3h 51m left

Help

All

1

2

3

4

Your task here is to implement a **Java** code based on the following specifications. Note that your code should match the specifications in a precise manner. Consider default visibility of classes, data fields and methods unless mentioned otherwise.

Specifications:

```
class definitions:
 class Implementation:
    method definitons:
      changeOccurrence(ArrayList<String> list,String a
          return type: ArrayList<String>
          visibility: public

     getIndex(ArrayList<String> list):
         return type: String
         visibility: public

    addAfter(ArrayList<String> list, String a, Strin
        return type: ArrayList<String>
        visibility: public
```

Task:

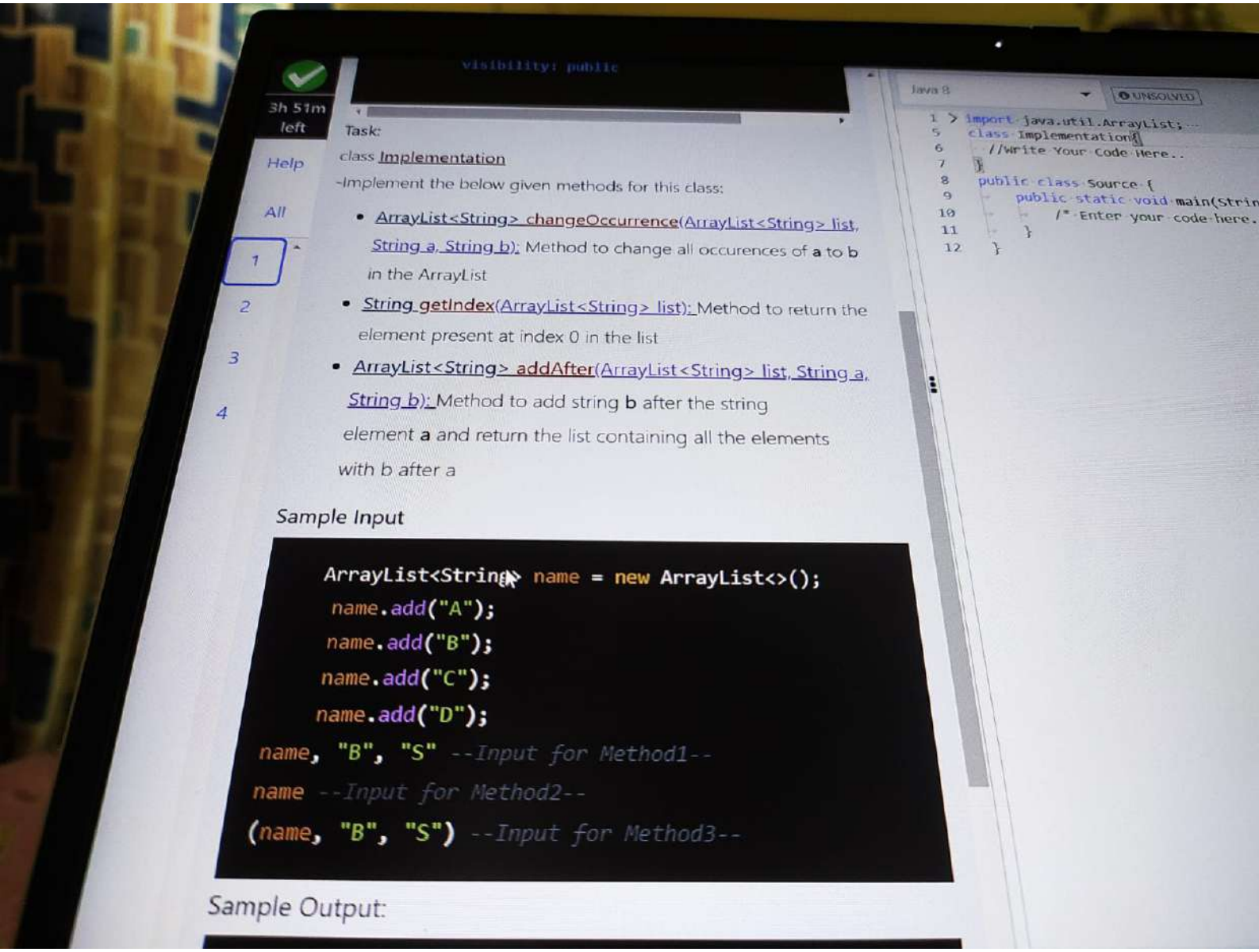
class Implementation

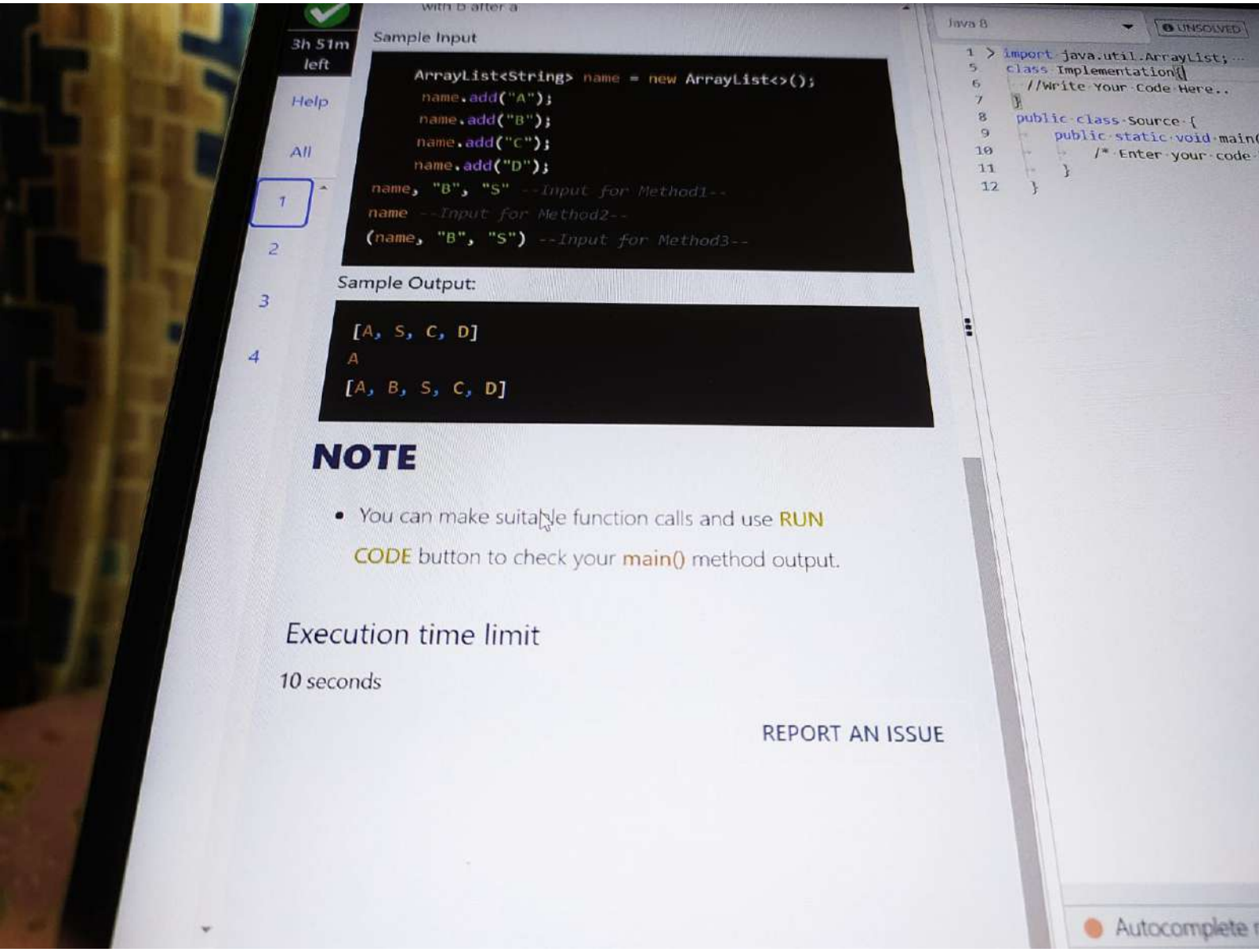
-Implement the below given methods for this class:

