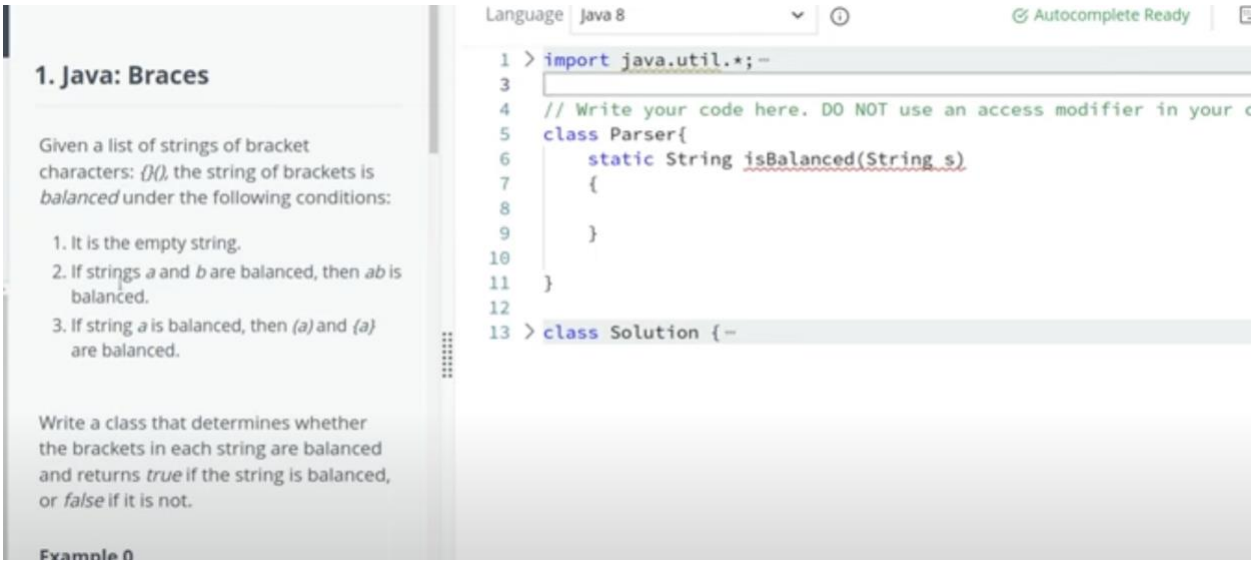
Java Basic Questions

1. **Java: Braces**



1. Java: Braces

Given a list of strings of bracket characters: *{}()*, the string of brackets is *balanced* under the following conditions:

1. It is the empty string.
2. If strings *a* and *b* are balanced, then *ab* is balanced.
3. If string *a* is balanced, then *(a)* and *{a}* are balanced.

Write a class that determines whether the brackets in each string are balanced and returns *true* if the string is balanced, or *false* if it is not.

Example 0

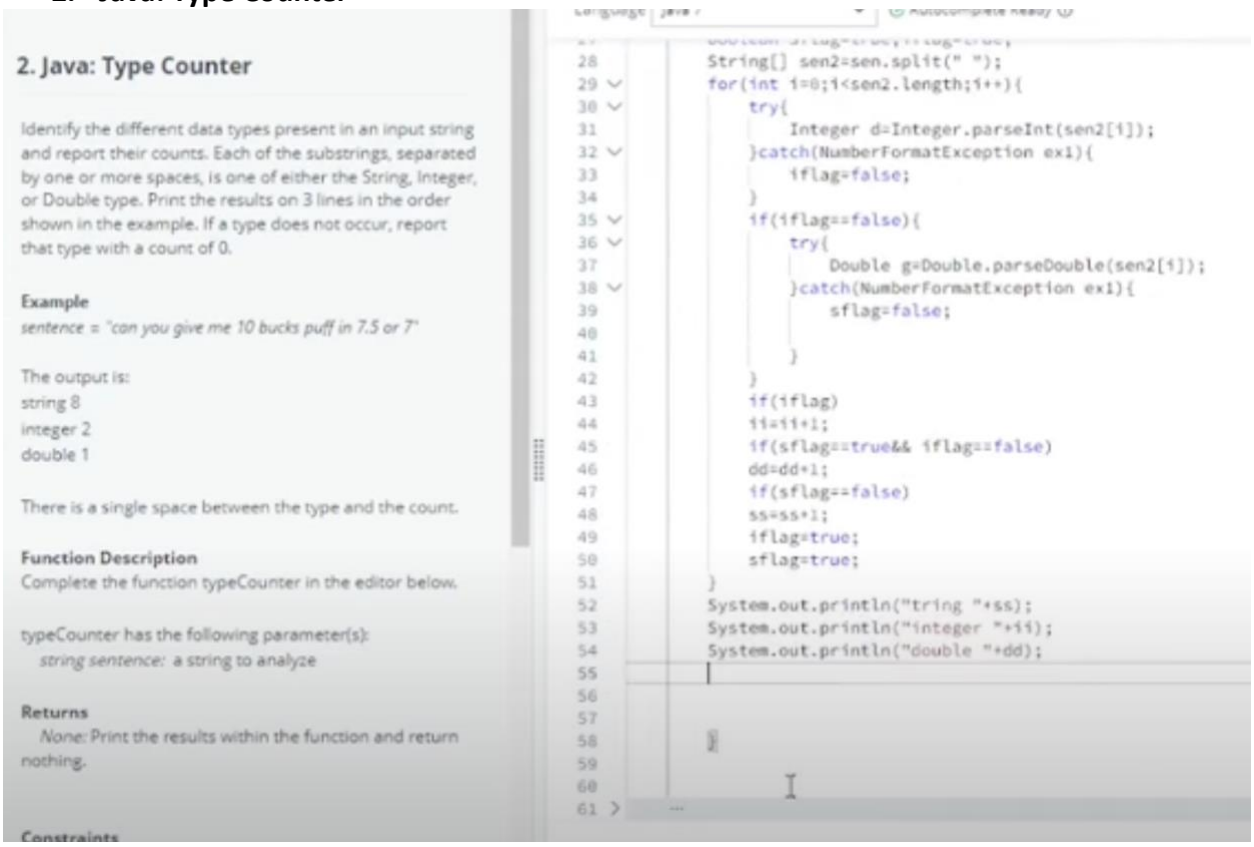
Language Java 8 — Autocomplete Ready

```
import java.util.*;
// Write your code here. DO NOT use an access modifier in your c
class Parser{
    static String isBalanced(String s)
    {

    }

}
class Solution {
```

