String class

String equality

String methods

String conversions

String Builder

String class

Stores a sequence of Unicode characters.

UTF-64 format

-Literals are enclosed in double quotes

-string support concatenation

+=

String immutable

String variabl do not directly hold the String vaiue

-Hold the reference of the instance of String

- Changes in the value create a new instance of the string.

package com.org.understandingmethods;  
  
public class CommandLinergument {  
  
 public static void main(String[] args) {  
 if (args.length < 1) {  
 System.*out*.println("No args provided");  
 } else {  
 for (String arg : args) {  
 System.*out*.println(arg);  
 }  
 }  
 }  
}

package com.org.understandingmethods;  
  
public class SwitchStatementArray {  
  
 public static void main(String[] args) {  
  
 double[] leftvals = {100.0d, 25.0d, 22.0d, 11.0d};  
 double[] rightvals = {50.0d, 92.0d, 17.0d, 3.0d};  
 char[] opscode = {'d', 'a', 's', 'm'};  
 double res[] = new double[opscode.length];  
  
 if (args.length == 0) {  
 for (int i = 0; i < opscode.length; i++) {  
 res[i] = *execute*(opscode[i], leftvals[i], rightvals[i]);  
 }  
 System.*out*.println(res);  
 for (int i = 0; i < res.length; i++) {  
 System.*out*.println(res[i]);  
 }  
 } else if (args.length == 3) {  
 *handleCommandLine*(args);  
 } else {  
 System.*out*.println("Please provide an operation code and 2 numeric values");  
 }  
 }  
  
 private static void handleCommandLine(String[] args) {  
 char opsCode = args[0].charAt(0);  
 double leftVal = Double.*parseDouble*(args[1]);  
 double rightValue = Double.*parseDouble*(args[2]);  
 double res = *execute*(opsCode, leftVal, rightValue);  
 System.*out*.println(res);  
 }  
  
 static double execute(char opCode, double leftvalue, double rightValue) {  
 double result = 0.0d;  
  
 switch (opCode) {  
 case 'a':  
 result = leftvalue + rightValue;  
 break;  
 case 's':  
 result = leftvalue - rightValue;  
 break;  
 case 'm':  
 result = leftvalue \* rightValue;  
 break;  
 case 'd':  
 if (rightValue != 0) {  
 result = leftvalue / rightValue;  
 } else {  
 System.*out*.println("given number val2 is zero");  
 }  
 break;  
 default:  
 System.*out*.println("invalid input");  
 result = 0.0d;  
 }  
 return result;  
 }  
}

package com.org.Strings;  
  
public class StringImmutable {  
 public static void main(String[] args) {  
  
 //==  
 String s1 = "Hello";  
 s1 += "java";  
  
 String s2 = "Hello";  
 s2 += "java";  
  
 System.*out*.println(s1);  
 System.*out*.println(s2);  
  
   
 /\*  
 this is going to check address.  
 \*/  
 System.*out*.println(s1 == s2);  
 /\*  
 checking string equality. it will chekc value  
 \*/  
 System.*out*.println(s1.equals(s2));  
 }  
}

Interning:

package com.org.Strings;  
  
public class InterningMethod {  
  
 public static void main(String[] args) {  
  
  
 String s1="hello";  
 s1+=" java";  
 String s2="hello";  
 s2+=" java";  
  
 System.*out*.println(s1 == s2);  
 /\*  
 checking string equality. it will chekc value  
 \*/  
 System.*out*.println(s1.equals(s2));  
  
  
 String s3=s1.intern();  
 String s4=s2.intern();  
  
 System.*out*.println(s3 == s4);  
 /\*  
 checking string equality. it will chekc value  
 \*/  
 System.*out*.println(s3.equals(s4));  
  
  
  
 }  
}

package com.org.Strings;  
  
public class Methods {  
  
 public static void main(String[] args) {  
  
 /\*(  
  
 Operation:  
 Length-----length  
 Create new string from exsting - concat,replace,toLowercase,toUppercase,trim,split  
 extract substring=== charAt, subString  
 Test substring- contains,endsith,strtsWIth,indexOf,lastIndexof  
 comparision = equals, equalsignorecase,isempty,compareTo,comparetoIgrnorecase  
 Formating- format  
 \*/  
  
 String s = "Hel9o";  
 char c = s.charAt(3);  
 System.*out*.println("s length " + s.length() + "and char is " + c);  
 String sente = "sun,raise,in,the,east";  
 String[] res = sente.split(",");  
 for (String eachword : res) {  
 System.*out*.println(eachword);  
 }  
 }  
  