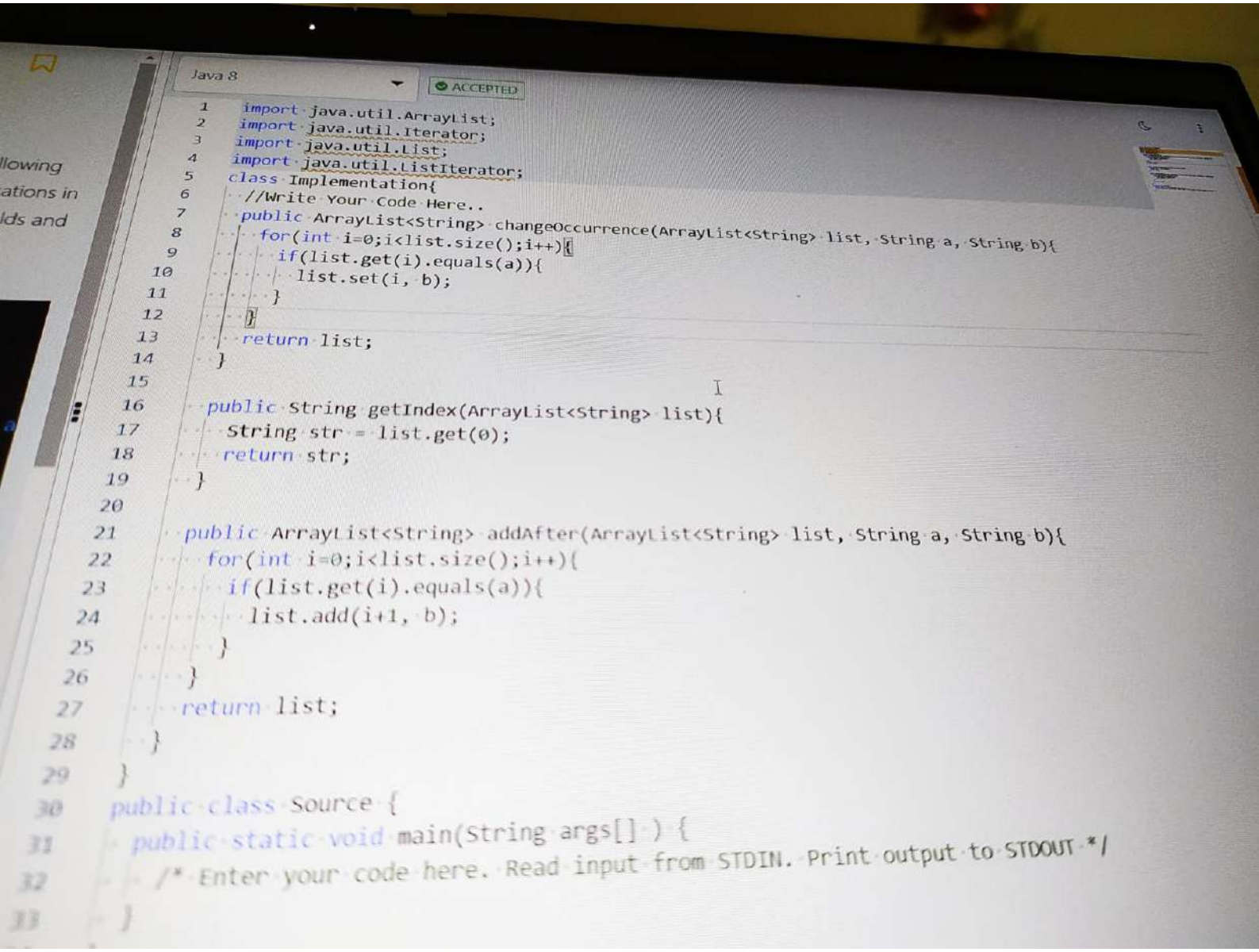
- ArrayList<String> changeOccurrence(ArrayList<String> list, String a, String b): Method to change all occurences of a to b in the ArrayList

UNSOLVED

```
1  import java.util.ArrayList;...
5  class Implementation{
6    //Write Your Code Here..
7  }
8  public class Source {
9      public static void main(Str
10         /* Enter your code here
11     }
12 }
```


visibility: public
3h 51m left
Help
All
1
2
3
4
Task:
class Implementation
-Implement the below given methods for this class:
ArrayList<String> changeOccurrence(ArrayList<String> list, String a, String b): Method to change all occurences of a to b in the ArrayList
String getIndex(ArrayList<String> list): Method to return the element present at index 0 in the list
ArrayList<String> addAfter(ArrayList<String> list, String a, String b): Method to add string b after the string element a and return the list containing all the elements with b after a
Sample Input
ArrayList<String> name = new ArrayList<>();
name.add("A");
name.add("B");
name.add("C");
name.add("D");
name, "B", "S" --Input for Method1--
name --Input for Method2--
(name, "B", "S") --Input for Method3--
Sample Output:
Java 8
UNSOLVED
1 import java.util.ArrayList;...
5 class Implementation{
6 //Write Your Code Here..
7 }
8 public class Source {
9 public static void main(Strin
10 /* Enter your code here.
11 }
12 }


with B after a
3h 51m left
Help
All
1
2
3
4
Sample Input
ArrayList<String> name = new ArrayList<>();
name.add("A");
name.add("B");
name.add("C");
name.add("D");
name, "B", "S" --Input for Method1--
name --Input for Method2--
(name, "B", "S") --Input for Method3--
Sample Output:
[A, S, C, D]
A
[A, B, S, C, D]
NOTE
You can make suitable function calls and use RUN CODE button to check your main() method output.
Execution time limit
10 seconds
REPORT AN ISSUE
Java 8
UNSOLVED
1 import java.util.ArrayList;...
5 class Implementation{
6 //Write Your Code Here..
7 }
8 public class Source {
9 public static void main(
10 /* Enter your code
11 }
12 }
Autocomplete



