Section 1: general information

Specifications

Description

The method calculates the letter grade based on several factors.

Javadoc

    public static enum LetterGrade { GRADEA, GRADEAPLUS,GRADEB,GRADEBPLUS,GRADEC, GRADECPLUS,GRADED, GRADEDPLUS, GRADEF , ERROR};

    /\*\*

     \* Determine the letter grade in the class based on points and perfect attendance

     \* we use the regular ranges: F less than 60, D less than 70, C less than 80, B less than 90, A up to 100.

     \* perfect attendance adds a plus to the letter grade except for F

     \* points less than 0 is an error

     \*

     \* @param points How many points the student has scored

     \* @param perfect Whether the student has perfect attendance

     \*

     \* @return

     \* A - at least 90 points

     \* B - at least 80 points

     \* C - at least 70 points

     \* D - at least 60 points

     \* F - zero up to 60 points

     \* PLUS added for perfect attendance

     \* ERROR - points less than 0

     \*/

Note: do not delete anything from this document, only add to it. Do not re-order things in this document.

-------------------------------------------end of section 1 -----------------------------------------------------

Section 2: chapter-specific data

Natural ranges for Equivalence partitions

Points

| Long.MIN\_VALUE --- long.MAX\_VALUE |

Perfect:

|True | false|

Lettergrade

| GRADEA | GRADEAPLUS | GRADEB | GRADEBPLUS | GRADEC | GRADECPLUS | GRADED | GRADEDPLUS | GRADEF | ERROR |

Specifications-based Ranges

|  |  |
| --- | --- |
| **Parameter** | **Natural Range** |
| points | Long.MIN\_VALUE .. -1 |
|  | 0 .. 59 |
|  | 60 .. 69 |
|  | 70 .. 79 |
|  | 80 .. 89 |
|  | 90 .. Long.MAX\_VALUE |
| Perfect | True |
|  | False |
| lettergrade | GRADEA |
|  | GRADEAPLUS |
|  | GRADEB |
|  | GRADEBPLUS |
|  | GRADEC |
|  | GRADECPLUS |
|  | GRADED |
|  | GRADEDPLUS |
|  | GRADEF |
|  | ERROR |

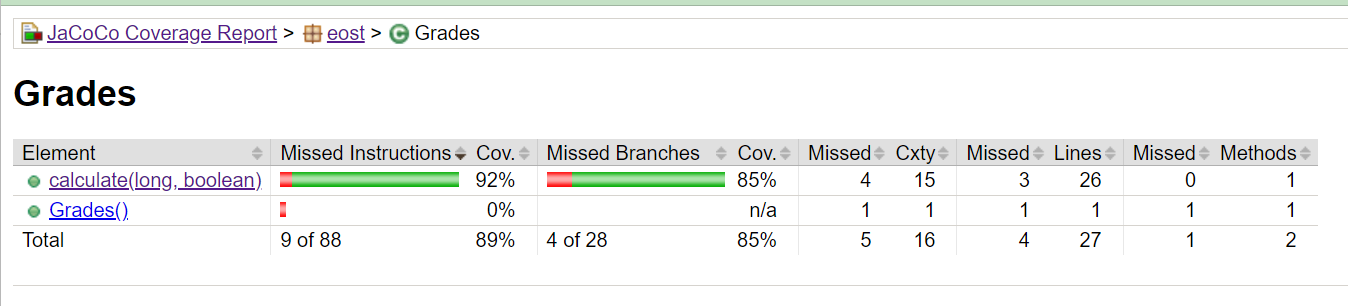
Rules for decision tables

* Points < 60
* Points < 70
* Points < 80
* Points < 90
* perfect

For the spreadsheet, double click in the area below.



Screenshot of jacoco in browser for statement coverage (before any new tests)



Screenshot of highlighted code for statement coverage (before any new tests)

A screenshot of a computer

Description automatically generated

Interpretation of screenshots for statement coverage

The Missed Instructions for the calculate method show that there is only 92% coverage, and the code shows in red that statement 35, 65, and 68 are not executed. We need to add tests for points = 128 (perfect not important), points = 77 (perfect not important) and points = 666 (points not important).

