

提出日 令和2年07月09日

アルゴリズムとデータ構造 第7回課題レポート

学籍番号 (A19117)

氏名 (永尾優磨)

【課題】

1. 「連想配列」についてまとめよ。

連想配列とは、添え字に好きな名前を付けることが出来る配列のこと  
一般的に考えられている配列は、最初が0、次に1、2と続いて値が入っているのに対して、  
連想配列だと、値を好きな数字や文字に変えることが出来るというもの。例えば、配列の1  
つめに、「和歌山」、次の配列には、「愛媛」という形に値を入れていくというもの。

2. 課題1を行え。

2.1. 以下のデータをHashMapクラスを使って記憶してみよ。

- 機械→M
- 電気電子→E
- 電子制御→D
- 制御情報→S
- 物質→C

2.2. 記憶したデータを表示させてみよ。

3. 課題2を行え。

3.1. HashMapを用いて与えられた単語データから単語の出現数を数え上げるプログラムを作  
れ。

- 単語データはこの授業の受講者全員にTeamsのコンテンツからダウンロードして、自分  
のフォルダに保存せよ。

3.2. さらに、結果として単語とその出現数を画面に表示せよ。

【作成したプログラム】

```
import java.util.*;

public class HashMapApp {
    public static void main(String[] args) {
        HashMap<String, String> map = new HashMap<String, String>();

        String[] gakkaj = { "機械", "電気電子", "電子制御", "制御情報", "物質" };

        String[] gakkae = { "M", "E", "D", "S", "C" };

        for (int i = 0; i < gakkae.length; i++) {
            map.put(gakkaj[i], gakkae[i]);
        }
    }
}
```

```

    Iterator it = map.keySet().iterator();
    while (it.hasNext()) {
        Object obj = it.next();
        System.out.println(obj + " = " + map.get(obj));
    }
}
}

```

プログラム1 HashMapAPP

```

import java.util.*;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;

public class WordCount {
    public static void main(final String[] args) throws IOException {
        final String content = Files.readString(Paths.get("第07回連想配列/alice.txt"));
        final String[] words = content.split(" ", 0);

        final HashMap<String, Integer> map = new HashMap<String, Integer>();

        for (int i = 0; i < words.length; i++) {
            if (map.containsKey(words[i])) {
                map.put(words[i], map.get(words[i]) + 1);
            } else {
                map.put(words[i], 1);
            }
        }

        System.out.println(map);
    }
}

```

プログラム2 WordCount

## 【プログラムの解説】

HashMapは、中に入るデータの数が変更してもすぐに対応できるようにするために、for文の回数を配列の長さにした。

WordCountは、aliceをテキストファイルに変更し、データを自動的に読み込めるようにした。読み込んだ後、単語数を数え上げ、単語数を出力するという形にした。

## 【結果】

```
lJava_575429d3\bin 第07回連想配列.HashMapApp  
電子制御 = D  
制御情報 = S  
電気電子 = E  
機械 = M  
物質 = C
```

図1 HashMapAppの出力結果

{half=1, don't=1, your=1, I've=2, hurt=1, pretend=1, without=1, air=1, person!'

Soon=1, would=5, poor=4, people!=1, rabbit=1, legs=1, leaves=1, sight=1, near=1, because=1, people.=1, ten=1, siste  
r=1, is=1, WATCH=1, I=23, talking=1, The=1, same=1, telescope.'

And=1, after=3, close=2, book?shelves=1, hand=3, it=11, a=51, wouldn't=1, it.=2, flavour=1, time=1, doorway=1, ne  
ither=1, wise=1, late!=1, cupboards=2, leave=1, further=1, corner=2, making=1, knowledge=1, off=1, Do=2, happe  
1, Alice=1, you're=1, curiosity=1, cried.

'Come=1, ventured=1, other=1, 'to=1, with=11, earth!=1, White=2, there=4, well=3, happens=1, to=1, tried=5, disapp  
which=1, larger=1, pink=1, other=1, not'=1, against=1, matter=1, turkey=1, shutting=2, first'=1, saw.=1, top=2, ca  
.

However=1, almost=1, large=1, upon=4, ME'=2, say=1, feeling!=1, whether=2, cats?'=1, quite=4, know,'=1, all=10, t  
=1, it.

'Well!=1, were=8, And=2, out?of?the?way=2, Dinah=1, dreamy=1, 'poison,'=2, and=56, Of=1, saying=2, row=1, fifteen=1,  
hought=8, taste=1, Antipathies=1, led=1, daisy?chain=1, decided=1, tumbling=1, tea?time.=1, jar=2, scolded=1, began=

Alice=1, eye=1, must=2, as=1, locked=1, cut=1, stupid)=1, another=3, ONE=1, killing=1, Would=1, two=2, worth=1, mo  
iced=2, anxiously=1, label=1, No=1, such=3, watch=1, ask=1, down=2, begin.=1, knelt=1, thing=1, down=4, through=

thing.

After=1, sleepy=1, those=2, end=1, curtsy=1, glass=1, might=4, loveliest=1, out=1, girl=1, longer=1, name=1, play  
=1, not=14, crying=1, now=2, somewhere.'