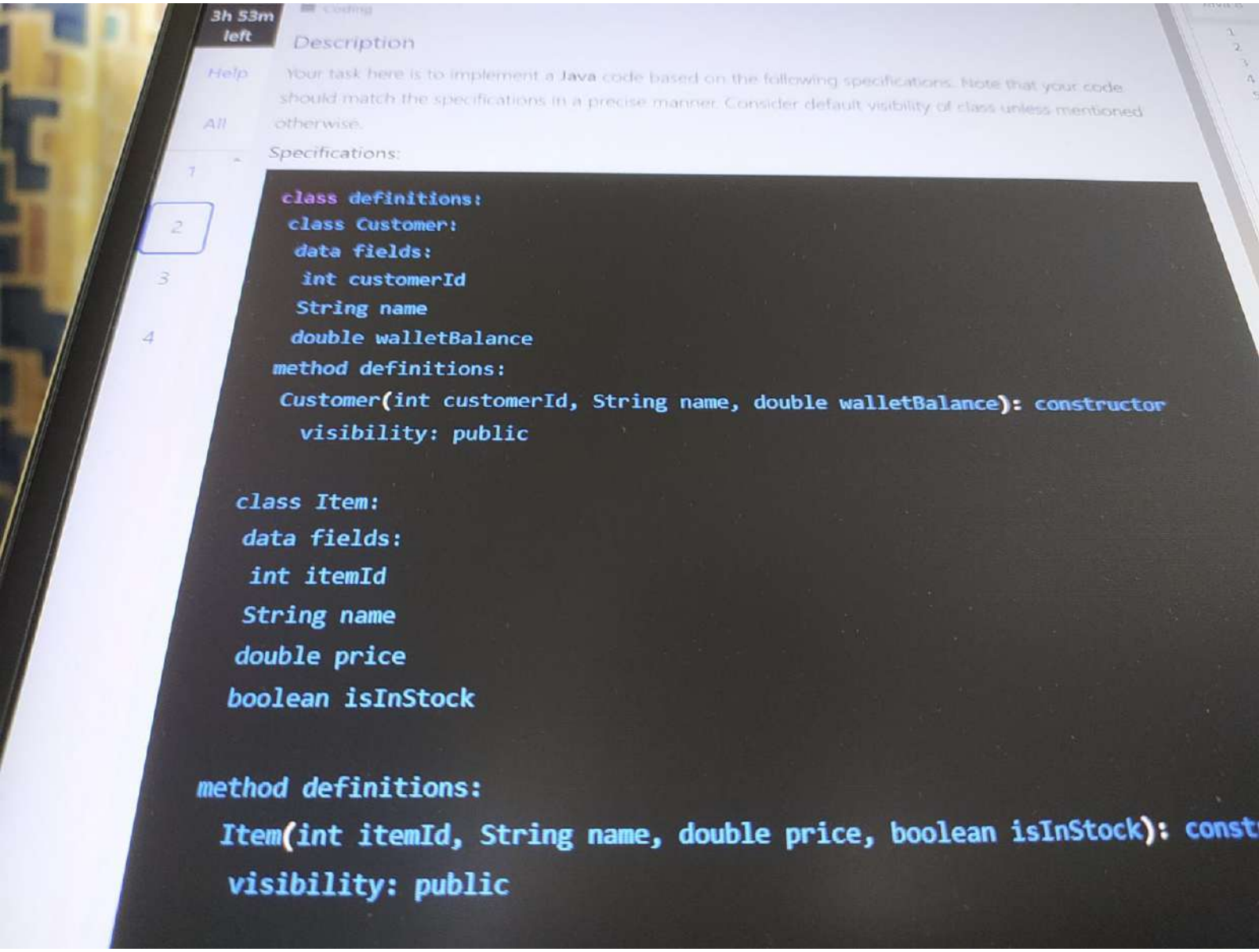
Java 8

ACCEPTED

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import java.util.ListIterator;
class Implementation{
    //Write Your Code Here..
    public ArrayList<String> changeOccurrence(ArrayList<String> list, String a, String b){
        for(int i=0;i<list.size();i++){
            if(list.get(i).equals(a)){
                list.set(i, b);
            }
        }
        return list;
    }

    public String getIndex(ArrayList<String> list){
        String str = list.get(0);
        return str;
    }

    public ArrayList<String> addAfter(ArrayList<String> list, String a, String b){
        for(int i=0;i<list.size();i++){
            if(list.get(i).equals(a)){
                list.add(i+1, b);
            }
        }
        return list;
    }
}
public class Source {
    public static void main(String args[] ) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    }
```


3h 53m left
Description
Help
All
Your task here is to implement a Java code based on the following specifications. Note that your code should match the specifications in a precise manner. Consider default visibility of class unless mentioned otherwise.
Specifications:
class definitions:
class Customer:
data fields:
int customerId
String name
double walletBalance
method definitions:
Customer(int customerId, String name, double walletBalance): constructor
visibility: public
class Item:
data fields:
int itemId
String name
double price
boolean isInStock
method definitions:
Item(int itemId, String name, double price, boolean isInStock): const
visibility: public

3h 52m left

Help

All

1

2

3

4

```
    Item(int itemId, String name, double price, boolean isInStock): constructor
     visibility: public

  class ShoppingWebsite:
   method definition:
      purchaseItem(Item i, Customer c) throws StockBalanceException:
         return type: String
         visibility: public
class StockBalanceException extends Exception:
 method definition:
    StockBalanceException(String message):
       visibility: public
```

Task:

-Implement class Customer according to the above specifications

-Implement class Item according to the above specifications

-Class ShoppingWebsite

Implement the below given method for this class:

-String purchaseItem(Item i, Customer c) throws StockBalanceException:

- Throw StockBalanceException when the item is out of stock with the message "item is out of stock".
- Throw StockBalanceException when customer wallet balance is not sufficient(Item price is greater than the wallet balance) with the message "customer wallet balance is not sufficient".
- If no exception found then return "Order Successful".

-Class StockBalanceException

- define custom exception class StockBalanceException by extending the Exception class.
- define a parameterized constructor with a String argument to pass the message to the super

3h 52m left

Help

All

- If no exception found then return "Order Successful".

-Class StockBalanceException

- define custom exception class **StockBalanceException** by extending the **Exception** class.
- define a parameterized constructor with a String argument to pass the message to the super class.

Sample Testcase

Input

