**Java Program to find the largest and smallest word in a string (input with in the program)**

package friday;

public class SmallestLargestWord{

public static void main(String[] args){

String string = "Hardships often prepare ordinary people for an extraordinary destiny";

System.out.println("The given main string is:\n" +string);

String word = "", small = "", large="";

String[] words = new String[100];

int length = 0;

string = string + " ";

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

if(string.charAt(i) != ' '){

word = word + string.charAt(i);

}

else{

words[length] = word;

length++;

word = "";

}

}

small = large = words[0];

for(int k = 0; k < length; k++){

if(small.length() > words[k].length())

small = words[k];

if(large.length() < words[k].length())

large = words[k];

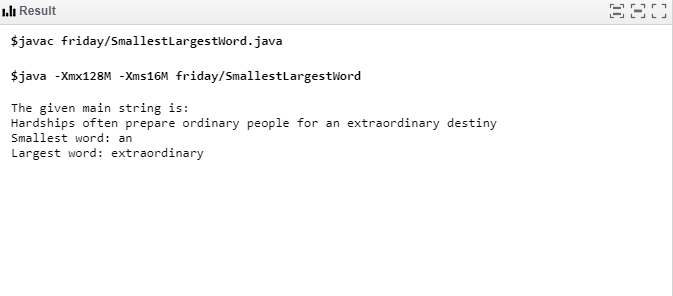
}

System.out.println("Smallest word: " + small);

System.out.println("Largest word: " + large);

} }

**OUTPUT**



**Java Program to find the largest and smallest word in a string(input from the keyboard)**

package friday;

import java.io.BufferedReader;

import java.io.InputStreamReader;

public class LargestAndSmallestWord {

static void printLargestAndSmallestWord(String str){

String[] arr=str.split(" ");

int i=0;

int maxlength,minlength;

maxlength=Integer.MIN\_VALUE;

minlength=Integer.MAX\_VALUE;

String largest,smallest;

largest = smallest = "";

for(i=0;i<arr.length;i++){

if(arr[i].length() < minlength){

smallest=arr[i];

minlength=arr[i].length();

}

if(arr[i].length() > maxlength) {

largest=arr[i];

maxlength=arr[i].length();

}

}

System.out.println("The largest and smallest word is: \"" + largest + "\" and \"" + smallest + "\"");

}

public static void main(String[] args) {

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the text string");

String str;

try{

str=br.readLine();

}

catch(Exception e){

System.out.println("Error reading input");

return;

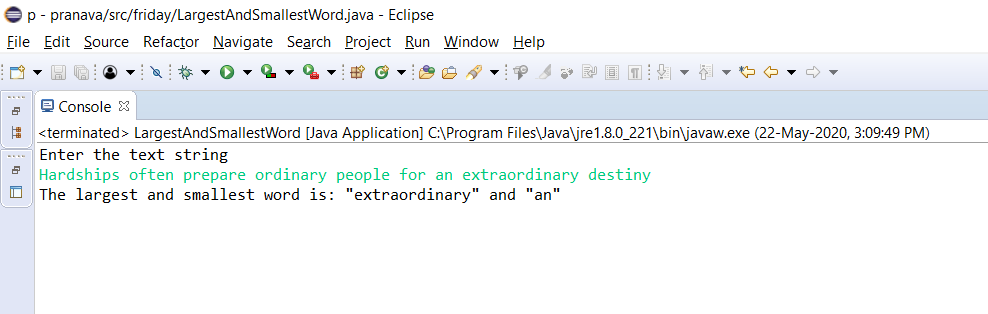
}

printLargestAndSmallestWord(str);

}

}

**OUTPUT**

****