CORE JAVA PROJECT: STUDENT GRADE CALCULATION

Project Design:

You need to implement a Student grade calculation system in Java. Here, data is available in an array of objects.

If the given object has any data errors, then, the program has to return appropriate error messages. On the other hand, if given object has no validation errors, then, we need to find the grade and print the same.

Packages Used:

Package 1: com.mile1.**bean**– All bean classes are defined.

Package 2: com.mile1.**exception** –All the used defined exceptions are defined.

Package 3: com.mile1.**service**–All functional classes are defined.

Package 4: com.mile1.**main** -- A class with main function is defined.

Package 1: **com.mile1.bean**

Description of the class:

Class Student	
Variables	Description

String name;
int marks[];
 	
// **note: need to allocate an int array of size 3**
	Instance variables

Constructors	// To be Auto generated

public Student() {
 	
}
public Student(String name, int[] marks)
{
// **do the initialization**
}	

Methods	// To be Auto generated
Provide public Getters And public Setters for all instance variables	

Package 2: **com.mile1.exception**

Description of the class:

All the classes in this package should ***extend*** the Exception class.

Class	**Method**	Description

NullMarksArrayException	public String toString()
	{
	}
		Return "NullMarksArrayException occurred" inside the toString() function.

NullNameException	public String toString()
	{
	}
		Return "NullNameException occurred" inside the toString() function.

NullStudentException	public String toString()
	{
	}
		Return "NullStudentException occurred" inside the toString() function.

Package 3: **com.mile1.service**

Description of the class:

Class **StudentReport**	
Method1	Description

public String **findGrade** (Student studentObject)
{
// write code here
}
	Only valid objects are passed to this function. So, just concentrate on the logic part.
	Get the marks from *studentObject*.
	if (any one of the marks is less than 35)
	then return the "F" grade;
	else
	{
	Find the **sum** of all the marks.
	if sum is less than or equal to 150)
	then return "D" grade
	else if sum is greater than 150 and less than or equal to 200, Then return "C" grade.
	else if sum is greater than 200 and less than or equal to 250, then return "B" grade
	else if sum is greater than 250 and less than or equal to 300, then return "A" grade
	}

Method2	Description

public String validate
(Student studentObject)
throws NullStudentException,
NullNameException,
NullMarksArrayException
{
// write code here
}
	Check whether there is any null data in the given object.
	If given Object itself is null, then,
	{
	Throw the NullStudentException.
	}
	Else we do the following:
	{
	1) Check whether name is null. If so, throw the NullNameException.
	2) Check whether marks array is null. If so, throw the NullMarksArrayException
	If NullNameException and NullMarksArrayException not thrown, all data is valid. We need to call the **findGrade** function that is in the same class.
	Return the message returned by this function.
	}

Package3 com.mile1.service

Description of the class:

Class StudentService	
Methods	Description

public int findNumberOfNullMarks
(Student data[])
{
// write code here
}
	This function is used to count the number of objects where the marks array is null.
	Note: If you are not careful, you will get NullPointerException in this function.

public int findNumberOfNullNames
(Student data [])
{
// write code here
}
	This function is used to count the number of objects where the name is null.
	Note: If you are not careful, you will get NullPointerException in this function.

public int
findNumberOfNullObjects
(Student data [])
{
// write code here
}
	This function is used to count the number of objects where the given object itself is null.
	Hint:
	To Check whether an object is null, use (obj== null);
	Note: If you are not careful, you will get NullPointerException in this function.

Package 4   com.mile1.main

Description of the class:

Class StudentMain

Variables

static   Student **data**[] = new Student[4];

```
// use a static block to initialize the objects
static{
        for (int i = 0; i < data.length; i++)      data [i]= new Student();

        data [0] = new Student("Sekar", new int[]{35,35,35});
        data [1] = new Student(null, new int[]{11,22,33});
        data [2] = null;
        data [3] = new Student("Manoj", null);
}
```

MAIN METHOD:

This main function used to call the various functions defined in StudentReport class and StudentService class.

