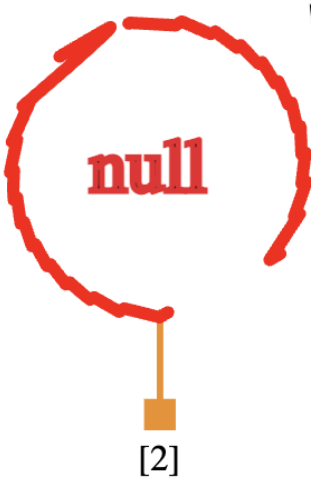
NULL

Digital Nothing

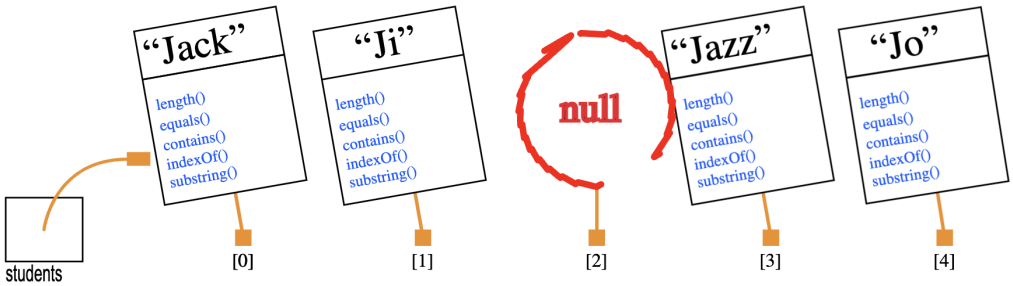


Scenario

A language analyst is working with sets of data relating to human names.

The analyst is processing the set of data and counting how many of the names are a certain length.

The problem is that the data is stored in a String array and not all of the elements reference a String - some of the elements reference **null** - digital nothing:



Why is this a problem? When we process the array, we want to use the .length() method of the String class (family) eg:

```
int count = 0;
for(int i = 0; i<students.length; i++){
    if(students[i].length() > 4){
       count = count + 1;
    }
}
```

But, in the example, ***students[2]*** does not reference a String and so there is no ***.length()*** method available! The program will crash when we execute it! In Java, a ***null pointer exception*** error will occur.

So, what is the solution to this common problem when working with arrays of Strings? We have to determine ***if the element in the array references null*** or not before we do any processing:

```
int count = 0;
for(int i = 0; i<students.length; i++){
     if(students[i]!=null){
        if(students[i].length() > 4){
          count = count + 1;
        }
     }
}
```

Optionally, we could combine the logic in the two if statements:

```
int count = 0;
for(int i = 0; i<students.length; i++){
     if(students[i]!=null && students[i].length()>4){
        count = count + 1;
     }
}
```

Task

A function, countDomain, processes an array of Strings and a String:

public static int countDomain(String[] addresses, String domain)

The function analyzes each email address in the String array and checks to see if the domain is in the address. For example, if a String array contains the following:

```
String[ ] emails = {"jack@hotmail.com", null , "joe@hotmail.com", "jeff@msn.com , null};
```

And we call the function:

```
int count = countDomain(emails, "hotmail");
```

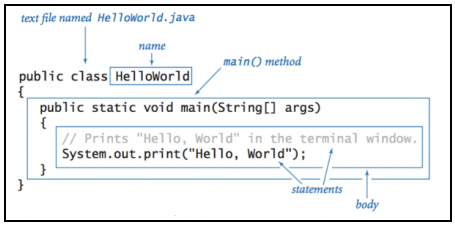
then count will store the value 2.

[See next page]

Code the function `countDomain`, including its signature:

JAVA CHEAT SHEET

type	*set of values*	*common operators*	*sample literal values*
`int`	integers	`+ - * / %`	`99 12 2147483647`
`double`	floating-point numbers	`+ - * /`	`3.14 2.5 6.022e23`
`boolean`	boolean values	`&& || !`	`true false`
`char`	characters		`'A' '1' '%' '\n'`
`String`	sequences of characters	`+`	`"AB" "Hello" "2.5"`



Printing.

`void System.out.print(String s)`	*print s*
`void System.out.println(String s)`	*print s, followed by a newline*
`void System.out.println()`	*print a newline*

Nested if-else statement.

```
if      (income <      0) rate = 0.00;
else if (income <   8925) rate = 0.10;
else if (income <  36250) rate = 0.15;
else if (income <  87850) rate = 0.23;
else if (income < 183250) rate = 0.28;
else if (income < 398350) rate = 0.33;
else if (income < 400000) rate = 0.35;
else                      rate = 0.396;
```

```
Scanner scan = new Scanner(System.in);

Double num1 = scan.nextDouble();

Int num2 = scan.nextInt();

String word = scan.nextLine();
```

Comparison operators.

op	*meaning*	*true*	*false*
`==`	*equal*	`2 == 2`	`2 == 3`
`!=`	*not equal*	`3 != 2`	`2 != 2`
`<`	*less than*	`2 < 13`	`2 < 2`
`<=`	*less than or equal*	`2 <= 2`	`3 <= 2`
`>`	*greater than*	`13 > 2`	`2 > 13`
`>=`	*greater than or equal*	`3 >= 2`	`2 >= 3`

```
int rand = (int)(Math.random()*100);
```

Booleans.

values	*true or false*		
literals	`true`	`false`	
operations	and	or	not
operators	`&&`	`||`	`!`