

|  |  |
| --- | --- |
|  | Description |
| Originally, q.Enqueue(A) | |
|  | **Visit** |
|  | While : Add all siblings of B to the Q  Exit while loop, go back  **Visit**  Add all siblings of to the Q |
|  | **Visit**  Add all children of to the Q |
|  | **Visit**  Add all children of (nothing) to the Q |
|  | **Visit**  Add all children (nothing) to the Q |
|  | **Visit**  Add all children of (nothing) to the Q |
|  | **Visit**  Add all children of () to the Q |
|  | **Visit**  Add all children of (nothing) to the Q |
|  | **Visit**  Add all children of (nothing) to the Q |
|  | **Visit**  Add all children of (nothing) to the Q |
| After the second while loop (that adds all the children of the last node), the queue is empty, meaning there is no element left.  Terminate.  **Order**: A B F C D E G H J K | |

Depth First Search

|  |  |
| --- | --- |
|  | Description / Steps |
| Originally, | |
|  | **Visit**  Push first sibling of (nothing) to stack |
|  | **Visit**  Push first sibling of () to stack |
|  | **Visit**  Push first sibling of () to stack  Exit inner loop and get from stack |
|  | **Visit**  Push first sibling of () to stack  Exiting inner loop and get from stack |
|  | **Visit**  Push first sibling of E (nothing) to stack  Exiting inner loop and get from stack |
|  | **Visit**  Pushing first sibling of (nothing) to stack |
|  | **Visit**  Pushing first sibling of (nothing) to stack |
|  | **Visit**  Pushing first sibling of () to stack  Exiting from the inner while loop and get from stack |
|  | **Visit**  Pushing first sibling of () to stack  Exiting from the inner while loop and get from stack |
|  | **Visit**  Pushing first sibling of (nothing) to stack  Exiting from the inner while loop and get from stack  But stack is empty… |
| Since the stack is empty when we are trying to pop from it, it is time to stop.  Order: A, B, C, D, E, F, G, H, J, K | |

Question 3: Cat Pedigree

*This section only include the test plan. For test results, refer to section… in the Appendix.*

Cat Class

**Cat.IsValidName**: Name must not be null, and must be between 1 and 8 characters long (inclusive)

|  |  |  |
| --- | --- | --- |
| **Description** | **Input Data** | **Expected Output** |
| Name is null should return false | name = null | False |
| Name is empty should return false | “” | False |
| Name is 9 characters long should return false | “Sebastian” | False |
| Name is more than 9 characters long should return false | “Shakespeare” (11 chars) | False |
| Name is 1 character long should return true | “A” | True |
| Name is 8 characters long should return true | “Lorenzo” | True |
| Name is between 2 and 7 characters long should return true | “Dragon” (6 chars) | True |

**Cat.IsValidAge**: Age must be between 0 and 30 (inclusive)

|  |  |  |
| --- | --- | --- |
| **Description** | **Input Data** | **Expected Output** |
| Age is -1 should return false | -1 | False |
| Age is < - 1 should return false | -5 | False |
| Age is 31 should return false | 31 | False |
| Age is > 31 should return false | 39 | False |
| Age is 0 should return true | 0 | True |
| Age is 30 should return true | 30 | True |
| Age is between 1 and 29 (inclusive) should return true | 15 | True |

*Pedigree Class*

**Pedigree.AddCat**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Precondition** | **Input Data** | **Expected Output** | **Postcondition** |
| Add duplicate cats should return false | Pedigree containing Paws. | Add cat named “Paws” to the pedigree. | False. | Pedigree only contains Paws |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Pedigree.FindCat**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Precondition** | **Input Data** | **Expected Output** | **Postcondition** |
| Find a cat in an empty pedigree should return null | Empty pedigree | Find(“Paws”) | null | Pedigree unchanged |
| Find a cat whom exists in a pedigree of one cat | Pedigree of one cat:  Paws | Find(“Paws”) | Cat with name Paws | Pedigree unchanged |
| Find a cat whom exists in a pedigree of multiple cats | Pedigree of 4 cats:  Paws  Luke  Xander  Moby | Find(“Xander”) | Cat with name Xander | Pedigree unchanged |
| Find a cat who does not exist in a pedigree of one cat | Pedigree of one cat:  Paws | Find(“Xander”) | null | Pedigree unchanged |
| Find a cat who does not exist in a pedigree of multiple cats | Pedigree of multiple cats:  Paws  Luke | Find(“Xander”) | null | Pedigree unchanged |

**Pedigree.ToArray**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Precondition** | **Input Data** | **Expected Output** | **Postcondition** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Example Test Result:

