



This presentation is about processing command-line arguments of your Java program.

This is a relatively short presentation so lets get started.

Topics

Command-line Arguments

Using Command-line Arguments

Demonstration

Wrap-up

The topics of this presentation are:

- What are command-line Arguments
- How to use command-line arguments in your Java program
- A short demonstration using Java command-line arguments
- And a Wrap-up of the major points of this presentation

Lets start with an introduction to command-line arguments.

Command-line Arguments

Provide data to Java program at run-time

- Very the execution of the program

- Usually small amount of data

String array parameter to main() method

- ...main(String [] args) ...

 - args is the traditional name, you can change it

Provided as part of java command

- java myJavaProgram "arg1" "arg2"

Command-line arguments are a way to provide data to your Java program which you can vary each time you run your program. This allows you to modify the execution path of your program without modifying your program code.

Command-line arguments are usually used to supply a small amount of data to your program. However, they can have a significant impact on the program results , for example, by supplying a name of a file to process.

The JVM supplies the program arguments via a String array. This is reflected in the signature of the main() method which is the entry-point of your Java program. The String args array parameter is how the command-line arguments are communicated to your Java program. Args is the traditional name of this array, going back to the early days of the C language. You can change the name of the array if you wish.

You supply the command-line arguments as part of the java run-time command as shown here. The arguments are listed after your entry class name with one or more spaces in-between. Note if each string is a single word, you do not have to put quotes around them but it is fine and better practice to do so.

Lets look at this in a bit more detail.

Using Command-line Arguments

Use like any String array method parameter

Should validate before use

Use quotes for multiple parts of same argument

`java myJavaProgram "line 1 is this" "line2 is that"`

IDEs provide mechanism to supply arguments

You can use the command –line arguments just like any String array using indexing to read the individual arguments.

Just like with other methods, you should validate the arguments before use since they are coming from a source external to the method, in this case the user.

If an individual argument consists of multiple words, you need to quote the string, otherwise each word would be treated as a separate argument. You can also put quotes around single-word arguments if desired although not required as mentioned previously.

Normally you use command line arguments as part of java command line but this makes developing your program somewhat inconvenient. Java Integrated Development Environments (IDEs) usually provide a way to supply run-time arguments while developing your program.

Lets look at a demonstration on using Java command-line arguments.

Demonstration Time!

The Wrap-Up

Java program arguments provide run-time data input

- Useful for small amounts of data

- Arguments placed on java command line

String parameter to the main() method

- Treat like any method String parameter

OK here is the wrap-up:

- Java program arguments is one way to provide run-time data input to your program.
 - These arguments are generally useful for supplying a small amount of run-time data.
 - Arguments are placed on the command-line after the entry Java class name
- Command-line arguments are reflected in your Java program as the String array parameter of the main() method. This String array can be treated just like any String array parameter.

That's it!



And that's it!