Down=1, house!=1, lately=1, passage=2, was=51, First=2, suddenly=3, way=6, manage=1, listening=1, what=9, falle  
imb=1, 'Do=2, remembered=1, her=24, golden=3, (Dinah=1, catch=1, severely=1, it'll=2, high=1, flowers=1, having=3, w

She=1, WAISTCOAT?POCKET=1, (it=1, low=2, seldom=1, think!=1, However=1, Ma'am=1, more=1, larger=1, centre=1, blo  
red=1, small=2, door=3, shut=1, before=3, over)=1, tell=1, telescope!=1, simple=1, listen=1, bats?=1, wondering=1, '  
2, lost:=1, "poison"=1, twice=1, to?'=1, hung=1, hurry.=1, eaten=1, Australia?'=1, three?legged=1, who=1, it)=1, gam  
saw=1, (Which=1, say=4, enough=1, ears=2, glad=1, stupid=1, been=1, dipped=1, garden.=1, bring=1, couldn't=1, garden,

Either=1, happen=1, 'And=1, grow=2, going=6, shrink=1, past=1, milk=1, opportunity=1, an=2, How=3, as=12, at=11, roof

There=1, eats=1, trying=2, beds=1, hedge.

In=1, 'and=4, be=13, bottle=4, hot=2, eyes=1, waited=1, how=6, see=6, by=6, however=1, dear!=3, 'Oh=2, mind=2, set=1

Alice=1, answer=1, taught=1, CURTSEYING=1, under=3, did=3, conversation?'

So=1, sometimes=1, ITS=1, way?'=1, do=3, cheated=1, down=14, hold=1, She=4, bit=1, bats?=1, now,'=1, slowly=1, for

There=1, anything=1, remained=1, creep=1, she=73, never=4, take=1, for=1, learnt=1, overhead=1, true.)

Down=1, spoke?fancy=1, little=15, some=1, deep=1, question=1, tears=1, beginning=1, waiting=1, rather=3, for=17, ba  
, up=1, pegs.=1, 'No=1, aloud=1, showing=1, tunnel=1, do=1, very=19, passed=1, 'I=6, practice=1, Which=1, (when=  
, me,'=1, likely=1, large=2, in=24, made=2, is=5, it=41, drink=1, hoping=1, printed=1, field=1, gave=1, turned=2, unp  
e=3, off=4, generally=2, deep=1, candle=2, bats=1, THAT=1, feet=1, use=5, several=2, me!=1, feel=2, key=1, second  
ort=3, my=5, picking=1, friends=1, labelled=1, cherry?tart=1, cat.)=1, 'Which=1, THROUGH=1, dream=1, curtain=1, rabb

Presently=1, advise=1, of=40, Dinah=2, heap=1, make=2, glass=1, hear=2, eyes=1, on=14, falling=2, certainly=1, or=

Suddenly=1, stairs!=1, feet=1, getting!=1, work=1, sitting=1, coming=1, 'after=1, empty:=1, it?)=1, NEVER=1, about  
itude=2, 'Drink=1, stopping=1, wander=1, except=1, lessons=1, lock=1, respectable=1, seen=2, long=1, seem=1, fell=4,  
remember=3, into=9, croquet=1, before,'=1, managed=1, distance?but=1, plainly=1, so=12, key=3, ought=1, surprised=1,  
But=1, though=1, one=8, makes=2, started=1, many=2, people=2, small=1, actually=1, see:=1, face=1, way.

So=1, to=71, open=1, but=16, country=1, Latitude=2, rules=2, locks=1, bat=1, up=11, written=1, maps=1, advice=1, th  
is=10, felt=2, finding=2, look=3, toast,)=1, remarkable=1, 'Well=1, NOT=1, once=3, know=1, drop=1, 'DRINK=1, saucer=  
1, idea=1, WAS=1, jumped=1, that's=2, fountains=1, shelves=1, life=1, bats=1, common=1, wonder=5, sleepy=1, slippery  
=1, every=1, I'll=3, size:=1, lovely=1, histories=1, indeed=1, itself=1, knife=1, disagree=1, reach=2, bat?'=1, in  
ches=2, she'll=1, children=1, long=1, long=1, deeply=1, still=2, come=2, off.

'What=1, they'll=2, I'm=1, (Alice=1, doors=2, had=1, among=2, herself=6, Alice's=1, natural)=1, pictures=3, out=9,  
across=2, 'for=1, fall=4, get=7, dark=3, eat=6, finger=1, afraid=1, cool=1, tired=2, Alice,)=1, nice=2, buttered=1,  
reading=1, first=1, lamps=1, passage=1, own=2, her=1, sound=1, begun=2, door=3, 'in=1, rat?hole=1, schoolroom=1  
, 'it=1, sharply=1, once=1, should=2, only=2, from=3, moment:=1, wind=1, seen:=1, expecting=1, well=1, curious=2,  
like=11, finished=2, alas!=1, book,'=1, altogether=1, Zealand=1, herself=7, here=4, red?hot=1, her.

There=1, again.

The=1, can=2, holding=1, middle=1, paper=1, that!=1, you=1, walking=1, letters.

It=1, said=7, will=1, hurrying=1, really=1, round=2, asking!=1, daisies=1, door=1, think?=2, wild=1}

図2 WordCountの出力結果

【考察】

初めてHashMapを用いて、プログラムを記述した。Pythonでの辞書のようなモノだと僕の中では解釈した。また、javaでのテキストファイルを読み込む大変さが身にしみた。静的型付け言語で、ファイルの変換などを行うのは、思ったよりも多くの事をしなければいけないことを初めて知った。throws IOExceptionを付け加える時はVSCodeの機能を用いて、付け加えた。そのため、コードを記述し、エラーが発生したときは、VSCodeのFixなどを用いて、コードを簡単に記述していきたいと思う。

【参考文献】

1. 「アルゴリズムとデータ解析の授業スライド、HashMapについて」
2. 「No enclosing instance of type Hoge is accessible.」<<https://qiita.com/watanabk/items/738988fac29e1e1d8d88>>
3. 増井敏克(2020)「Pythonで始めるアルゴリズム入門」翔泳社
4. 「参考文献の書き方」<[http://www7a.biglobe.ne.jp/nifongo/ron/ron\\_04.html](http://www7a.biglobe.ne.jp/nifongo/ron/ron_04.html)>