

Forum:[Hack Forums](#)

Thread: [Guide to the Java Forum | Etiquette and FAQ | Read Before Posting!](#)

_wind - February 15, 2016, 11:32 pm

Java Section Info, Etiquette, and FAQ

After spending a lot of time in this forum, I noticed a lot of recurring mistakes that people make -- which can be easily prevented by following a set of guidelines.

Anyways, here's the guide to the Java Forum, with basic information, do's and don't's, and some common questions and answers.

What is the Java Section?

The Java section is:

for sharing your projects with the HF community

a place to ask for help when you can't find any

for help others

a place to learn and improve your programming skills

The Java section is NOT:

a place to ask others to do your homework

for spreading viruses/malware

for JavaScript, that's belongs [here](#)

Posting Guidelines

Don't post your homework assignment and try to get others to do it just because you're lazy. Programming is supposed to be fun, and you'll only enjoy it through learning. Getting others to do your work while you lean back doesn't teach you anything, and you'll be back to square one in no time.

Don't post a thread along these guidelines: "hi add me on skype if you are a java expert kthxbye"

Please do not post code dumps without comments. They are low quality and do not provide any explanation. If you're posting a solution, explain the process, step by step.

If your problem was solved, don't just lock your thread and say thanks. Choose the best solution,

and if there isn't one, post your own code and explain it. This helps others in the future when they encounter the same issue as you.

When posting, it's best to mention what you've tried so far, show that you've spent effort, and POST your code! I can't stress how many posts I've seen saying I have x problem, and want y solution. It's kind of hard to do that if you're not providing what you have so far.

Formatting Code

It's recommended that you stick to the following format when posting your code, as it'll make work much easier for those helping you.

Where do I post my code?

Use the BBTags

<http://pastebin.com>

GitHub, if a large project with several classes/packages

Indents

Don't be this guy:

```
Code import java.io.*;
import java.util.Scanner;
public class minesweeper {
static char[][] table;
static int rows = 1;
static int columns = 1;
static Scanner sc;
static String[] d;
static int hasMines(int i, int j){
int count = 0;

if(i-1 != -1){
if(table[i-1][j] == "*").charAt(0)){
count++;
}
if(j-1 != -1){
if(table[i-1][j-1] == "*").charAt(0)){
count++;
}
}
}
if(i+1 != rows){
if(table[i+1][j] == "*").charAt(0)){
count++;
}
if(j+1 != columns){
if(table[i+1][j+1] == "*").charAt(0){
```

```

count++;
}
}
}
if(j-1 != -1){
if(table[i][j-1] == "*".charAt(0)){
count++;
}
if(i+1 != rows){
if(table[i+1][j-1] == "*".charAt(0)){
count++;
}
}
}
if(j+1 != columns){
if(table[i][j+1] == "*".charAt(0)){
count++;
}
if(i-1 != -1){
if(table[i-1][j+1] == "*".charAt(0)){
count++;
}
}
}
return count;
}
public static void main(String[] args) throws IOException{
String line = null;
int count = 0;
while(rows != 0){
sc = new Scanner(System.in);
if(sc.hasNextLine()){
d = sc.nextLine().split("\\s+");
}
rows = Integer.parseInt(d[0]);
columns = Integer.parseInt(d[1]);
table = new char[rows][columns];
if(rows != 0 && columns != 0){
count++;
}
for(int i = 0; i < rows;i++){
String input = sc.nextLine();
char[] temp = input.toCharArray();
for(int j = 0; j < columns; j++){
table[i][j] = temp[j];
}
}
for(int i = 0; i < rows;i++){
for(int j = 0; j < columns; j++){
if(table[i][j] == ".".charAt(0)){
table[i][j] = Character.forDigit(hasMines(i, j),10);
}
}
}
}

```

```

}

if(rows != 0 && columns != 0){
    System.out.println("Field #" + count);
}
for(int r=0; r<table.length; r++) {
    for(int c=0; c<table[r].length; c++)
        System.out.print(table[r][c]);
    System.out.println();
}
}
}
}
}

```

Indent your code with either four spaces or an indent with each new block.

Naming

Rather simple, but a common mistake that often confuses me when reading others' code.

Variables and Methods are named starting with a lowercase first letter, followed by uppercase first letters of following words.

Code `int myInteger;`

`String[] myStringArray;`

`boolean aVeryCoolBoolean;`

Classes are named with uppercase letters on each word.

Code `public class HelloWorld{`

`private abstract class Forum{`

Frequently Asked Questions

Why doesn't my String comparison work?

Example problem:

Code `String myString = "foobar";`

`if(myString == "foobar") System.out.println("yay!");`

The above is wrong, as Strings are compared using `myString.equals("foobar");` instead of the `==` operator.

