

Command-line Arguments in Java

By
M. BABY ANUSHA,
ASST.PROF IN CSE DEPT.,
RGUKT,NUZVID

Objectives

At the end of the lesson, the student should be able to:

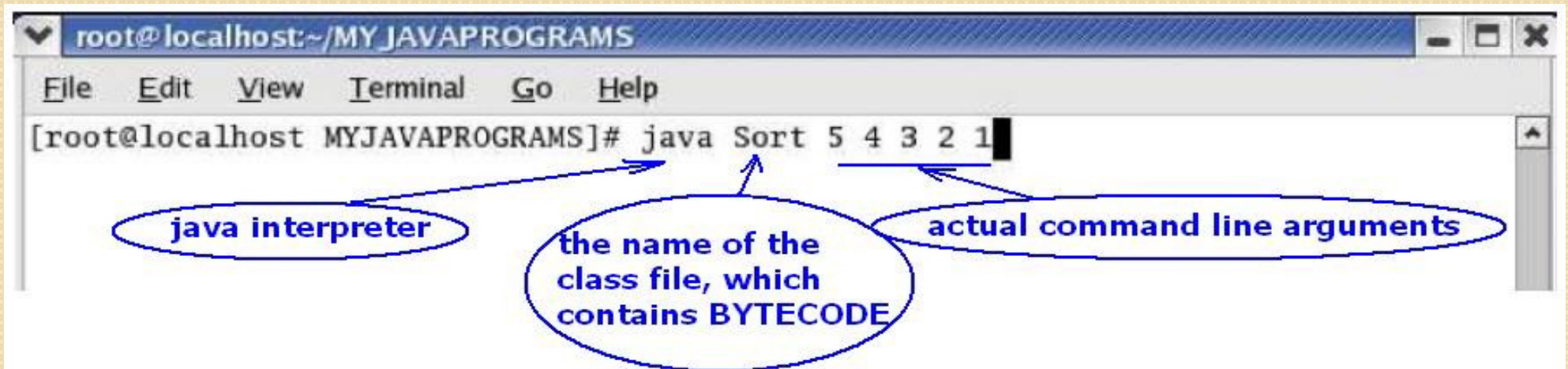
- Know and explain what a command-line argument is
- Get input from the user using command-line arguments
- How to pass command-line arguments in Command prompt

Command-line Arguments

- Command line arguments means a way to pass parameters to the main function while executing the program.
- A Java application can accept any number of arguments from the command-line.
- Command-line arguments allow the user to affect the operation of an application.
- The user enters command-line arguments when invoking the application and specifies them after the name of the class to run.

Command-line Arguments

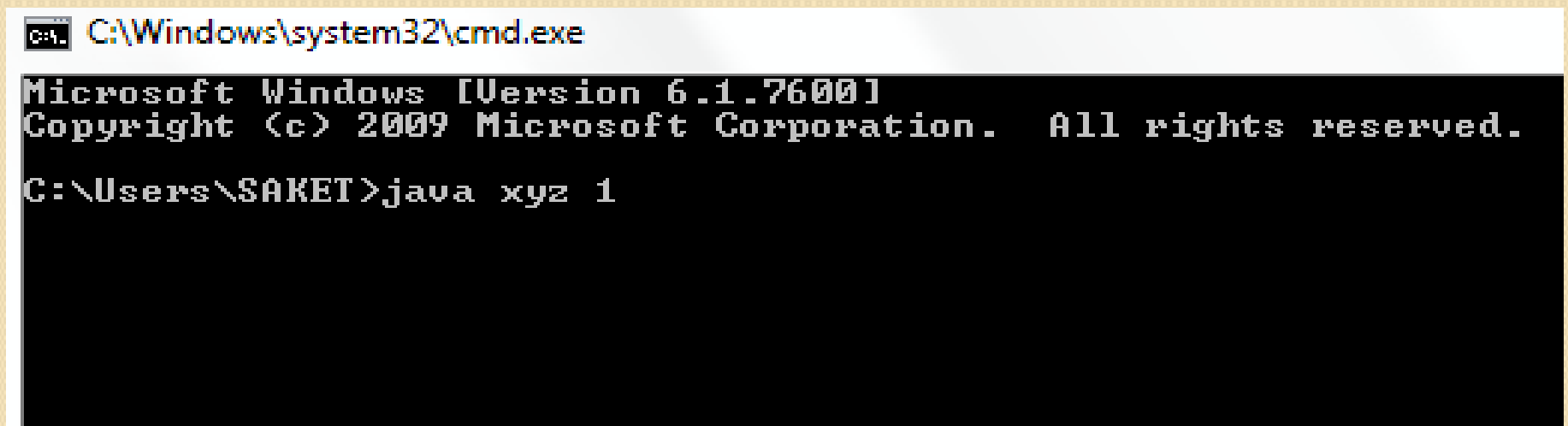
- For example, suppose you have a Java application, called Sort, that sorts five numbers, you run it like