```
Customer cusDet = new Customer(927392, "Steve" ,5000.0);
Item itemDet = new Item(27392, "T-Shirt", 800, true);
---------------------------------------------------------------
purchaseItem(itemDet, cusDet);
```

output

```
"Order Successful"
```

NOTE

- You can make suitable function calls and use the RUN CODE button to check your main() method output.

Execution time limit

10 seconds

REPORT AN IS

the following
specifications in
less mentioned

```
3   import java.text.*;
4   import java.math.*;
5   import java.util.regex.*;
6   class Customer {
7       //Write Your Code Here..
8       int customerId;
9       String name;
10      double walletBalance;
11
12      public Customer(int customerId, String name, double walletBalance){
13          this.customerId=customerId;
14          this.name=name;
15          this.walletBalance=walletBalance;
16      }
17  }
18
19  class Item {
20      int itemId;
21      String name;
22      double price;
23      boolean isInStock;
24      //Write Your Code Here..
25      public Item(int itemId, String name, double price, boolean isInStock){
26          this.itemId = itemId;
27          this.name=name;
28          this.price=price;
29          this.isInStock=isInStock;
30      }
31  }
32
33  class ShoppingWebsite {
34      //Write Your Code Here..
35      public String purchaseItem(Item i, Customer c) throws StockBalanceException{
36          if(!i.isInStock){
37              throw new StockBalanceException("item is out of stock");
```

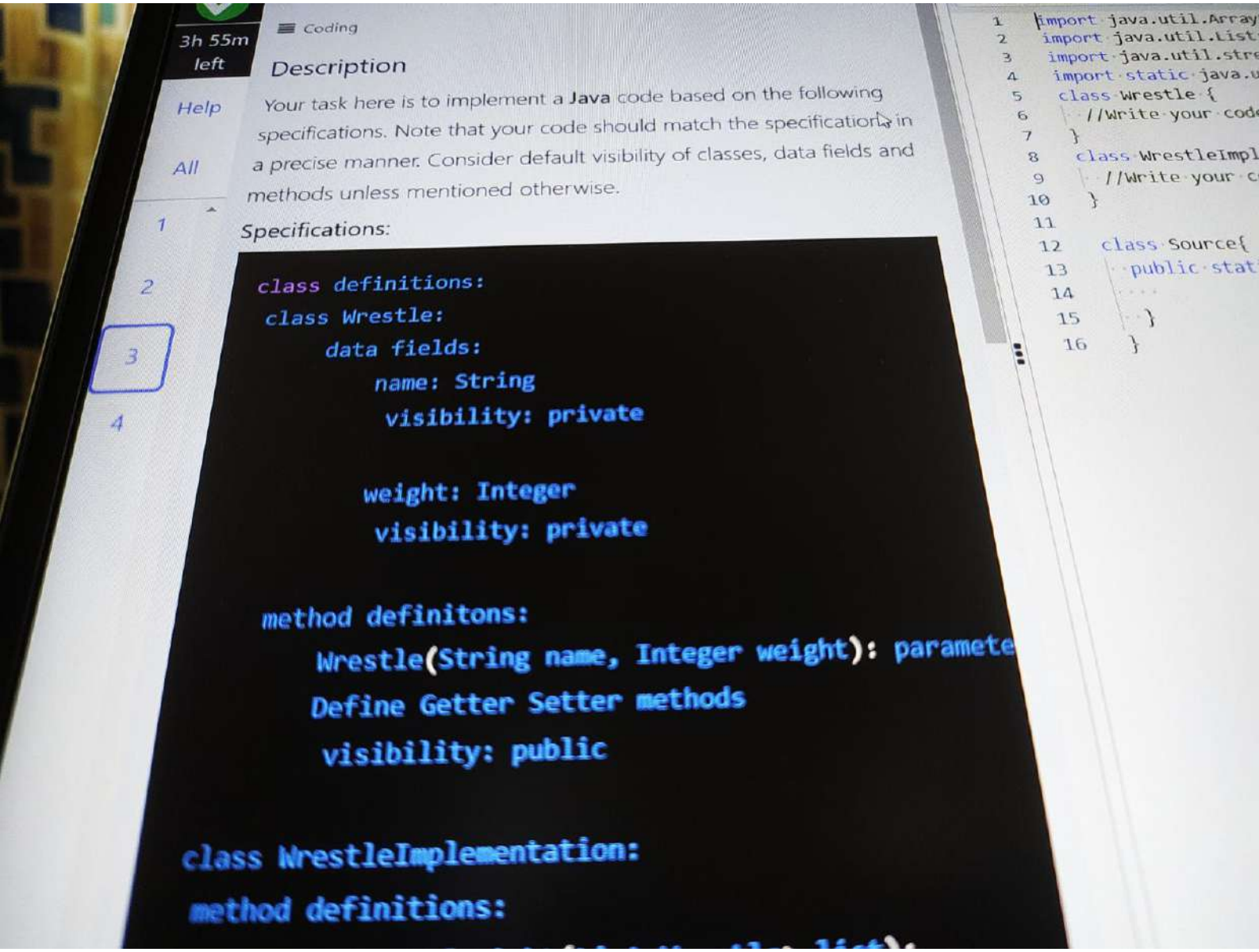
Autocomplete reconnecting...

```
36          if(!i.isInStock){
37              throw new StockBalanceException("item is out of stock");
38          }
39          else if(c.walletBalance < i.price){
40              throw new StockBalanceException("customer wallet balance is not sufficient");
41          }
42          else{
43              return "Order Successful";
44          }
45      }
46  }
47
48  class StockBalanceException extends Exception {
49      public StockBalanceException(String message){
50          super(message);
51      }
52      //Write Your Code Here..
53  }
54
55  public class Source {
56      public static void main(String args[] ) throws Exception {
57          /* Enter your code here. Read input from STDIN. Print output to STDOUT */
58      }
59  }
```

Autocomplete reconnecting...

RUN CODE

Test Results | Custom Input


3h 55m left
Coding
Description
Help
All
Your task here is to implement a Java code based on the following specifications. Note that your code should match the specifications in a precise manner. Consider default visibility of classes, data fields and methods unless mentioned otherwise.
Specifications:
1
2
3
4
class definitions:
class Wrestle:
data fields:
name: String
visibility: private
weight: Integer
visibility: private
method definitons:
Wrestle(String name, Integer weight): paramete
Define Getter Setter methods
visibility: public
class WrestleImplementation:
method definitions:
1 import java.util.Array
2 import java.util.List
3 import java.util.str
4 import static java.u
5 class Wrestle {
6 //Write your cod
7 }
8 class WrestleImpl
9 //Write your c
10 }
11
12 class Source{
13 public stat
14
15 }
16 }

3h 55m left

Help

All

1

2

3

4