Note: this leads to additional errors in the results but that is because the faults introduced violate the specifications. What matters is that the statements are now all covered.

Screenshot of jacoco in browser for branch coverage (before any new tests)

A screenshot of a computer

Description automatically generated

Screenshot of highlighted code for branch coverage (before any new tests)



Interpretation of screenshots for branch coverage

All statements are now covered as evidenced by the Missed Instructions column now showing 100% coverage. However, branch coverage is only 96%. One of four branches on line 52 is not followed. This can be corrected by adding a test for points = 62 (perfect not important). This resolves all coverage.

---------------------------------------------- end of section 2---------------------------------------------

Section 3: TCI table and Tests table

**Test Coverage Items table. Complete the Test Case column after you add the tests for each section.**

|  |  |  |  |
| --- | --- | --- | --- |
| **TCI** | **Parameter** | **Equivalence partition, boundary value, decision, etc** | **Test case** |
| EP1\* | points | (\*) Long.MIN\_VALUE .. -1 | TEP6 |
| EP2 |  | 0 .. 59 | TEP1 |
| EP3 |  | 60 .. 69 | TEP2 |
| EP4 |  | 70 .. 79 | TEP3 |
| EP5 |  | 80 .. 89 | TEP4 |
| EP6 |  | 90 .. Long.MAX\_VALUE | TEP5 |
| EP7 | Perfect | True | TEP1 |
| EP8 |  | False | TEP2 |
| EP9 | lettergrade | GRADEA | TEP5 |
| EP10 |  | GRADEAPLUS | TDT5 |
| EP11 |  | GRADEB | TEP4 |
| EP12 |  | GRADEBPLUS | TDT4 |
| EP13 |  | GRADEC | TEP3 |
| EP14 |  | GRADECPLUS | TDT3 |
| EP15 |  | GRADED | TEP2 |
| EP16 |  | GRADEDPLUS | TDT2 |
| EP17 |  | GRADEF | TEP1 |
| EP18\* |  | (\*) ERROR | TEP6 |
| BV1 | Points | Long.MIN\_VALUE | TBV11 |
| BV2 |  | -1 | TBV12 |
| BV3 |  | 0 | TBV1 |
| BV4 |  | 59 | TBV2 |
| BV5 |  | 60 | TBV3 |
| BV6 |  | 69 | TBV4 |
| BV7 |  | 70 | TBV5 |
| BV8 |  | 79 | TBV6 |
| BV9 |  | 80 | TBV7 |
| BV10 |  | 89 | TBV8 |
| BV11 |  | 90 | TBV9 |
| BV12 |  | Long.MAX\_VALUE | TBV10 |
| BV13 | Perfect | True | TBV1 |
| BV14 |  | False | TBV2 |
| BV15 | lettergrade | GRADEA | TBV9 |
| BV16 |  | GRADEAPLUS | TDT5 |
| BV17 |  | GRADEB | TBV7 |
| BV18 |  | GRADEBPLUS | TDT4 |
| BV19 |  | GRADEC | TBV5 |
| BV20 |  | GRADECPLUS | TDT3 |
| BV21 |  | GRADED | TBV3 |
| BV22 |  | GRADEDPLUS | TDT2 |
| BV23 |  | GRADEF | TBV1 |
| BV24 |  | ERROR | TBV11 |
| DT1 | Rule 1 | 55,true | TDT1 |
| DT2 | Rule 2 | 55,false | TEP1 |
| DT3 | Rule 3 | 65, true | TDT2 |
| DT4 | Rule 4 | 65, false | TEP2 |
| DT5 | Rule 5 | 75, true | TDT3 |
| DT6 | Rule 6 | 75, false | TEP3 |
| DT7 | Rule 7 | 85, true | TDT4 |
| DT8 | Rule 8 | 85, false | TEP4 |
| DT9 | Rule 9 | 95, true | TDT5 |
| DT10 | Rule 10 | 95, false | TEP5 |
| NOTE: tests in red are evaluated late to show that it is not necessary to have all outcomes covered during equivalence partition testing. | | | |

