Table of Contents

Queue

Queue is a first in first out (FIFO) data structure, allowing to process their elements in the order they come.

Usage

Create and Clear

You can create a queue manually by using the new or construct it using the from_ array.

```
1
2    test {
3      let _queue : @queue.Queue[Int] = @queue.new()
4      let _queue1 = @queue.of([1, 2, 3])
5
6    }
```

To clear the queue, you can use the clear method.

```
1
2    test {
3      let queue = @queue.of([1, 2, 3])
4      queue.clear()
5    }
```

Length

You can get the length of the queue by using the length method. The is_empty method can be used to check if the queue is empty.

```
1
2    test {
3      let queue = @queue.of([1, 2, 3])
4      assert_eq(queue.length(), 3)
5      assert_eq(queue.is_empty(), false)
6    }
```

Pop and Push

You can add elements to the queue using the push method and remove them using the pop method.

```
1
2    test {
3      let queue = @queue.new()
4      queue.push(1)
5      queue.push(2)
6      assert_eq(queue.pop(), Some(1))
7      assert_eq(queue.pop(), Some(2))
8    }
```

Peek

You can get the first element of the queue without removing it using the peek method.

```
1
2    test {
3      let queue = @queue.of([1, 2, 3])
4      assert_eq(queue.peek(), Some(1))
5    }
```

Traverse

You can traverse the queue using the each method.

```
1
2    test {
3      let queue = @queue.of([1, 2, 3])
4      let mut sum = 0
5      queue.each(x => sum += x)
6      assert_eq(sum, 6)
7    }
```

You can fold the queue using the fold method.
```
1
2    test {
3      let queue = @queue.of([1, 2, 3])
4      let sum = queue.fold(init=0, (acc, x) => acc + x)
5      assert_eq(sum, 6)
6    }
```

Copy and Transfer

You can copy a queue using the copy method.
```
1
2    test {
3      let queue = @queue.of([1, 2, 3])
4      let _queue2 = queue.copy()
5
6    }
```

Transfer the elements from one queue to another using the transfer method.
```
1
2    test {
3      let dst : @queue.Queue[Int] = @queue.new()
4      let src : @queue.Queue[Int] = @queue.of([5, 6, 7, 8])
5      src.transfer(dst)
6    }
```