When do I use Arrays? ArrayLists?

Arrays have fixed lengths, so use them when you know the length of your list.

Array Syntax ([Click to View](#))

To create a new array, put your datatype, followed by a pair of square brackets. At this stage, you can also directly initialize its values with curly brackets.

Code `int[] integerArray = new int[10];` //creates an integer array of length 10

`String[] strArray = {"foo", "bar", "foobar"};` //creates a String array with existing values

Looping through arrays:

There are two ways to loop through Arrays -- but one returns an index value, and one does not.

Code `int[] myArray = {1, 2, 3, 4, 5};`

//method 1, has index i

`for(int i = 0; i < myArray.length; i++){`

`System.out.println(myArray[i]);`

`}`

//method 2, directly gets value

`for(int value : myArray){`

`System.out.println(value);`

```
}
```

ArrayLists are flexible and you can add items without having to define a fixed length.

ArrayList Syntax ([Click to View](#))

Creating a new ArrayList:

Code `ArrayList<Integer> myArrayList = new ArrayList<Integer>();` Note how I use Integer instead of int, as you need a class, and not a primitive type.

Adding values:

Code `myArrayList.add(5);` // adds the #5 to arraylist

Removing values:

Code `myArrayList.remove(0);` //removes element from index 0

`myArrayList.clear();` //remove all values

Getting/Setting values:

Code `myArrayList.contains(5);` // returns true or false if your array contains value

`myArrayList.get(0);` //gets value at index 0

`myArrayList.size();` //gets the length/size of arraylist

Looping -- again, there are two ways:

Code //method 1, has index

```
for(int i = 0; i < myArrayList.size(); i++){  
    System.out.println(myArrayList.get(i));  
}
```

//method 2, no index

```
for(int value : myArrayList){  
    System.out.println(value);  
}
```

NullPointerException

Every single person who has coded Java gets this error at one point. This is caused by a variable that you're calling that has a null (nonexistent) value, which throws an error.

Solution: Look at the error dump, and refer to the line which the error occurred on. Check to see that each variable has been defined. At times* you'll need to check if the variable is == null, and only run your code if it isn't.

How do I use a library?

Maybe you've gone online and found a library that you want to use in your project. After all, it wouldn't be practical to re-invent the wheel. To add a library to your Java project (Eclipse) right click on your project, go to 'Build Path' -> 'Configure Build Path' -> 'Add external jars' -> Select, and import.

Why does my code keep running without stopping?

Chances are, you're in a while loop and either:

Your statement is faulty

the break; statement is not running (only if you have it)

General notes:

`while(true){}` will run forever and ever and ever until you break out of it.

`while(comparison){}` will run until the comparison becomes false

break; breaks out of a loop

How do I convert from one data type to another?

Maybe you want to convert an integer to a String, or vice-versa.

Or maybe a int to double, or any other conversion.

Converting datatype x to a String has two options:

Code //method 1

someObject.toString();

//method 2

String.parse(someObject);

And method 2 works for most other data types as well. For example, converting a double to integer would look like:

Code Integer.parse(myDouble);

Why am I not writing to file?

A common mistake when writing text to a file would be not closing the writer.

Example:

Code PrintWriter writer = new PrintWriter("myfilename.txt", "UTF-8");

writer.println("this is line 1");

writer.println("line numero dos");

This won't write anything to the file, because you're missing the

Code writer.close(); statement. This applies not only to PrintWriter, but other types such as BufferedWriter as well.

My code doesn't work/crashes, but I'm not seeing any errors?

Make sure you don't have an empty try/catch statement with exceptions. A common mistake is doing the following:

Code try {

// Your code goes here..

}catch (Exception e){

//nothings in here, it's not like it matters or anything... right?

}

}

Make sure that you're always printing the stack trace when handling exception. You can do this with

Code e.printStackTrace();

My program skips over the first value in array?

Don't forget that Java (and every other programming language) is zero-indexed, meaning that you count from zero. Say you had an array:

Code int[] myArray = {1, 2, 3, 4, 5};

So myArray[0] is equal to 1

myArray[1] is equal to 2

myArray[2] is equal to 3

and so on...

That's basically it. There are also probably a lot of JavaFX questions that I missed, but I don't have experience in that. [+Haze](#) would be your best bet.

Happy coding!

csrk - *February 16, 2016, 2:30 am*

HQ post, but isn't a moderator supposed to do this?

hoLe[R]a - *February 16, 2016, 2:51 am*

I thought this would be a redundant thread but found some interesting things here actually.

Sunquyman - *February 16, 2016, 4:31 am*

Nice, can this get stickied? It would definitely help a lot for newer users

youw0t - *February 16, 2016, 8:09 am*

Definitely a nice addition to the forum and very helpful for people that are new to Java.