Graphical user interface, text, application

Description automatically generated

Note: to get a precise average string length, I use double instead of int as the data type.   
  
public class Lab7 {  
 public static void main(String[] args) {  
 int total=0;  
 String[] ar={"rabbit","tiger","lion","cat","koala","emu","giraffe","penguin"};  
 for (String animal: ar){  
 total+=animal.length();  
 }  
 double average=total/(double)ar.length;  
 System.out.println(average);  
 }  
}  
  
output: 5.0

Graphical user interface, text, application, email

Description automatically generated

a. The error occurs in line 7 (ar[i]=ar[j];) and it shows ArrayIndexOutOfBoundsException.

b. In this code, the value of i remains 0, which means the while loop condition i< ar.length-1 remains true and never changes from true to false. In this situation, the statement block will be executed endlessly. the expression j+=2; will be executed endlessly. Initially, j=1, then j=3, then j=5, the j=7. When j=7, ar[j] is ar[7]. However, the array ar only has  6 elements, which means the index can only range from 0 to 6 and index 7 is out of the bounds. This is how the error occurs.

c. solution:

int[] ar = {2, 5, 2, 7, 8, 1};  
int i = 0;  
int j = i + 1;  
int t;  
while (i < ar.length - 1) {  
 t = ar[i];  
 ar[i] = ar[j];  
 ar[j] = t;  
 j += 2  
 i+=2;  
}

The solution is to add the expression "i+=2;" in the while loop so that i changes from 0 to 2 to 4 to 6. First of all, the value of ar[0] is assigned to t, so t=2.Then the value of ar[j] (ar[1]) is assigned to ar[0] and ar[0] becomes 5. Third, the value of t is assigned to ar[j] (ar[1]) and ar[1] becomes 2. Now the first two elements of the array complete swapping. After adding the expression “i+=2;”, j=3, i=2, and the current value of i (which is 2 currently) is evaluated in the while condition statement. Similar to the previous steps, this while loop will now begin to swap the 3rd and 4th elements of the array ar. And j=5,i=4. As i(4)is smaller than ar.length-1 (5), this while loop will be executed again. Now the 5th and 6thelements of the array ar will be swapped. After swapping, j=7, i=6, and till this point, the while condition statement becomes false, and the while loop will not be executed any more. Therefore, the index i and j will not exceed the index boundary of the array.

Graphical user interface, text, application

Description automatically generated

public class Lab7 {  
 public static void main(String[] args) {  
 String[] ar = {"abc", "defghi", "d", "week7", "olivia","hu","hello"};  
 int n = 1;  
 int i = 0;  
 while (i < ar.length && n<=2) {  
 if (ar[i].length() % 2 != 0) {  
 System.out.println(ar[i]);  
 n++;  
 }  
 i++;  
 }  
 }  
}  
  
output:   
abc  
d