}

package com.org.Strings;  
  
public class Conversion {  
  
 public static void main(String[] args) {  
 int a=100;  
  
 String s = String.*valueOf*(a);  
  
 System.*out*.println(s);  
  
  
 String number="900";  
  
 int i = Integer.*parseInt*(number);  
  
 System.*out*.println(i);  
  
  
 }  
}

package com.org.Strings;  
  
import java.util.Scanner;  
  
public class SwitchStatementString {  
  
 public static void main(String[] args) {  
  
 double[] leftvals = {100.0d, 25.0d, 22.0d, 11.0d};  
 double[] rightvals = {50.0d, 92.0d, 17.0d, 3.0d};  
 char[] opscode = {'d', 'a', 's', 'm'};  
 double res[] = new double[opscode.length];  
  
 if (args.length == 0) {  
 for (int i = 0; i < opscode.length; i++) {  
 res[i] = *execute*(opscode[i], leftvals[i], rightvals[i]);  
 }  
 System.*out*.println(res);  
 for (int i = 0; i < res.length; i++) {  
 System.*out*.println(res[i]);  
 }  
 }  
 else if(args.length==1 && args[0].equals("interactive"))  
 {  
 *executeInteractively*();  
 }  
  
 else if (args.length == 3) {  
 *handleCommandLine*(args);  
 } else {  
 System.*out*.println("Please provide an operation code and 2 numeric values");  
 }  
 }  
  
 private static void handleCommandLine(String[] args) {  
 char opsCode = args[0].charAt(0);  
 double leftVal = Double.*parseDouble*(args[1]);  
 double rightValue = Double.*parseDouble*(args[2]);  
 double res = *execute*(opsCode, leftVal, rightValue);  
 System.*out*.println(res);  
 }  
  
 static double execute(char opCode, double leftvalue, double rightValue) {  
 double result = 0.0d;  
  
 switch (opCode) {  
 case 'a':  
 result = leftvalue + rightValue;  
 break;  
 case 's':  
 result = leftvalue - rightValue;  
 break;  
 case 'm':  
 result = leftvalue \* rightValue;  
 break;  
 case 'd':  
 if (rightValue != 0) {  
 result = leftvalue / rightValue;  
 } else {  
 System.*out*.println("given number val2 is zero");  
 }  
 break;  
 default:  
 System.*out*.println("invalid input");  
 result = 0.0d;  
 }  
 return result;  
 }  
  
 static void executeInteractively() {  
 System.*out*.println("Enter an operation and two number: ");  
 Scanner scanner = new Scanner(System.*in*);  
 String userInput = scanner.nextLine();  
 String[] parts = userInput.split(" ");  
 *performOperation*(parts);  
  
  
 }  
  
 private static void performOperation(String[] parts) {  
 char opsCode = *opsCodeFromString*(parts[0]);  
 double leftVal = *valueFromWord*(parts[1]);  
 double rightVal = *valueFromWord*(parts[2]);  
 double res = *execute*(opsCode, leftVal, rightVal);  
 System.*out*.println(res);  
  
 }  
  
 static char opsCodeFromString(String operationName) {  
 char opscode = operationName.charAt(0);  
 return opscode;  
 }  
  
 static double valueFromWord(String word) {  
  
 String[] numberWords = {  
 "zero", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine"  
 };  
 double val = 0.0d;  
 for (int index = 0; index < numberWords.length; index++) {  
 if (word.equals(numberWords[index])) {  
 val = index;  
 break;  
 }  
 }  
 return val;  
 }  
}

String Builder:

package com.org.Strings;  
  
public class StringBuilder2 {  
  
 /\*  
  
 provides mutable string buffer  
 Efficiently constructs string value  
 add new content to end with append  
 add new content within with insert  
  
 \*/  
  
 public static void main(String[] args) {  
 String location = "Florida";  
 int flightNumber = 175;  
 String time="9:00";  
  
 StringBuilder sb = new StringBuilder(40);  
  
 sb.append("i flew to ");  
 sb.append(location);  
 sb.append(" on Flight #");  
  
 sb.append(flightNumber);  
  
 int pos = sb.indexOf(" on");  
 System.*out*.println(pos);  
  
 sb.insert(pos," at ");  
  
 sb.insert(pos+4, time);  
  
 String string = sb.toString();  
  
 System.*out*.println(string);  
  
// String location = "Florida";  
// int flightNumber = 175;  
//  
// String res= "i flew to "+ location +" on FLighe #"+flightNumber;  
// System.out.println(res);  
  
  
 }  
  
  
}