

Special Care Set for Java Beginners/ Ex 14

• Notice!

The motivation of this page is to lead regular exercises.
We prepare many useful information in this page as hints of exercises, but this page has no answer.
If you have question, please don't hesitate to ask and discuss your teacher and teaching assistants.
このページは、通常の演習問題への導きとして作られています。
多くの有用な情報をヒントとして準備しておりますが、ここに回答が隠されているとかはありません。
もし、演習に対して質問等ございましたら、遠慮なく教員・TAに質問・議論をしてください。

Example Problem 1: Scannerクラスを使った入力 Input using Scanner class

今週は、文字列や数字の変換、様々なビルトインクラスを使うことを勉強しています。
先週、Input/Output (I/O) について勉強して、ファイル入出力などを行うことを勉強しましたが、
今回はScannerクラスを使って、様々なInput Streamを読み込むことをしましょう。
サンプルコードを入力して、コンパイルしてみてください。

1. 標準入出力からScannerクラスを使う例

```
import java.util.Scanner;

public class ScannerTest {

    public static void main(String[] args) {

        System.out.println("文字を入力して下さい。");

        Scanner scan = new Scanner(System.in);

        String str = scan.next();
        System.out.println("最初のトークンは: "+ str);

        str = scan.next();
        System.out.println("次のトークンは    : "+ str);

    }

}
```

Output Result:

文字を入力して下さい。

Foo Bar

最初のトークンは: Foo

次のトークンは : Bar

2. ファイル入力からScannerクラスを使う例

```
import java.util.Scanner;
import java.io.FileNotFoundException;
import java.io.File;

class ScannerFile{
    public static void main(String args[]){
        try{
            File file = new File("targetfile.txt");
            Scanner scan = new Scanner(file);
            scan.useDelimiter("¥¥r¥¥n");

            int line = 1;

            while(scan.hasNext()){
                String str = scan.next();
                System.out.println(line + ":" + str);
                line++;
            }
        }catch(FileNotFoundException e){
            System.out.println(e);
        }
    }
}
```

(targetfile.txt)

Hello, I am Aizu Nuri.

Hello, I am Inawa Shiroko.

Hi, I'm Ken.

Output Result:

1:Hello, I am Aizu Nuri.

2:Hello, I am Inawa Shiroko.

3:Hi, I'm Ken.

なお、この Scanner クラスのメソッド next() は、入力に対して自動的に、文字列の『空白』を区切りにして、"単語"を取得できます。

これは、長い文章を入力とする場合、非常に有効ですが、一方で、カンマ (,) やアポストロフィ (') 等がついたものは1つの単語として処理されてしまうので注意が必要です。

Java API: Scanner <http://docs.oracle.com/javase/jp/6/api/java/util/Scanner.html>

Example Problem 2: 文字列の加工

String Processing

Stringクラスは、もう既に私たちは勉強して、文字列を格納して簡単に使えることを知っていますが、Stringクラスの中には、文字列を簡単に加工するメソッドがいくつかあります。

1. 文字列の分割 String.split(String Regex)

```
class ExSplit {
    public static void main(String[] args) {
        //(1)分割する文字列を生成
        String str = new String("java,Java,JAVA");
        //(2)文字列をsplitメソッドで分割
        String[] strArray = str.split(",");

        //(3)分割された文字列の表示
        for (int i=0; i<strArray.length; i++) {
            System.out.println(strArray[i]);
        }
    }
}
```

splitメソッドに引き渡された正規表現にマッチするものを用いて、文字列を分割します。

Output Result:

```
java
Java
JAVA
```

2. 文字列の置換 String.replaceAll(String target, String replace)

```
public class ReplaceTest {
    public static void main(String[] args) {
        String name = "Programming Java 1: Happy Programming";
        name = name.replaceAll("Programming", "Application");
        System.out.println(name);
    }
}
```

replaceAllは、文字列中に含まれるすべてのものを、指定の文字列に置換します。

Output Result:

```
Application Java 1: Happy Application
```

Stringクラスはこのほかにも沢山メソッドがありますので、APIを参考に使用してみてください。

<http://docs.oracle.com/javase/jp/6/api/java/lang/String.html>