Equivalence Values

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Equivalence partition, boundary value, decision, etc** | **Equivalence value** |
| points | Long.MIN\_VALUE .. -1 | -10 |
|  | 0 .. 59 | 55 |
|  | 60 .. 69 | 65 |
|  | 70 .. 79 | 75 |
|  | 80 .. 89 | 85 |
|  | 90 .. Long.MAX\_VALUE | 95 |
| Perfect | True | True |
|  | False | False |
| lettergrade | GRADEA | GRADEA |
|  | GRADEAPLUS | GRADEAPLUS |
|  | GRADEB | GRADEB |
|  | GRADEBPLUS | GRADEBPLUS |
|  | GRADEC | GRADEC |
|  | GRADECPLUS | GRADECPLUS |
|  | GRADED | GRADED |
|  | GRADEDPLUS | GRADEDPLUS |
|  | GRADEF | GRADEF |
|  | ERROR | ERROR |

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **TCI covered** | **Inputs** | | | **Exp. results** | **comment** |
| points | perfect |  | return value |  |
| TEP1 | EP2,EP7,EP17 | 55 | true |  | GRADEF |  |
| TEP2 | EP3,EP8,EP15 | 65 | false |  | GRADED |  |
| TEP3 | EP4,EP8,EP13 | 75 | false |  | GRADEC |  |
| TEP4 | EP5,EP8,EP11 | 85 | false |  | GRADEB |  |
| TEP5 | EP6,EP8,EP9 | 95 | false |  | GRADEA |  |
| TEP6 | EP1,EP8,EP18 | -10 | false |  | ERROR |  |
| TBV1 | BV3,BV13,BV23 | 0 | true |  | GRADEF | Will find an error due to deliberate error in the code |
| TBV2 | BV4, BV14,BV23 | 59 | false |  | GRADEF |  |
| TBV3 | BV5, BV14,BV21 | 60 | false |  | GRADED |  |
| TBV4 | BV6, BV14,BV21 | 69 | false |  | GRADED |  |
| TBV5 | BV7, BV14,BV19 | 70 | false |  | GRADEC |  |
| TBV6 | BV8, BV14,BV19 | 79 | false |  | GRADEC |  |
| TBV7 | BV9, BV14,BV17 | 80 | false |  | GRADEB |  |
| TBV8 | BV10, BV14,BV17 | 89 | false |  | GRADEB |  |
| TBV9 | BV11, BV14,BV15 | 90 | false |  | GRADEA |  |
| TBV10 | BV12, BV14,BV15 | Long.MAX\_VALUE | false |  | GRADEA | Will find an error due to deliberate error in the code |
| TBV11 | BV1, BV14,BV24 | Long.MIN\_VALUE | false |  | ERROR |  |
| TBV12 | BV2, BV14,BV24 | -1 | false |  | ERROR |  |
| TDT1 | DT1 | 55 | true |  | GRADEF |  |
| TDT2 | DT3 | 65 | true |  | GRADEDPLUS |  |
| TDT3 | DT5 | 75 | true |  | GRADECPLUS |  |
| TDT4 | DT7 | 85 | true |  | GRADEBPLUS |  |
| TDT5 | DT9 | 95 | true |  | GRADEAPLUS |  |
| TSC1 | Line 35 | 128 | true |  | GRADEAPLUS |  |
| TSC2 | Line 65 | 77 | true |  | GRADECPLUS | Will find an error due to deliberate error in the code |
| TSC3 | Line 68 | 666 | true |  | GRADEAPLUS | Will find an error due to deliberate error in the code |
| TBC1 | Line 52 | 62 | true |  | GRADEF |  |