**Block Chain Bitcoin and Security Fundamental – Code Project**

Contents

[Attributes and Composite Types: 3](#_Toc104938146)

[Methods/Function: 3](#_Toc104938147)

[Function \_createTweets 3](#_Toc104938148)

[Function tweet\_length 3](#_Toc104938149)

[Function getTweetId 4](#_Toc104938150)

[Function \_readTweets 4](#_Toc104938151)

[Function \_update\_tweet 5](#_Toc104938152)

[Function \_delete\_tweet 5](#_Toc104938153)

# Attributes and Composite Types:

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| tweet\_id\_len | uint | Variable to control the length of the tweet\_id |
| struct tweet\_type | struct | Structure type having two simple types uint (tweet\_id) and string (tweet\_text) |
| tweets | tweet\_type[] | array of type struct tweet\_type for holding tweets |

# Methods/Function:

# Function \_createTweets

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameters** | **Parameter Type** | **Return Type** |
| \_createTweets | \_tweet\_text | string memory | NA |

**Description:**

* Calls getTweetId to generate tweet\_id. Push (tweet\_id, \_tweet\_text) into tweets array.

# Function tweet\_length

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameters** | **Parameter Type** | **Return Type** |
| tweet\_length | NA | NA | uint |

**Description**

* It returns the length of the tweets array. This function as been defined since the array is of dynamic size.

# Function getTweetId

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameters** | **Parameter Type** | **Return Type** |
| getTweetId | \_tweet\_text | string memory | uint |

**Description**

* Using keccak256 we are generating a random hexa-decimal number, typecasting into uint. Since we are fixing the size of the tweet\_id as a 16 digits long number, we return the modules of tweet\_id\_len.

# Function \_readTweets

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameters** | **Parameter Type** | **Return Type** |
| \_readTweets | NA | string memory | string[] memory |

**Description:**

* Creating a new array of type string memory called just\_tweets. From tweets array copying only the tweet\_text into just\_tweets array. Returning just\_tweets array.

# Function \_update\_tweet

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameters** | **Parameter Type** | **Return Type** |
| \_update\_tweet | \_old\_tweet | string memory | Bool |
| \_new\_tweet | string memory |

**Description:**

* Generating the old\_tweet\_id of \_old\_tweet parameter using getTweetId function. Loop through the tweets array, check each tweet\_id with the old\_tweet\_id. When found update its tweet\_text with the \_new\_tweet parameter.
* When the update operation fails return false.
* When the update operation is successful generate new\_tweet\_id of \_new\_tweet using getTweetId function. Update the new tweet’s tweet\_id with the new\_tweet\_id. Finally return true.

# Function \_delete\_tweet

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Parameters** | **Parameter Type** | **Return Type** |
| \_delete\_tweet | \_tweet\_text | string memory | Bool |

**Description:**

* Generate tweet\_id of \_tweet\_text parameter using getTweetId function. Loop through tweet array check the generated tweet\_id of the tweets array, if found delete the tweet from the tweets array.
* When delete operation fails return false.
* When the delete operation is successful:

1. We remove the gap created in the tweets array, by the delete operation.
2. We remove the tweet at the last index because after the previous step the gap would have propagated to the end of the tweets array.