2. **Java: Type Counter**



2. Java: Type Counter

Identify the different data types present in an input string and report their counts. Each of the substrings, separated by one or more spaces, is one of either the String, Integer, or Double type. Print the results on 3 lines in the order shown in the example. If a type does not occur, report that type with a count of 0.

Example
sentence = "can you give me 10 bucks puff in 7.5 or 7"

The output is:
string 8
integer 2
double 1

There is a single space between the type and the count.

Function Description
Complete the function typeCounter in the editor below.

typeCounter has the following parameter(s):
string sentence: a string to analyze

Returns
None: Print the results within the function and return nothing.

Constraints

```
String[] sen2=sen.split(" ");
for(int i=0;i<sen2.length;i++){
    try{
        Integer d=Integer.parseInt(sen2[i]);
    }catch(NumberFormatException ex1){
        iflag=false;
    }
    if(iflag==false){
        try{
            Double g=Double.parseDouble(sen2[i]);
        }catch(NumberFormatException ex1){
            sflag=false;

        }
    }
    if(iflag)
    ii=ii+1;
    if(sflag==true&& iflag==false)
    dd=dd+1;
    if(sflag==false)
    ss=ss+1;
    iflag=true;
    sflag=true;
}
System.out.println("tring "+ss);
System.out.println("integer "+ii);
System.out.println("double "+dd);

}
```

```java
// Sample Case 1
// Sample Input For Custom Testing
// string 2 3.54

// Sample Output
// string 1
// integer 1
// double 2

public static void typeCounter(String sentence) {
    // Input: give me 10 dollars
    // Output: string 3 integer 1 double 0
    //Input: string 2 3.54
    //Output: string 1 integer 1 double 2
    // Write your code here
    String[] words = sentence.split(" ");
    int stringCount = 0;
    int integerCount = 0;
    int doubleCount = 0;
    for(String word: words){
        if(word.matches("[a-z]+")){
            stringCount++;
        }else if(word.matches("[0-9]+")){
            integerCount++;
        }else if(word.matches("[0-9]+.[0-9]+")){
            doubleCount++;
        }
    }
    System.out.println("string " + stringCount);
    System.out.println("integer " + integerCount);
    System.out.println("double " + doubleCount);




}
public class Solution {
    Run | Debug
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));

        String sentence = bufferedReader.readLine();
```

Test.java > Test

```java
public class Test{
    Run | Debug
    public static void main(String[] args){
        int i = 010;
        int j = 07;
        System.out.println(i);
        System.out.println(j);
    }
}
```

PROBLEMS 40 OUTPUT DEBUG CONSOLE SQL CONSOLE

TERMINAL

```
(base) mirage@eduroam-198-96-32-218 IsAnagram %  /usr/bin/env /Users/mi
ibrary/Java/JavaVirtualMachines/openjdk-17.0.2/Contents/Home/bin/java -
owCodeDetailsInExceptionMessages -cp /Users/mirage/Library/ApplicationS
rt/Code/User/workspaceStorage/494cad81db698d7a9160ce25bc3edcb7/redhat.j
t_ws/IsAnagram_ebc01c5f/bin Test
8
7
```

What is a convariant return type?

