Section 1: general information

Specifications

Description

The method determines if someone can get a concealed carry license based on several factors.

Javadoc

    public static enum Status { GRANTED, DENIED, TEMPORARY, ERROR};

    /\*\*

     \* Determine whether someone gets a license to carry concealed based on several factors.

     \* Age: Must be at least 21 years old (or at least 18 years old and a member of the military or honorably discharged veteran). Applicants over 80 years old do not qualify and are denied. Negative ages result in an error.

     \* Criminal History: Must not have a felony conviction or be subject to an outstanding felony warrant

     \* Training: completed a firearms safety training course. If they have not, they can get a temporary license for 90 days to complete the course.

     \*

     \* @param age integer

     \* @param military boolean

     \* @param no\_criminal boolean

     \* @param trained boolean

     \*

     \* @return

     \* GRANTED - all conditions are met. license is granted for five years

     \* DENIED - at least one disqualifying condition. license is denied

     \* TEMPORARY - licensed for up to 90 days after which the license is either granted or denied

     \* ERROR - for incomplete information or incorrect data

     \*/

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

age

| Integer.MIN\_VALUE --- Integer.MAX\_VALUE |

military:

|True | false|

No\_criminal:

|True | false|

trained:

|True | false|

decision

| GRANTED | DENIED | TEMPORARY | ERROR |

Specifications-based Ranges

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Natural Range** | **Equivalence value** |
| age | Long.MIN\_VALUE .. -1 | -10 |
|  | 0 .. 17 | 15 |
|  | 18 .. 20 | 19 |
|  | 21 .. 80 | 30 |
|  | 81 .. Long.MAX\_VALUE | 90 |
| military | True | True |
|  | False | False |
| No\_criminal | True | True |
|  | False | False |
| trained | True | True |
|  | False | False |
| Decision | GRANTED | GRANTED |
|  | DENIED | DENIED |
|  | TEMPORARY | TEMPORARY |
|  | ERROR | ERROR |

Rules for decision tables

* Age < 18
* Age < 21
* Age <= 80
* Military
* No\_criminal
* trained

For the spreadsheet, double click in the area below. (in this solution, I simply pasted a picture of the spreadsheet to make it fit better)

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

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

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)

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\* | age | (\*) Integer.MIN\_VALUE .. -1 | TEP5 |
| EP2 |  | 0 .. 17 | TEP1 |
| EP3 |  | 18 .. 20 | TEP2 |
| EP4 |  | 21 .. 80 | TEP3 |
| EP5 |  | 81 .. Integer.MAX\_VALUE | TEP4 |
| EP6 | military | True | TEP1 |
| EP7 |  | False | TEP2 |
| EP8 | No\_criminal | True | TEP1 |
| EP9 |  | False | TEP2 |
| EP10 | trained | True | TEP1 |
| EP11 |  | False | TEP2 |
| EP12 | Decision | GRANTED |  |
| EP13 |  | DENIED | TEP1 |
| EP14 |  | TEMPORARY |  |
| EP15\* |  | ERROR | TEP5 |
|  |  |  |  |
|  |  |  |  |
| BV1\* | Points | Integer.MIN\_VALUE | TBV9 |
| BV2\* |  | -1 | TBV10 |
| BV3 |  | 0 | TBV1 |
| BV4 |  | 17 | TBV2 |
| BV5 |  | 18 | TBV3 |
| BV6 |  | 20 | Tbv4 |
| BV7 |  | 21 | TBV5 |
| BV8 |  | 80 | TBV6 |
| BV9 |  | 81 | TBV7 |
| BV10 |  | Integer.MAX\_VALUE | TBV8 |
| BV11 | military | True | TBV1 |
| BV12 |  | False | TBV2 |
| BV13 | No\_criminal | True | TBV1 |
| BV14 |  | False | TBV2 |
| BV15 | trained | True | TBV1 |
| BV16 |  | False | TBV2 |
| BV17 | Decision | GRANTED |  |
| BV18 |  | DENIED | TBV1 |
| BV19 |  | TEMPORARY |  |
| BV20 |  | ERROR | TBV10 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| DT1 | Rule 1 | 55,true |  |
| DT2 | Rule 2 | 55,false |  |
| DT3 | Rule 3 | 65, true |  |
| DT4 | Rule 4 | 65, false |  |
| DT5 | Rule 5 | 75, true |  |
| DT6 | Rule 6 | 75, false |  |
| DT7 | Rule 7 | 85, true |  |
| DT8 | Rule 8 | 85, false |  |
| DT9 | Rule 9 | 95, true |  |
| DT10 | Rule 10 | 95, false |  |
| NOTE: tests in red are evaluated late to show that it is not necessary to have all outcomes covered during equivalence partition testing. | | | |

**Test Cases**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **TCI covered** | | **Inputs** | | | | | | **Exp. results** | **comment** |
| age | | Military | No\_criminal | trained |  | decision |  |
| TEP1 | EP2,EP7,EP9,EP11,EP13 | | 15 | | True | True | true |  | DENIED |  |
| TEP2 | EP3,EP7,EP9,EP11,EP13 | | 19 | | false | false | false |  | DENIED |  |
| TEP3 | EP4,EP7,EP9,EP11,EP13 | | 30 | | false | false | false |  | DENIED |  |
| TEP4 | EP5,EP7,EP9,EP11,EP13 | | 90 | | false | false | false |  | DENIED |  |
| TEP5 | EP1, ,EP6,EP8,EP10,EP15 | | -10 | | false | false | false |  | ERROR |  |
| TBV1 |  | | 0 | | true | true | true |  | DENIED |  |
| TBV2 |  | | 17 | | false | false | false |  | DENIED |  |
| TBV3 |  | | 18 | | false | false | false |  | DENIED |  |
| TBV4 |  | | 20 | | false | false | false |  | DENIED |  |
| TBV5 |  | | 21 | | false | false | false |  | DENIED |  |
| TBV6 |  | | 80 | | false | false | false |  | DENIED |  |
| TBV7 |  | | 81 | | false | false | false |  | DENIED |  |
| TBV8 |  | | Integer.MAX\_VALUE | | false | false | false |  | DENIED |  |
| TBV9 |  | | Integer.MIN\_VALUE | | false | false | false |  | ERROR |  |
| TBV10 |  | | -1 | | false | false | false |  | ERROR |  |
|  |  | |  | |  |  |  |  |  |  |
| TBV11 |  | |  | |  |  |  |  |  |  |
| TBV12 |  | |  | |  |  |  |  |  |  |
| TDT1 |  | |  | |  |  |  |  |  |  |
| TDT2 |  | |  | |  |  |  |  |  |  |
| TDT3 |  | |  | |  |  |  |  |  |  |
| TDT4 |  | |  | |  |  |  |  |  |  |
| TDT5 |  | |  | |  |  |  |  |  |  |
| TSC1 |  | |  | |  |  |  |  |  |  |
| TSC2 |  | |  | |  |  |  |  |  |  |
| TSC3 |  | |  | |  |  |  |  |  |  |
| TBC1 |  | |  | |  |  |  |  |  |  |
|  | |  | | Note: repetitive TCIs are noted in red rather than with square brackets as in the book. | | | | | | |