追加した！
ベトベトンを追加した！
モクローを追加した！
イワンコを追加した！

モンスター一覧を表示します！
トランセルを読み出した！
ムックルを読み出した！
ベトベトンを読み出した！
モクローを読み出した！
イワンコを読み出した！

扱うファイルは J2Kad16C と同じ「test.txt」。
J2Kad16B 完成後に再度 J2Kad16C を実行してファイルの内容を表示してみると FileReader のみの読み込みと BufferedReader を使った読み込みの違いがわかる。

● J2Kad16A 「ドットパターンの表示」(フォルダの操作は実践編 P.173 「フォルダの操作」)

課題フォルダの中にある data フォルダに A~Z および 0~9 の英数字のドットパターンが入ったファイルが準備されている (8×8 ドット、ファイル名は A.txt、B.txt、...)。data フォルダ内にあるファイル一覧を表示し、指定した英数字 (A~Z、0~9) のドットパターンを画面に表示する処理を作成せよ。

課題完成時の画面

0. txt1. txt2. txt3. txt4. txt5. txt6. txt7. txt8. txt9. txt

A. txtB. txtC. txtD. txtE. txtF. txtG. txtH. txtI. txtJ. txt

K. txtL. txtM. txtN. txtO. txtP. txtQ. txtR. txtS. txtT. txt

U. txtV. txtW. txtX. txtY. txtZ. txt

どのドットパターンを見ますか？(拡張子不要、-1:終了) >0

—***—

—*—*

—*—*

—*—*

—*—*

—*—*

—***—

—

0. txt1. txt2. txt3. txt4. txt5. txt6. txt7. txt8. txt9. txt

A. txtB. txtC. txtD. txtE. txtF. txtG. txtH. txtI. txtJ. txt

K. txtL. txtM. txtN. txtO. txtP. txtQ. txtR. txtS. txtT. txt

U. txtV. txtW. txtX. txtY. txtZ. txt

どのドットパターンを見ますか？(拡張子不要、-1:終了) >a

—***—

—*—*

—*—*

—*—*

—*****—

—*—*

—*—*

—

0. txt1. txt2. txt3. txt4. txt5. txt6. txt7. txt8. txt9. txt

A. txtB. txtC. txtD. txtE. txtF. txtG. txtH. txtI. txtJ. txt

K. txtL. txtM. txtN. txtO. txtP. txtQ. txtR. txtS. txtT. txt

U. txtV. txtW. txtX. txtY. txtZ. txt

どのドットパターンを見ますか？(拡張子不要、-1:終了) >ecc

そのデータは存在しません！

0. txt1. txt2. txt3. txt4. txt5. txt6. txt7. txt8. txt9. txt

A. txtB. txtC. txtD. txtE. txtF. txtG. txtH. txtI. txtJ. txt

K. txtL. txtM. txtN. txtO. txtP. txtQ. txtR. txtS. txtT. txt

U. txtV. txtW. txtX. txtY. txtZ. txt

どのドットパターンを見ますか？(拡張子不要、-1:終了) >-1

ファイル名の間にはタブを入れている。

ファイル一覧は 10 ファイル単位で改行。

アルファベットは大文字でも小文字でも OK

ファイルが存在しないときは「そのデータは存在しません！」と表示

-1 で終了

● J2Kad16X1 「文字列描画システム③ (Flyweight パターン)」 ※J2Kad16S2 をコピーして作成

J2Kad16S2 では文字列の中に同じ文字があっても何度も読み込みを行うようになっている (例えば「ECCCOMP」では「C」を3回読み込んでいる)。Flyweight パターン (←検索) を使って同じ文字は読み込まないようにせよ。

ヒント :

- Before : BigString で BigChar を生成
- After : BigChar をプールする BigCharFactory を作る。BigString は BigCharFactory から BigChar の参照をもらう。

BigCharManager クラスの仕様 (ファイル「BigCharManager.java に作成」)

メンバ	仕様
private HashMap<String, BigChar> pool	BigChar をプールする連想配列 (初期化しておくこと)。
public BigChar createBigChar(char charname)	charname (大文字の文字列に変換) が pool に登録されているときは該当する BigChar の参照を返す。まだ pool に登録されていないときは該当する BigChar を生成し pool に登録、その BigChar の参照を返す。

課題完成時の画面

文字列 (英数字) を入力してください>ECCCOMP
E を読み込みました！
C を読み込みました！
O を読み込みました！
M を読み込みました！
P を読み込みました！

「ECCCOMP」に「C」は3回出てくるが、読み込みは1回のみ。

● J2Kad16X2 「文字列描画システム④ (Singleton パターン)」 ※J2Kad16X1 をコピーして作成

J2Kad16X2 では「ECCCOMP」(1回目) と入力したあとに「ABCDE」と入力すると「A」「B」「D」だけでなく再び「C」「E」が読み込まれてしまう。これまでに読み込んだ文字がある場合も再読み込みしないように修正せよ。

ヒント : BigCharManager を Singleton (←検索) にする。

J2Kad16X1 の画面 (ECCCOMP のあとに ABCDE)

文字列 (英数字) を入力してください>ECCCOMP
:
文字列 (英数字) を入力してください>ABCDE
A を読み込みました！
B を読み込みました！
C を読み込みました！
D を読み込みました！
E を読み込みました！
:

「A」「B」「D」だけでなく「C」「E」も読み込む。

課題完成時の画面 (ECCCOMP のあとに ABCDE)

文字列 (英数字) を入力してください>ECCCOMP
:
文字列 (英数字) を入力してください>ABCDE
A を読み込みました！
B を読み込みました！
D を読み込みました！
:
:

「C」「E」は再読み込みしない。