**A\_2.1.7\_Classes\_and\_Boundaries**

1953067 宋潇歌

2052225 张勤杭

2051840 梁厚

1950389 季艺

**Question**

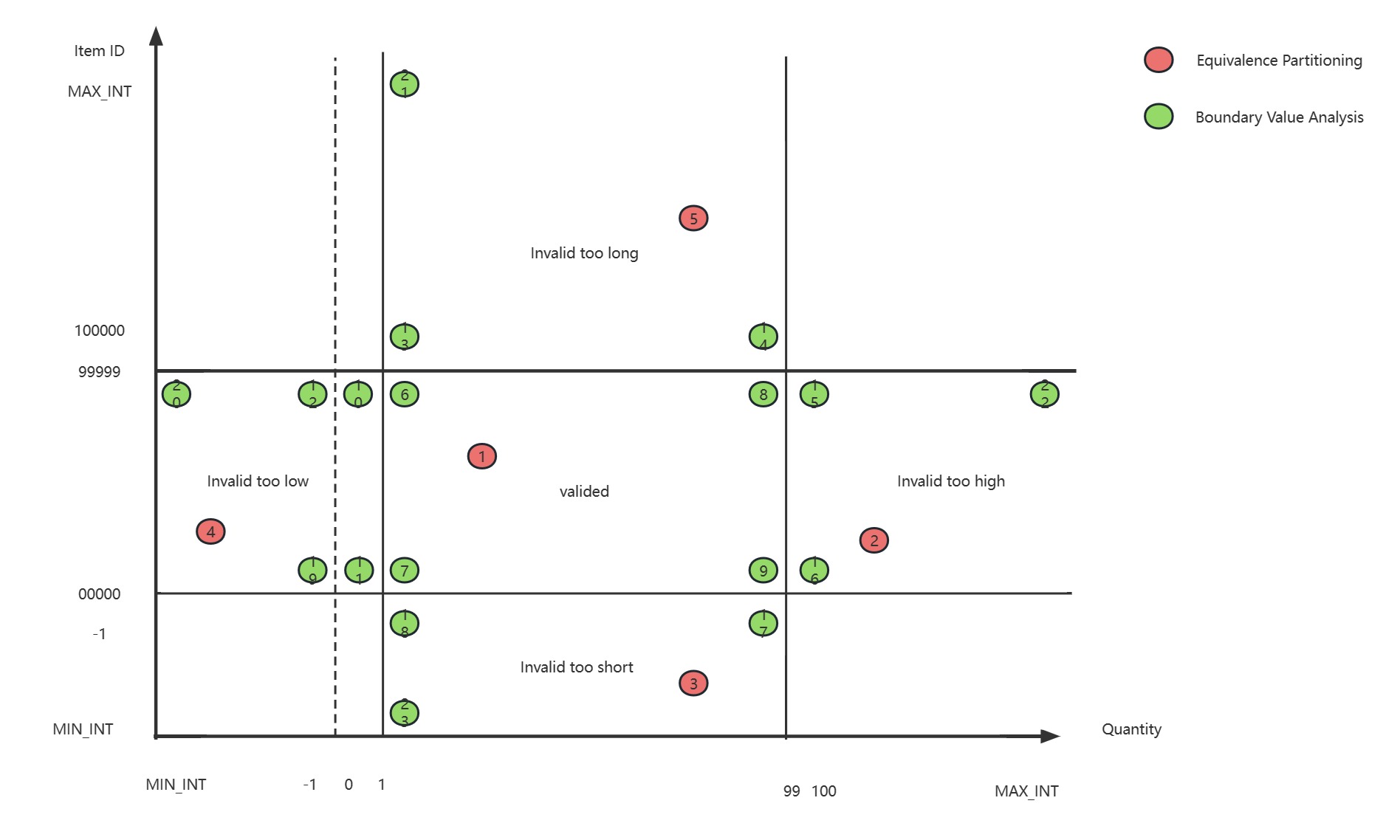
* You are testing an e-commerce site that sells Omninet knick-knacks like baseball caps, jackets, etc.
* Create functional tests for accepting orders
* The system accepts a five-digit numeric item ID number from 00000 to 99999 (integer / string?)
* Item IDs are sorted by price, with the cheapest items having the lower (close to 00000) item ID numbers and the most expensive items having the higher (close to 99999) item ID numbers
* The system accepts a quantity to be ordered, from 1 to 99
* If the user enters a previously-ordered item ID and a 0 quantity to be ordered, that item is removed from the shopping cart
* The maximum total order is $999.99
* Use boundary value analysis and equivalence class partitioning to create tests in the following template
* Discuss

**Answer**

**Overview**

According to the question, there are two inputs (item ID and quantity), and the data type of item ID is unknown, so we offer two testing solutions (when item ID is string and when item ID is integer).

**When item ID is integer, we have:**



* In Equivalence Partitioning, we plan to conduct 5 functional tests (one testing matches one area in the aforementioned chart). However, we do not test the cases in the white area, because we think that the frequency of occurrence is too low.
* In Boundary Value Analysis, we plan to conduct 4 tests to validate the boundaries of the Valided area. And 2 tests for validating the cases where quantity is equal to 0 (one is validated, because previously-ordered item ID existed. Another is not). Then 8 testing cases in the 'Invalided too long', 'Invalided too short', 'Invalid too high' and 'Invalid too low' area. Finally, we have to test the cases where input (item ID and quantity) beyond MAX\_INT and MIN\_INT.