Create an Object for StudentReport and do the following.

1) Call the validate function for all the objects available data [] array.
2) If any exception occurs display, the details of the exception occurred.
3) If no exception raised, then, print the result returned by the ***validate*** *function*.

Create an Object for StudentService. Using this object, do the following:

Call the findNumberOfNullMarks (data) function and print the result.

Call the findNumberOfNullNames (data) function and print the result.

Call the findNumberOfNullObjects (data) function and print the result.

Sample main method looks like this:

```
static   Student data [] = new Student [4];

static {

        for (int i = 0; i < s.length; i++)         data [i] = new Student ();

        data [0] = new Student ("Sekar", new int [] {35, 35, 35});

        data [1] = new Student(null,new int[]{11,22,33});

        data [2] = null;

        data [3] = new Student ("Manoj", null);

}


public static void main (String a []) {

        StudentService studentService = new StudentService ();

        StudentReport studentReport = new StudentReport ();

        System.out.println (" Grades Calculation: ");

        String x = null;

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

                try {x = studentReport.validate (data [i]) ;}

                catch (NullNameException e) {x="NullNameException occurred“; }

                catch (NullMarksArrayException e) {x="NullMarksArrayException occurred“;}

                catch (NullStudentException e) { x="NullStudentException occurred “; }

                System.out.println ("GRADE="+x);

        }

        System.out.println ("Number of Objects with Marks array as null ="
                + studentService.findNumberOfNullMarks (data));

        System.out.println (“Number of Objects with Name as null="

                + studentService.findNumberOfNullNames(data));

        System.out.println (“Number of Objects that are entirely null=“

                + studentService.findNumberOfNullObjects(data));

 }
```

SAMPLE OUTPUT:

```
Grades Calculation:

GRADE= D

GRADE= NullNameException occurred

GRADE= NullStudentException occurred

GRADE= NullMarksException occurred

Number of Objects with Marks array as null =1

Number of Objects with Name as null=1

Number of Objects that are entirely null=1
```

A NOTE ON TEST CASES:

Your solution is tested with the following set of test cases.

GRADE CALCULATION FOR VALID OBJECT:

TC1 -- Calculate the grade for **valid** objects – Check for A grade computation.

TC2 -- Calculate the grade for **valid** objects – Check for D grade computation.

TC3 -- Calculate the grade for **valid** objects – Check for F grade computation.

THROW ERROR MESSAGE FOR INVALID OBJECT:

Check whether the validate function handles the following situations.

TC4 -- If the Object is null, throw NullStudentException ().

TC5-- If the Name is null, throw NullNameException ().

TC6 -- If the Marks array is null, throw NullMarksArrayException ().

COUNTING THE NULL:

TC7 – Test findNumberOfNullName function.

TC8 – Test findNumberOfNullObjects function.

TC9 -- Test findNumberOfNullMarks function.

SAMPLE INPUT2:

```
data [0] = new Student ("A1", new int [ ] {72, 73, 74});
data [1] = new Student ("B1", new int [ ] {75, 76, 77});
data [2] = new Student ("C1", new int[ ] {99, 99, 99});
data [3] = new Student ("C3", new int[ ] {100, 100, 99});
data [4] = new Student ("B2", new int[ ] {13, 88, 13});
data [5] = new Student ("C3", new int[ ] {14, 14, 99});
data [6] = new Student ("A2", new int[ ] {77, 55, 12});
data [7] = new Student (“A5", new int[ ] {13, 88, 13});
```

SAMPLE OUTPUT2:

```
Grades Calculation:
GRADE= B
GRADE= B
GRADE= A
GRADE= A
GRADE= F
GRADE= F
GRADE= F
GRADE= F
Number of Objects with Marks array as null =0
Number of Objects with Name as null=0
Number of Objects that are entirely null=0
```

//-----------------------------***************************************--//