- **The overriding method can have derived type as the return type instead of the base type**
- The overriding method can have base types as the return type instead of the derived type
- The return type is of the class type Convariant
- The return type is void

The map interface is implemented by which of the following classes:

- **HashMap**
- DynamicList
- Stack
- **TreeMap**

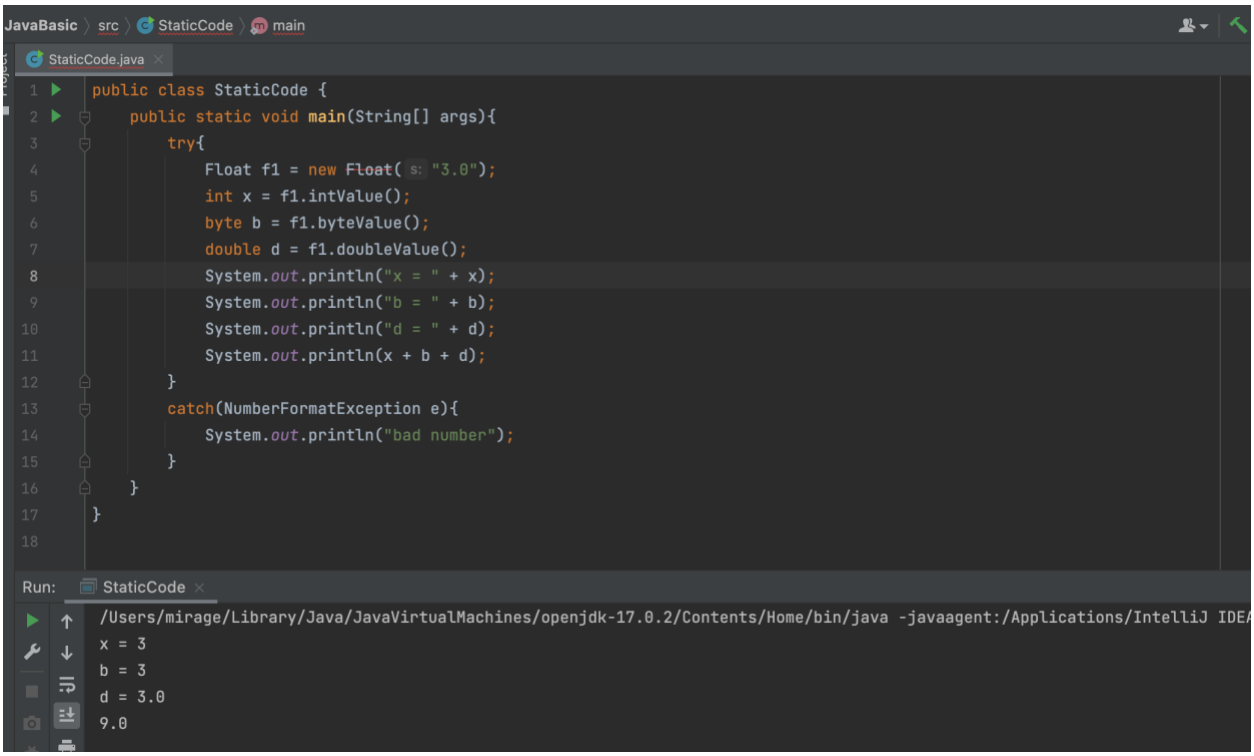
```java
// double 2

public static void typeCounter(String sentence) {
    // Input: give me 10 dollars
    // Output: string 3 integer 1 double 0
    //Input: string 2 3.54
    //Output: string 1 integer 1 double 2
    // Write your code here
    String[] words = sentence.split(" ");
    int stringCount = 0;
    int integerCount = 0;
    int doubleCount = 0;
    for(String word: words){
        if(word.matches("[a-z]+")){
            stringCount++;
        }else if(word.matches("[0-9]+")){
            integerCount++;
        }else if(word.matches("[0-9]+.[0-9]+")){
            doubleCount++;
        }
    }
    System.out.println("string " + stringCount);
    System.out.println("integer " + integerCount);
    System.out.println("double " + doubleCount);



}
```

Static Code


JavaBasic
src
StaticCode
main
StaticCode.java
public class StaticCode {
public static void main(String[] args){
try{
Float f1 = new Float(s: "3.0");
int x = f1.intValue();
byte b = f1.byteValue();
double d = f1.doubleValue();
System.out.println("x = " + x);
System.out.println("b = " + b);
System.out.println("d = " + d);
System.out.println(x + b + d);
}
catch(NumberFormatException e){
System.out.println("bad number");
}
}
}
Run:
StaticCode
/Users/mirage/Library/Java/JavaVirtualMachines/openjdk-17.0.2/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA
x = 3
b = 3
d = 3.0
9.0