**When item ID is string, we have:**

**Testing Case**

**Integer**

1. Equivalence Partitioning, testing 5 points in 5 different classes.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | 1.1 | 1.2 | 1.3 | 1.4 | 1.5 |
| Inputs, Action |  |  |  |  |  |
| Item ID | 70000 | 00500 | -50 | 00100 | 666666 |
| Quantity | 20 | 110 | 70 | -60 | 80 |
| Con. Shopping | Y | Y | Y | Y | N |
| Check out | N | N | N | N | Y |
| Expected Result |  |  |  |  |  |
| Item Price |  |  |  |  |  |
| Cart Contents | 70000\*20 | - | - | - | 70000\*20 |
| Cart Total | 70000Price\* 20 | - | - | - | 70000Price\* 20 |
| Wrong: quantity invalidated |  | √ |  | √ |  |
| Wrong: item ID invalidated |  |  | √ |  | √ |
| Wrong: total price too large |  |  |  |  |  |
| Wrong: quantity of this item could not be 0 |  |  |  |  |  |

1. 4 points in the 4 corners of the 'valided' area and one legal 0 case.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | 1.6 | 1.7 | 1.8 | 1.9 | 1.10 |
| Inputs, Action |  |  |  |  |  |
| Item ID | 99999 | 00000 | 99999 | 00000 | 99999 |
| Quantity | 1 | 1 | 99 | 99 | 0 |
| Con. Shopping | Y | Y | Y | Y | N |
| Check out | N | N | N | N | Y |
| Expected Result |  |  |  |  |  |
| Item Price |  |  |  |  |  |
| Cart Contents | 99999\*1 | 00000\*1 | 99999\*99 | 00000\*99 | - |
| Cart Total | 99999Price\*1 | 00000Price\*1+99999Price\*1 | 00000Price\*1+99999Price\*1+99999Price\*99 | 00000Price\*1+99999Price\*1+99999Price\*99+00000Price\*99 | 00000Price\*1+00000Price\*99 |
| Wrong: quantity invalidated |  |  |  |  |  |
| Wrong: item ID invalidated |  |  |  |  |  |
| Wrong: total price too large |  |  |  |  |  |
| Wrong: quantity of this item could not be 0 |  |  |  |  |  |

1. Illegal 0 and 8 testing cases in separated corners.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | 1.11 | 1.12 | 1.13 | 1.14 | 1.15 |
| Inputs, Action |  |  |  |  |  |
| Item ID | 00000 | 99999 | 100000 | 100000 | 99999 |
| Quantity | 0 | -1 | 1 | 99 | 100 |
| Con. Shopping | Y | Y | Y | Y | Y |
| Check out | N | N | N | N | N |
| Expected Result |  |  |  |  |  |
| Item Price |  |  |  |  |  |
| Cart Contents | - | - | - | - | - |
| Cart Total | - | - | - | - | - |
| Wrong: quantity invalidated |  | √ |  |  | √ |
| Wrong: item ID invalidated |  |  | √ | √ |  |
| Wrong: total price too large |  |  |  |  |  |
| Wrong: quantity of this item could not be 0 | √ |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | 1.16 | 1.17 | 1.18 | 1.19 |  |
| Inputs, Action |  |  |  |  |  |
| Item ID | 00000 | -1 | -1 | 00000 |  |
| Quantity | 100 | 99 | 1 | -1 |  |
| Con. Shopping | Y | Y | Y | N |  |
| Check out | N | N | N | Y |  |
| Expected Result |  |  |  |  |  |
| Item Price |  |  |  |  |  |
| Cart Contents | - | - | - | - |  |
| Cart Total | - | - | - | - |  |
| Wrong: quantity invalidated | √ |  |  | √ |  |
| Wrong: item ID invalidated |  | √ | √ |  |  |
| Wrong: total price too large |  |  |  |  |  |
| Wrong: quantity of this item could not be 0 |  |  |  |  |  |

1. 4 cases to validate the boundary of data type (INT)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | 1.20 | 1.21 | 1.22 | 1.23 |  |
| Inputs, Action |  |  |  |  |  |
| Item ID | 99999 | MAX\_INT | 99999 | MIN\_INT |  |
| Quantity | MIN\_INT | 1 | MAX\_INT | 1 |  |
| Con. Shopping | Y | Y | Y | N |  |
| Check out | N | N | N | Y |  |
| Expected Result |  |  |  |  |  |
| Item Price |  |  |  |  |  |
| Cart Contents | - | - | - | - |  |
| Cart Total | - | - | - | - |  |
| Wrong: quantity invalidated | √ |  | √ |  |  |
| Wrong: item ID invalidated |  | √ |  | √ |  |
| Wrong: total price too large |  |  |  |  |  |
| Wrong: quantity of this item could not be 0 |  |  |  |  |  |

**String**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | 2.1 | 2.2 | 2.3 | 2.4 | 2.5 |
| Inputs, Action |  |  |  |  |  |
| Item ID |  |  |  |  |  |
| Quantity |  |  |  |  |  |
| Con. Shopping |  |  |  |  |  |
| Check out |  |  |  |  |  |
| Expected Result |  |  |  |  |  |
| Item Price |  |  |  |  |  |
| Cart Contents |  |  |  |  |  |
| Cart Total |  |  |  |  |  |
| Wrong: quantity invalidated |  |  |  |  |  |
| Wrong: item ID invalidated |  |  |  |  |  |
| Wrong: total price too large |  |  |  |  |  |
| Wrong: quantity of this item could not be 0 |  |  |  |  |  |