```
class WrestleImplementation:
method definitions:
        getSumOfWeight(List<Wrestle> list):
            return type: int
            visibilty: public

        getWrestlerName(List<Wrestle> list):
            return type: List<String>
            visibilty: public

        getMaxWeight(List<Wrestle> list):
            return type: int
            visibilty: public
```

```
1  import java.util.Arrayl
2  import java.util.List;
3  import java.util.strea
4  import static java.ut
5  class Wrestle {
6      //Write your code
7  }
8  class WrestleImple
9      //Write your co
10 }
11
12 class Source{
13     public stati
14
15     }
16 }
```

Task:

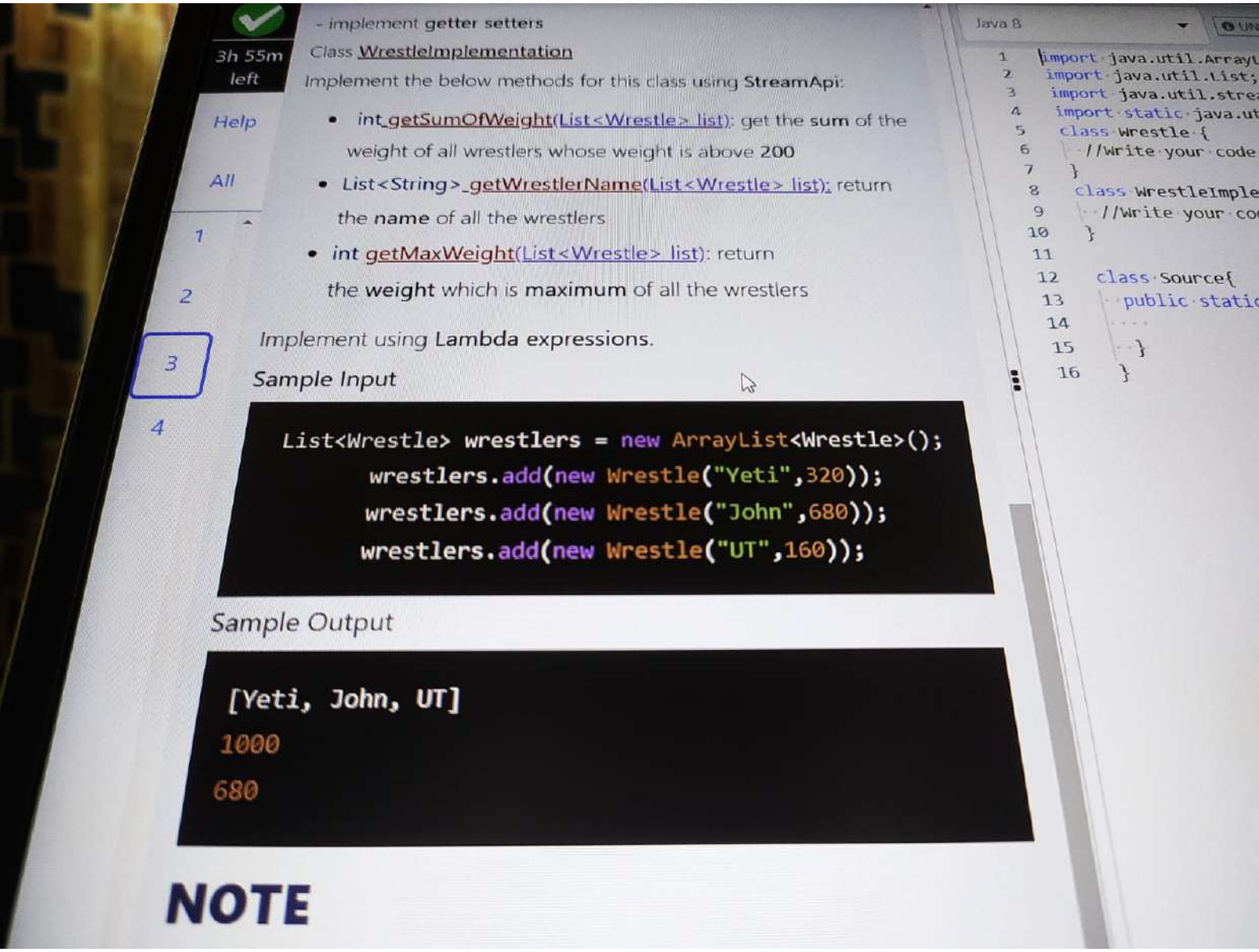
Class Wrestle

- define the String variable name
- define the Integer variable weight
- define a constructor
- implement getter setters

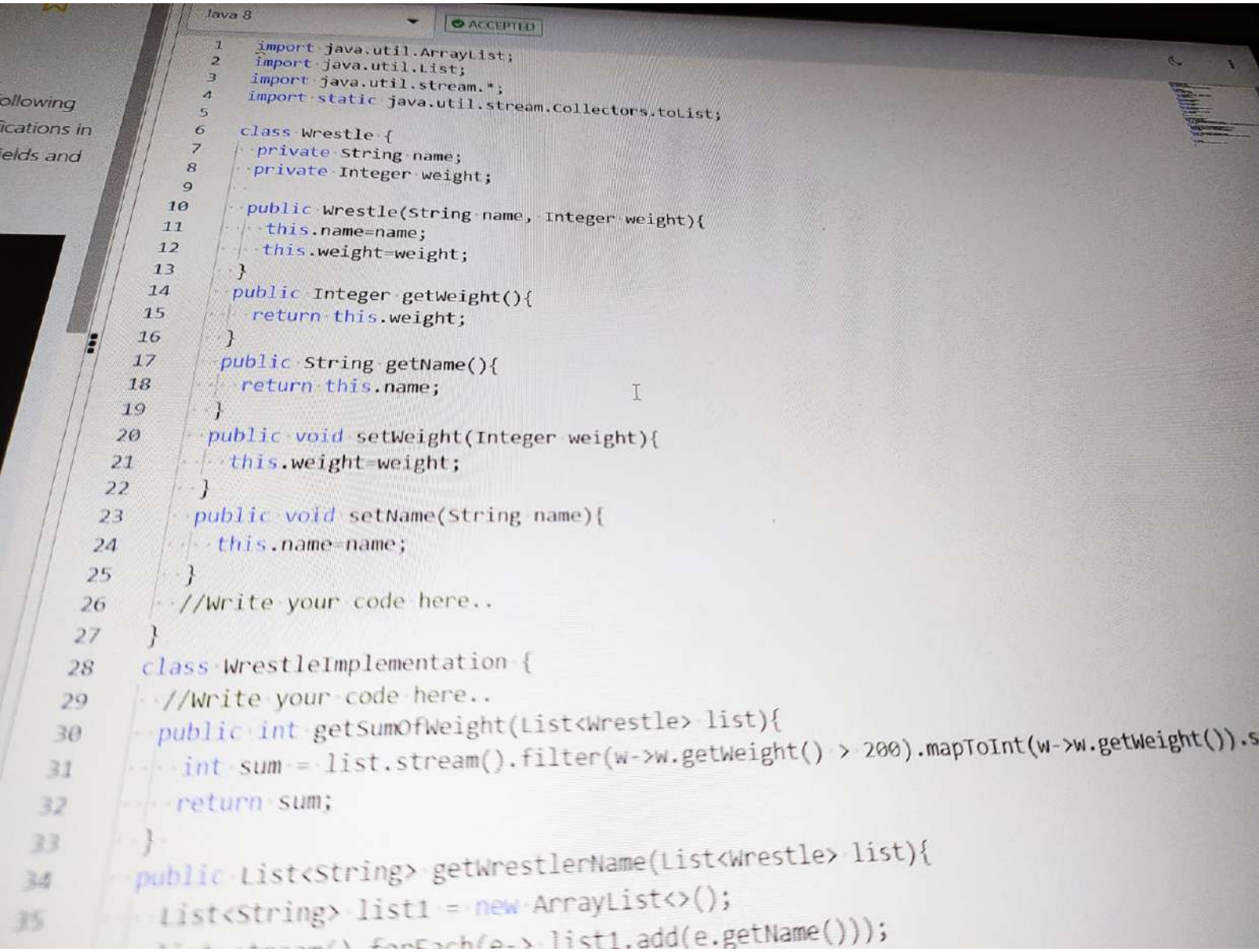
Class WrestleImplementation

Implement the below methods for this class using StreamApi:

- int getSumOfWeight(List<Wrestle> list): get the sum of the weight of all wrestlers whose weight is above 200


- implement getter setters
3h 55m left
Class WrestleImplementation
Implement the below methods for this class using StreamApi:
Help
All
int getSumOfWeight(List<Wrestle> list): get the sum of the weight of all wrestlers whose weight is above 200
List<String> getWrestlerName(List<Wrestle> list): return the name of all the wrestlers
int getMaxWeight(List<Wrestle> list): return the weight which is maximum of all the wrestlers
Implement using Lambda expressions.
Sample Input
List<Wrestle> wrestlers = new ArrayList<Wrestle>();
wrestlers.add(new Wrestle("Yeti",320));
wrestlers.add(new Wrestle("John",680));
wrestlers.add(new Wrestle("UT",160));
Sample Output
[Yeti, John, UT]
1000
680
Java 8
import java.util.Arrayl
import java.util.List;
import java.util.stre
import static java.ut
class Wrestle {
//Write your code
}
class WrestleImple
//Write your co
}
class Source{
public static
}
}

NOTE



Java 8 ACCEPTED

