

<https://www.kde.cs.tut.ac.jp/~aono/data/20news/> 以下に英語のメールファイルを置いている。(すべてのメールファイルの名前は数字です)。事前準備として上記のファイルのうち、自分の学籍番号の末尾の下1ケタと一致するファイルを1つ選択し、そのファイルを自分の作業フォルダにコピーし、以下の項目を満たすプログラムを作成せよ。

- (1) 処理するファイルをプログラムから読み込み、自分の名前と学籍番号、ならびにファイル名を書き出した後、ファイル内の**英単語**（定義は「コメントとヒント」内にあるが、すべての大文字アルファベットは全部小文字に変換するものとする）の出現数をカウントし、ファイルを最後（EOF）まで読み終わったら、出現した総単語数と、すべての英単語とその出現頻度を出力するプログラムを作成せよ。（ソーティングは必須ではない。）
- (2) プログラムの実行では、必ず、引数にメールファイル名を与えること（プログラムにファイル名を埋め込まない）とする。なお、プログラムは `Kadail.java` とすること。

```
$ java Kadail [ファイル名]
```

ファイル名がない場合や、open 出来ない場合は警告を出して終了すること。

提出は Java プログラム(Kadail.java)と実行結果(Kadail.txt)、ならびに選んだメールアドレスを ZIP(ファイル名=kadail.zip, 学籍番号はファイル自体にはつけないで結構です)でまとめて Moodle LMS から提出すること。締め切りは 6 月 29 日（火）の夜までとする。

コメントとヒント：

第一回なので、大事なところは、赤字にしています。

この課題での英単語とは、アルファベット【小文字に変換しているので小文字のみ】の文字列を意味します。たとえば、**He is 20 years old.**という元の文章があるとしたら、

he, is, years, old の 4 つがこの文章での英単語になります。また、**It's a well-known fact.**という文章であれば、

it, s, a, well, known, fact の 6 つがこの文章での英単語となります。

ファイルの読み込みは、第一回資料の `java.io.BufferedReader` クラスの説明で示した `FileReader` と `BufferedReader` の組み合わせが簡単だと思います。

アルファベット文字列の取得は、たとえば、`BufferedReader` の `readLine()`関数で行単位の文字列を取得（`String line = br.readLine();`）した後、各行の文字列の長さをみながら、`char c = line.charAt(i)`で文字列の `i` 番目の文字（`i=0,1,...line.length()-1`）を取得し、それが英単語かどうかの判定をしながら文字列を連結していけば得られると思います。その際、たとえば、読み込んだ文字（小文字変換する前と仮定）を変数 `c` で宣言したとすると、

```
if (('A' <= c && c <= 'Z') || ('a' <= c && c <= 'z'))
```

のような if 文でとしてチェックできます。得られた文字から文字列の生成では、+演算で連

結するといいいでしょう。その際 `String token=""`;で定義した文字列に、ループで `token = token + c;`でどんどん連結すれば、ここでの英単語文字列が得られます。大文字⇒小文字への変換も上記のような if 文でできます。

こうして得られた英単語と、その頻度を保持するデータ構造が必要となります。たとえば総単語数が 1000 を超えないと仮定してサイズ 1000 の文字列と頻度保持の配列を使って結構です。別な方法としては、英単語と頻度を保持するクラスを作成し `ArrayList`, `HashMap`, `TreeMap` などのシステムが提供するクラスのオブジェクトを使っても結構です。

例を示します。入力ファイル=GoogleNews.txt (以下の文章)

Alphabet will be run by Google co-founder Larry Page and each of its businesses will have its own chief executive.

```
$ javau Kadai1 GoogleNews.txt
```

```
*****
作成者：青野雅樹：01162069
日付：2021年6月8日 9:36:40 JST
入力ファイル名：GoogleNews.txt
*****
```

```
alphabet 1
google 1
larry 1
page 1
and 1
be 1
businesses 1
by 1
chief 1
co 1
each 1
executive 1
founder 1
have 1
its 2
of 1
own 1
run 1
will 2
```

総単語数は19です

なお、課題の出力をファイル出力するには、ターミナル窓で

```
$ javau Kadai1 9161 > Kadai1.txt
```

のように、> ファイル名で出力させるか、それができない場合は、上述のような出力結果をコピー&ペーストして、**Kadai1.txt**という名前のファイルに書き出す等で適宜、対応してください。

```

/*
課題1サポート用プログラム （文字単位の処理は、例えばcharAt(i)関数も利用ください）
..英単語を抽出する部分は含まれません。また、クラス名（ファイル名）は課題と異なります。
FileReadline.java（ファイル名を与え行単位に読み出し、プリントするクラス）
*/
import java.util.*;
import java.io.*;
import java.text.Format; /* フォーマットのため */
import java.text.DateFormat; /* 日付のフォーマット用 */

public class FileReadline {
    public String[] myName = { /* プリントヘッダー */
        "*****",
        "作成者：青野雅樹：01162069",
        "日付：",
        "入力ファイル名：",
        "*****"
    };
};

public void myPrint(String name){ /* mainの最初に呼ばれる */
    Date now = new Date(); /* Dateクラスのオブジェクト生成 */
    Format fmt= /* 日付を日本語ロカールに従ってフォーマット */
        DateFormat.getDateInstance(DateFormat.FULL, DateFormat.LONG);
    for (int i = 0 ; i < myName.length ; i++){
        System.out.print(myName[i]);
        switch (i) {
            case 2: System.out.print(fmt.format(now)); break; /* 日付 */
            case 3: System.out.print(name); break; /* データファイル名 */
        }
        System.out.print('\n');
    }
}

public static void main(String[] args){
    String fileName = args[0]; /* 第一引数 */
    if (args.length != 1){
        System.out.println("java(u) FileReadline [ファイル名]");
        System.exit(0);
    }
    FileReadline fr = new FileReadline(); /* 行単位の読み込み */
    try {
        fr.myPrint(args[0]); /* ヘッダーをプリント */
        /* BufferedReaderクラスのオブジェクト生成 */
        BufferedReader br = new BufferedReader(new FileReader(args[0]));
        String line; /* 行単位の読み込んだ文字列を保持 */
        while ((line = br.readLine()) != null){ /* 行単位で処理 */
            System.out.println(line); /* そのままプリント */
        }
    }
    catch(Exception e){
        e.printStackTrace(); /* エラー処理：トレース */
    }
}
}

```