**Homework T2 Submission Document**

You can work individually or in groups of 2.

|  |  |  |  |
| --- | --- | --- | --- |
| Names: | KD Adkins |  |  |

**Test Derivation for *addEmp(e:Employee)***

We will use the category-partition method to derive a set of test cases for this method.

1. **Identify what criteria must hold true for each test case to pass**

The side-effects of this method are:

1. **Identify inputs/parameters for each feature under test.**
2. **Identify the categories/characteristics for each input/parameter.**
3. **Partition categories into choices**
4. **Number of tests before constraints added**
5. **TSL Input**

The *TSL* input file is shown below.

Justification for the constraints:

|  |  |
| --- | --- |
| **Constraint** | **Justification** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

1. **TSL Output – Test Frames**

The \_\_ test frames are show below:

1. **Test Cases**

The test cases we derived are shown below.

[Remove unused columns]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test** | **[Put name of characteristic 1 here]** | **[Put name of characteristic 2 here]** | ***Etc.*** |  |  |  |  |  |
| 1 |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

**Test Derivation for *removeEmployee(pos:int):Employee***

We will use the category-partition method to derive a set of test cases for this method.

1. **Identify what criteria must hold true for each test case to pass**

The side-effects of this method are:

* OutOfBoundsException. If a user removes an employee at the end of the array, it’ll try and shift out of bounds.
* If we run the method and the array is empty.

1. **Identify inputs/parameters for each feature under test.**

The index being passed through is needed.

Variable to return the moved employee will be needed.

Will need the employees array and variable for number of employees

1. **Identify the categories/characteristics for each input/parameter.**

The index that gets passed is an integer.

Variable that holds the removed employee should be of type Employee.

Employee array is type employee with values.

Number of employees is a positive integer.

1. **Partition categories into choices**

Exceed Boundaries: index = Is the index < 0 or > 20

Valid Data: Null, Non-Null, Zero, Non-Zero

1. **Number of tests before constraints added**

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Category | Choices | Coded Choices |
| index | Value | Negative, Positive, Exceed | N, P, X |
| emps | Value | Null, NN | N, NN |
| numEmp | Value | Zero, Non-Zero | Z, NZ |

I have 12 combinations (3\*2\*2).

NNZ, NNNZ, NNNZ, NNNNZ, PNZ, PNNZ, PNNZ, PNNNZ, XNZ, XNNZ,

XNNZ, XNNNZ

1. **TSL Input**

The *TSL* input file is shown below.

Justification for the constraints:

|  |  |
| --- | --- |
| **Constraint** | **Justification** |
| NNNZ | There’s no way you can have a null array of employees and numEmp is >= 1 |
| PNNZ | There’s no way you can have a null array of employees and numEmp is >= 1 |
| XNNZ | There’s no way you can have a null array of employees and numEmp is >= 1 |
| PNNZ | numEmps can’t be zero if there’s non-null values in emps |
| XNNZ | numEmps can’t be zero if there’s non-null values in emps |
| NNNZ | numEmps can’t be zero if there’s non-null values in emps |

1. **TSL Output – Test Frames**

The \_6\_ test frames are show below:

1. **Test Cases**

The test cases we derived are shown below.

[Remove unused columns]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test** | **[Value]** | **[Value]** | **[Value]** |  |  |  |  |  |
| 1 | N | N | Z |  |  |  |  |  |
| 2 | N | N | NZ |  |  |  |  |  |
| 3 | N | NN | Z |  |  |  |  |  |
| 4 | N | NN | NZ |  |  |  |  |  |
| 5 | P | N | Z |  |  |  |  |  |
| 6 | P | N | NZ |  |  |  |  |  |
| 7 | P | NN | Z |  |  |  |  |  |
| 8 | P | NN | NZ |  |  |  |  |  |
| 9 | X | N | Z |  |  |  |  |  |
| 10 | X | N | NZ |  |  |  |  |  |
| 11 | X | NN | Z |  |  |  |  |  |
| 12 | X | NN | NZ |  |  |  |  |  |