```
import java.util.ArrayList;
import java.util.List;
import java.util.stream.*;
import static java.util.stream.Collectors.toList;

class Wrestle {
    private String name;
    private Integer weight;

    public Wrestle(String name, Integer weight){
        this.name=name;
        this.weight=weight;
    }
    public Integer getWeight(){
        return this.weight;
    }
    public String getName(){
        return this.name;
    }
    public void setWeight(Integer weight){
        this.weight=weight;
    }
    public void setName(String name){
        this.name=name;
    }
    //Write your code here..
}
class WrestleImplementation {
    //Write your code here..
    public int getSumOfWeight(List<Wrestle> list){
        int sum = list.stream().filter(w->w.getWeight() > 200).mapToInt(w->w.getWeight()).s
        return sum;
    }
    public List<String> getWrestlerName(List<Wrestle> list){
        List<String> list1 = new ArrayList<>();
```

following
cifications in
fields and

```
    public String getName(){
        return this.name;
    }
    public void setWeight(Integer weight){
        this.weight=weight;
    }
    public void setName(String name){
        this.name=name;
    }
    //Write your code here..
}
class WrestleImplementation {
    //Write your code here..
    public int getSumOfWeight(List<Wrestle> list){
        int sum = list.stream().filter(w->w.getWeight() > 200).mapToInt(w->w.getWeight()).sum();
        return sum;
    }
    public List<String> getWrestlerName(List<Wrestle> list){
        List<String> list1 = new ArrayList<>();
        list.stream().forEach(e-> list1.add(e.getName()));
        return list1;
    }
    public int getMaxWeight(List<Wrestle> list){
        int max = list.stream().mapToInt(w->w.getWeight()).max().getAsInt();
        return max;
    }
}

class Source{
    public static void main(String[] args){
    ....

    }
}
```

Autocomplete reconnecting...

RUN CO

3h 58m left

Help

All

1

2

3

4

Coding

Description

Danish has opened a seed bags selling shop. He wants to arrange the bags in the order in which a bag expires first. These bags have a twelve-digit code where:

- the first four characters are Batch Number.
- The next 8 digits represent the seed expiry date in **YYYYMMDD** format.

Generate a **Java** code to extract the batch number and expiry date from package code and validate it.

Validations:

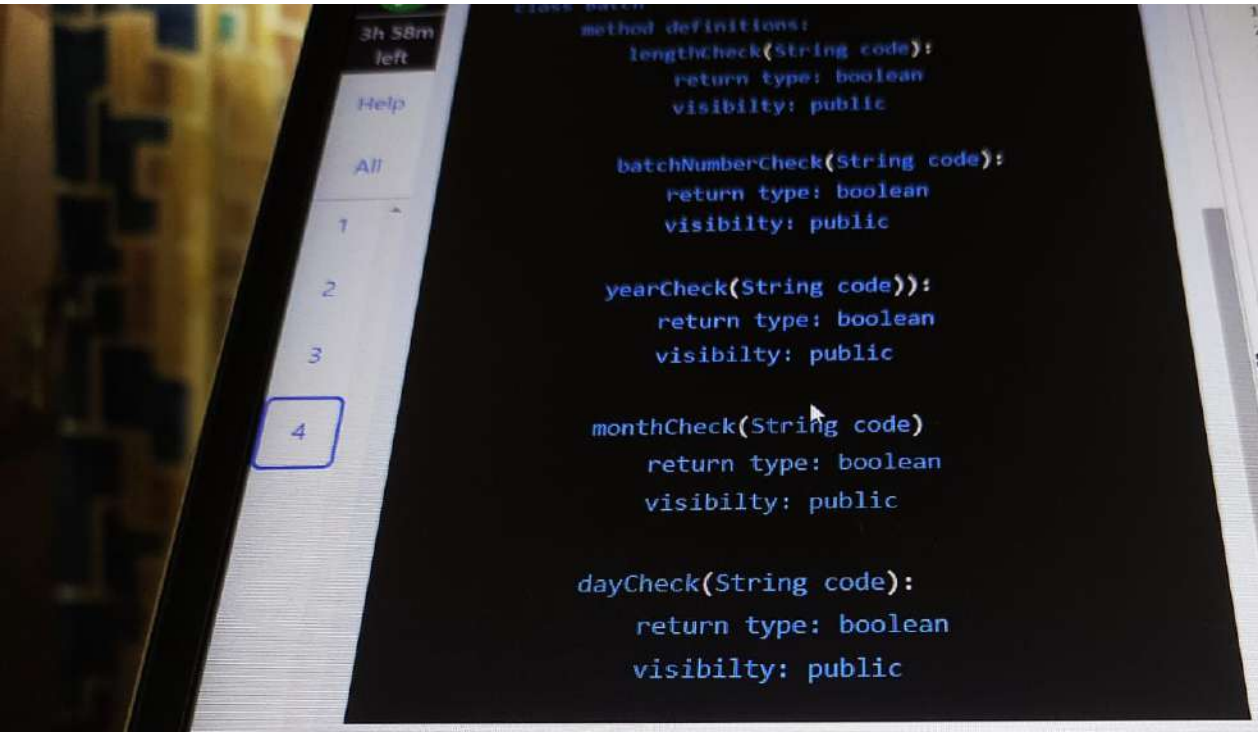
- Batch number is valid only if the first, second and fourth characters are alphabets in uppercase and the third character is a number.
- Year is valid only if it is between 2015 and 2020, both inclusive.
- Month is valid only if it is in between 1 and 12 (both inclusive).
- Day is valid only if it is between 1 and 31 (both inclusive).

Assumption:

- All characters in the input string will be in UPPER CASE.