- Note: The arguments are separated by spaces.**

Command line arguments in DOS

- The following screen image shows **Command line arguments in DOS:**



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\SAKET>java xyz 1
```

The user enters command-line arguments when invoking the application and specifies them after the name of the class to be run.

How Java Application Receive Command-line Arguments

- In Java, when you invoke an application, the runtime system passes the command-line arguments to the application's main method via an array of Strings.

public static void main(String[] args)

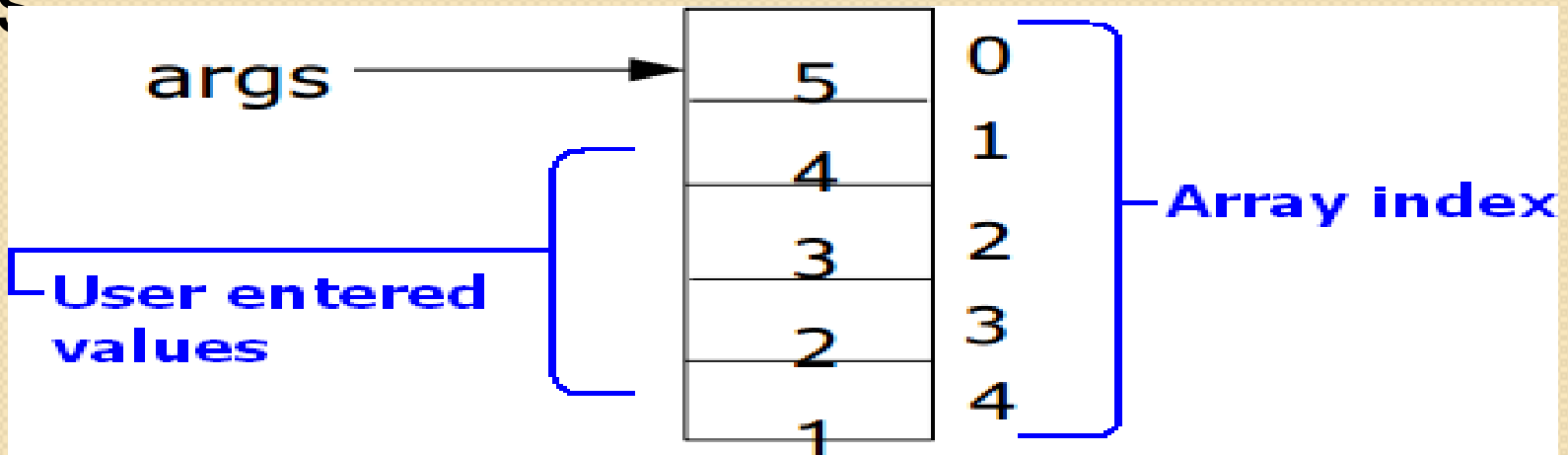
- Each String in the array contains one of the command-line arguments.

args[] array

- Given the previous example where we run:

java Sort 54321

- the arguments are stored in the args array of the main method declaration as



Example :

To print the array of arguments, we write:

```
public class CommandLineSample
{

    public static void main( String args[])
    {
        for(int i=0; i<args.length; i++)
        { System.out.println( args[i] );
        }
    }
}
```


Conversion of Command-line Arguments

- If your program needs to support a numeric command-line argument, it must convert a `String` argument that represents a number, such as "34", to a number.
- Here's a code snippet that converts a command-line argument to an integer,

```
int firstArg = 0;  
    if (args.length > 0){  
        firstArg = Integer.parseInt(args[0]);  
    }
```

Command-line Arguments: Coding Guidelines

- Before using command-line arguments, always check the number of arguments before accessing the array elements so that there will be no exception generated.
- For example, if your program needs the user to input 5 arguments,

```
if( args.length!= 5 ){  
    System.out.println("Invalid number of arguments");  
    System.out.println("Please enter 5 arguments");  
}  
else{  
    //some statements here  
}
```

Example :

- The code snippet shows the switch method working through command line

Switch.java

```
/**
 * Switch.java
 * Integer passing in command-line arguments.
 */
public class Switch
{
    public static void main(String key[])
    {

        int x=Integer.parseInt(key[0]);
        switch(x) {
            case 1: System.out.println("Monday");
                    break;
            case 2: System.out.println("Tuesday");
                    break;
            case 3: System.out.println("Wednesday");
                    break;
            case 4: System.out.println("Thursday");
                    break;
            case 5: System.out.println("Friday");
                    break;
            case 6: System.out.println("Saturday");
                    break;
            case 7: System.out.println("Sunday");
                    break;
            default :System.out.println("Invalid Number of Day");
                    }
        }
    }
}
```


Compiling and executing Switch.java

Compiling Switch.java

C:\Windows\system32\cmd.exe

```
C:\saket>javac Switch.java
```

```
C:\saket>java Switch 5  
Friday
```

```
C:\saket>java Switch 9  
Invalid Number of Day
```

```
C:\saket>java Switch 3  
Wednesday
```

```
C:\saket>
```

Another Example...

StringCLA.java

StringCLA.java displays each of its command line arguments on a line by itself.

```
/**
 * StringCLA.java
 * String passing in command-line arguments.
 */
class StringCLA {

    public static void main(String[] str) {
        int length = str.length;
        if (length <= 0) {
            System.out.println("Enter Some String");
        }
        for (int i = 0; i < length; i++) {
            System.out.println(str[i]);
        }
    }
}
```

Summary :

- Command-line arguments
 - How to access the arguments
 - How to convert String arguments to integer using Integer.parseInt method
 - How to pass command-line arguments in Command prompt



Thank you!

shutterstock.com · 567687052