

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

Class Source

Implement the below methods for this class:

- `main()`:

- create objects for `Age` and `ExceptionCheck`
- call the required methods

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

```

35  .  String str1;
36  .  try{
37  .  .  if(age.length()>2 || Integer.valueOf(age)<21){
38  .  .  .  throw new IllegalAgeException("Invalid age detail or drinking age");
39  .  .  }
40  .  .  else if(age.length()<3 && Integer.valueOf(age)>=21){
41  .  .  .  str1="legal";
42  .  .  .  return str1;
43  .  .  }
44  .  }
45  .  catch(IllegalAgeException iae){
46  .  .  str1="illegal";
47  .  .  return str1;
48  .  }
49  .  return str1;
50  .  }
51
52
53  }
54
55 //User defined exception class
56 class IllegalAgeException extends Exception {
57     public IllegalAgeException(String s){
58         super(s);
59     }
60 }
61
62 // otherwise solution won't be accepted
63 public class Source {
64     public static void main(String args[] ) throws Exception {
65         /* Enter your code here. Read input from STDIN. Print output to STDOUT */
66         //
67     }
68 }

```

● Autocomplete reconnecting... 

Ln 10, Col 1 Java

Test Results

Custom Input

RUN CODE

SUBMIT

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
```

```
class Age{
    String drink;
}

class ExceptionCheck{
    public String validateIntAgeToDrink(Age a, int age){
        String str="";
        try{
            if(age<21){
                throw new IllegalAgeException("Illegal drinking age");
            }
            else if(age>=21){
                str="legal";
            }
            return str;
        }
        catch(IllegalAgeExceptioniae){
            str="illegal";
            return str;
        }
        return str;
    }

    public String validateStringAgeToDrink(Age a, String age){
        String str1="";
        try{
            if(age.length()>2 || Integer.valueOf(age)<21){
                throw new IllegalAgeException("Invalid age detail or drinking age");
            }
            else if(age.length()<3 && Integer.valueOf(age)>=21){
                str1="legal";
            }
        }
    }
}
```

Autocomplete reconnecting... 

100% java

Test Results

Custom Input

RUN CODE

SUBMIT

14 }

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

Class IllegalAgeException

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

All

1

Class Source

Implement the below methods for this class:

- main():

- create objects for **Age** and **ExceptionCheck**
- call the required methods

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](#)





32m 9s
left

Help

All

1

2

3

4

- If the variable `age` is greater than or equal to 21, assign the string "legal" to `a.drink`
- catch the thrown exception and assign the string "illegal" to `a.drink`
- return default message if an exception is thrown, else return `a.drink`.

- `String validateStringAgeToDrink(Age a, String age):`

- This method receives the `age` as `String`, validates it and sets the value of `drink` variable available in `Age a` as below
- If the length of the variable `age` is greater than 2 or numerical value of `age` is less than 21, throw a user-defined exception `IllegalAgeException("Invalid age detail or drinking age")`
- If the length of variable `age` is less than 3 and numerical value of `age` is greater than or equal to 21, assign the string "legal" to `a.drink`
- catch the thrown exception and assign the string "illegal" to `a.drink`
- return default message if an exception is thrown, else return `a.drink`.

Java 8

```
14 }  
15  
16 class  
17 public  
18  
19  
20 ... i  
21  
22 ... J  
23 ... e  
24 ... str  
25 ... ret  
26  
27  
28  
29  
30  
31  
32  
33 }  
34 pu  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46
```

Class IllegalAgeException

- define custom exception class `IllegalAgeException` by extending the `Exception` class

Autoco

Test Re



32m 16s
left

Help

All

1

2

3

4

Tasks

Class Age

- define the String variable **drink**

Class ExceptionCheck

Implement the below methods for this class:

- String validateIntAgeToDrink(Age a, int age) :

- This method validates the int age argument and accordingly sets the value of drink variable available in Age a as below
- If the variable age is less than 21, throw a user-defined exception **IllegalAgeException("Illegal drinking age")**
- If the variable age is greater than or equal to 21, **assign the string "legal" to a.drink**
- **catch the thrown exception and assign the string "illegal" to a.drink**
- **return default message if an exception is thrown, else return a.drink**



2. Tell your Age!



≡ Coding

Description

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 unless mentioned otherwise.

Specifications

```
class definitions:  
    class Age  
        class variables:  
            String drink  
  
    class ExceptionCheck  
        method definitions:  
            String validateIntAgeToDrink()(Age  
                visibility: public  
  
            String validateStringAgeToDrink(Age  
                visibility: public  
  
    class IllegalAgeException  
        method definitions:  
            IllegalAgeException(String s)  
                visibility: public  
  
class Source
```