```
class definitions:
```

```
1  import java.io.*;
2  import java.util.*;
3  import java.text.SimpleDater
4  import java.util.Date;
5
6  class Batch{
7
8      public boolean lengt
9          //RETURN true if
10     }
11
12     public boolean bat
13         //RETURN true
14     }
15
16     public boolean
17         //Check if
18     }
19
20     public boole
21         //Check
22     }
23
24     public bo
25         //Che
26     }
27
28 }
29
30 class Sou
31     publi
32
33     }
34 }
```



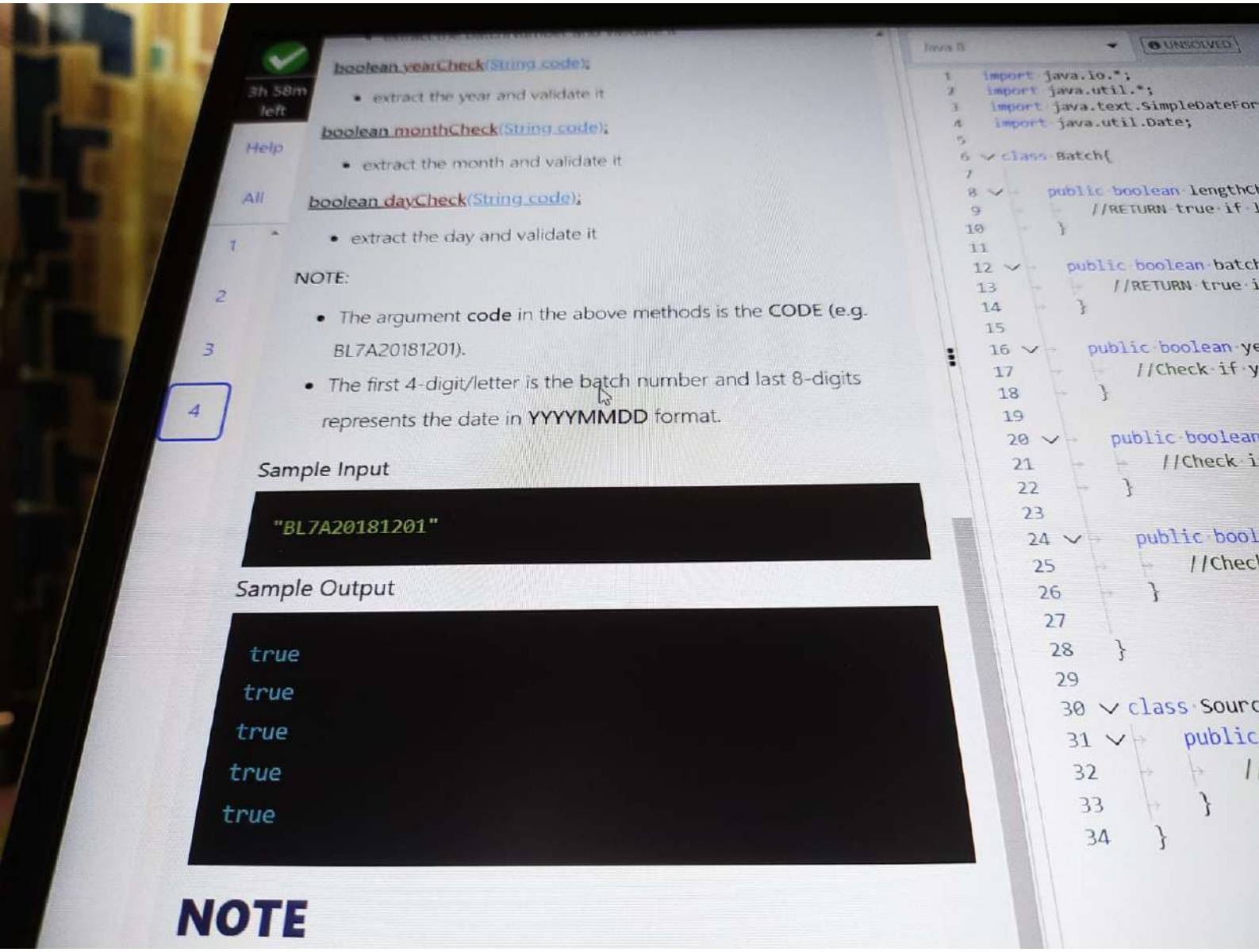
Methods to be Implemented:

boolean lengthCheck(String code);

- Compute the length of String **code** and return true if length is 12 else return false.

boolean batchNumberCheck(String code);

- extract the batchNumber and validate it


3h 58m left
Help
All
1
2
3
4
boolean yearCheck(String code);
extract the year and validate it
boolean monthCheck(String code);
extract the month and validate it
boolean dayCheck(String code);
extract the day and validate it
NOTE:
The argument code in the above methods is the CODE (e.g. BL7A20181201).
The first 4-digit/letter is the batch number and last 8-digits represents the date in YYYYMMDD format.
Sample Input
"BL7A20181201"
Sample Output
true
true
true
true
true
NOTE
Java 8
UNSOLVED
import java.io.*;
import java.util.*;
import java.text.SimpleDateFor
import java.util.Date;
class Batch{
public boolean lengthCh
//RETURN true if
public boolean batch
//RETURN true i
public boolean ye
//Check if y
public boolean
//Check i
public bool
//Chec
class Sourc
public
/
}

wants to arrange
hese bags have a

expiry date

rth

haracter is

clusive.

sive).

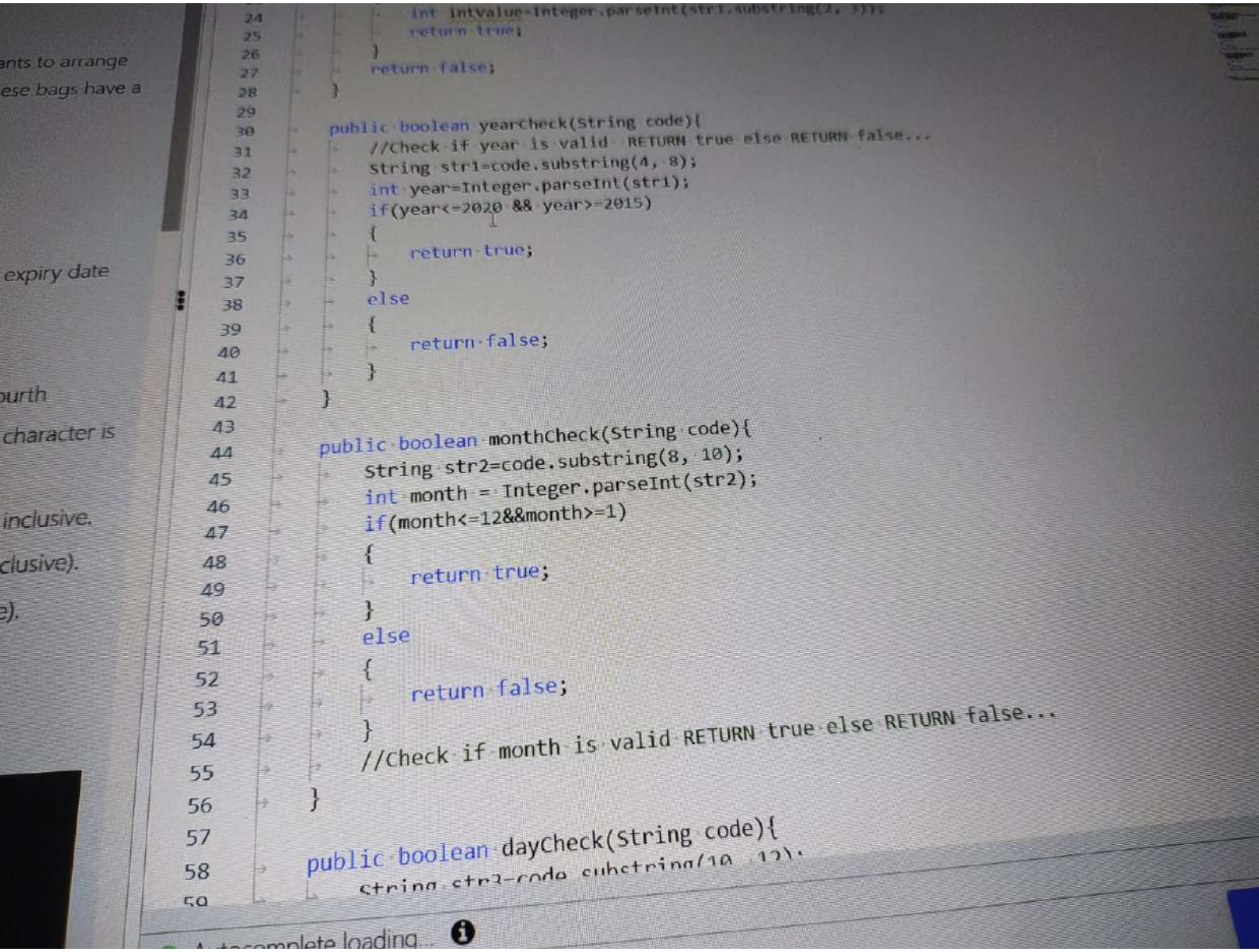
```java
import java.io.*;
import java.util.*;
import java.text.SimpleDateFormat;
import java.util.Date;

