TutoriaI 3 − BIack−Box Testing: Boundary VaIue

AnaIysis

# Black−Box Testing. Black−Box Testing Techniques.

Boundary Value Analysis.

In black−box testing the software product is viewed as an opaque system (there is no access to the internal details, including the source code). Black−box testing techniques are also known as behavioral or behavior−based testing techniques. Black−box testing can be applied for both functional and nonfunctional testing.

Black−box testing techniques focus on the inputs and outputs of the item tested.

When applying black−box testing techniques, the test conditions, test cases, test data are created from a test basis that may include software requirements, specifications, use cases, and user stories.

Test coverage is a measurement of the number of tests that have been executed. Test cases are written to ensure maximum coverage of requirements (e.g. software requirements specification, functional requirements specification, user requirements specification, etc.).

Black−box testing techniques:

* Equivalence Partitioning
* Boundary Value Analysis
* Decision Table Testing
* State Transition Testing
* Use Case Testing

# Boundary Value Analysis

Boundary value analysis is a refinement/extension of the equivalence partitioning technique. Recall that the equivalence partitioning technique divides a set of test conditions/data into partitions/ equivalence classes, where all the elements that belong to the same partition/equivalence are expected to be considered the same, so the software system should process/handle them in the same way (i.e. handle them equivalently). The boundary value analysis technique can be applied only for ordered partitions, formed from numeric or sequential data.

For boundary value analysis, each limit of an equivalence class is covered by a test, together with a value just above and just below the limit.

**Key Insight:** *the behavior at the boundaries of equivalence partitions is more likely to be incorrect than behavior within the partitions.*

**Exercises**

1. A marks input form includes an integer field that should contain values between 1 and

100. Apply the boundary value analysis technique to identify all the test values so that boundary coverage is achieved.

1. Any Number greater than 100 entered is considered invalid.
2. Any Number less than 1 that is 0 or below, then it is considered invalid.
3. Numbers 1 to 100 are considered valid
4. Any 4 Digit Number say 1000 or -1000 is invalid.

|  |  |  |  |
| --- | --- | --- | --- |
| Invalid | Valid | Invalid | Invalid |
| 0 | 1 100 | 101 999 | 1000 |

Boundary Value Analysis- in Boundary Value Analysis, you test boundaries between equivalence partitions

Sample tests:

Test cases of entering the value 0

Test cases of entering the value 1

Test cases of entering the value 2

Test cases of entering the value 99

Test cases of entering the value 100

Test cases of entering the value 999

Test cases of entering the value 1000

1. An input box accepts a single integer value as an input which represents the age of a person. Based on the age entered it should be decided whether the person is legible to drive (i.e. age between 17 inclusive and 70 inclusive). Apply the boundary value analysis technique to identify all the test values so that boundary coverage is achieved.
2. You are required to test a system that based on the entered achieved overall programme grade assigns an award classification. Based on the overall grade achieved the classification are as follows:

|  |  |
| --- | --- |
| 0 − 39 | Fail |
| 40 − 49 | Pass |
| 50 − 59 | H2.2 |
| 60 − 69 | H2.1 |
| 70 − 100 | H1 |

Apply the boundary value analysis technique to identify all the test values so that boundary coverage is achieved.

1. An employee has the first 4000EUR of salary tax free. The next 1500 is taxed at 10%. The next 28000 is taxed at 22%. Any further amount is taxed at 40%. Apply the boundary value analysis technique to identify all the test values so that boundary coverage is achieved. (Adapted from a question from ISTQB).

**Reference**

Brian Hambling et al. (2019). *Software Testing: An ISTQB−BCS Certified Tester Foundation guide*