class Batch{

    public boolean lengthCheck(String code){
        if(code.length()==12)
            return true;
            return false;
    }

    public boolean batchNumberCheck(String code){
        String str1=code.substring(0, 4);
        String check1=str1.substring(0, 1);
        String check2=str1.substring(1, 2);
        String check3=str1.substring(3,4);
        char x=check1.charAt(0);
        char y=check2.charAt(0);
        char z=check3.charAt(0);
        if(Character.isUpperCase(x)&& Character.isUpperCase(y) && Character.isUpperCase(z) )
        {
            int intValue=Integer.parseInt(str1.substring(2, 3));
            return true;
        }
        return false;
    }

    public boolean yearCheck(String code){
        //Check if year is valid  RETURN true else RETURN false...
        String str1=code.substring(4, 8);
        int year=Integer.parseInt(str1);
        if(year<=2020 && year>=2015)
        {
            return true;
        }
```



```
        int intValue=Integer.parseInt(str1.substring(2, 3));
        return true;
    }
    return false;
}

public boolean yearcheck(String code){
    //Check if year is valid  RETURN true else RETURN false...
    String str1=code.substring(4, 8);
    int year=Integer.parseInt(str1);
    if(year<=2020 && year>=2015)
    {
        return true;
    }
    else
    {
        return false;
    }
}

public boolean monthCheck(String code){
    String str2=code.substring(8, 10);
    int month = Integer.parseInt(str2);
    if(month<=12&&month>=1)
    {
        return true;
    }
    else
    {
        return false;
    }
    //Check if month is valid RETURN true else RETURN false...
}

public boolean dayCheck(String code){
    String str3=code.substring(10, 12);
```

Autocomplete loading...

op. He wants to arrange
s first. These bags have a

ber
piry date

ber and expiry date

d and fourth
e third character is

), both inclusive.
both inclusive).
nclusive).

CASE.

```
    public boolean monthCheck(String code){
        String str2=code.substring(8, 10);
        int month = Integer.parseInt(str2);
        if(month<=12&&month>=1)
        {
            return true;
        }
        else
        {
            return false;
        }
        //Check if month is valid RETURN true else RETURN false...
    }

    public boolean dayCheck(String code){
        String str3=code.substring(10, 12);
        int date=Integer.parseInt(str3);
        if(date<=31&&date>=1)
        {
            return true;
        }
        else
        {
            return false;
        }
        //Check if day is valid RETURN true or else RETURN false...
    }

}

class Source{
    public static void main(String[] args) {
        //Don't touch this or else your program might not run...
    }
}
```

Autocomplete reconnecting...