

Test Plan

1. Create a Quokka object with the default constructor.
2. Create a Quokka object with the non-default constructor.
 - a. With valid field values.
3. Test all get methods.
 - a. Test `getBabyCounted()`.
 - b. Test `getConsecutiveDaysNotEaten()`.
 - c. Test `getDaysNotEaten()`.
 - d. Test `getFoodSupply()`.
 - e. Test `getHasBaby()`.
 - f. Test `getId()`.
 - g. Test `getIsAlive()`.
4. Test all set methods.
 - a. Test `setBabyCounted()`.
 - i. With valid field values.
 - ii. With invalid field values.
 - b. Test `setConsecutiveDaysNotEaten()`.
 - i. With valid field values.
 - ii. With invalid field values.
 - c. Test `setDaysNotEaten()`.
 - i. With valid field values.
 - ii. With invalid field values.
 - d. Test `setFoodSupply()`.
 - i. With valid field values.
 - ii. With invalid field values.
 - e. Test `setHasBaby()`.
 - i. With valid field values.
 - ii. With invalid field values.
 - f. Test `setId()`.
 - i. With valid field values.
 - ii. With invalid field values.
 - g. Test `setIsAlive()`.
 - i. With valid field values.

- ii. With invalid field values.
- 5. Test the other methods.
 - a. Test decrementFoodSupply().
 - i. With valid field values.
 - ii. With invalid field values.
 - b. Test incrementConsecutiveDaysNotEaten().
 - c. Test incrementDaysNotEaten().
 - d. Test incrementFoodSupply().
 - i. With valid field values.
 - ii. With invalid field values.
 - e. Test toString().

Actual Test

Test 1

Create a Quokka object with the default constructor.

Test data:

- n/a

Expected results:

- id: "Q001"
- isAlive: true
- foodSupply: 2
- hasBaby: false
- daysNotEaten: 0
- consecutiveDaysNotEaten: 0
- babyCounted: false

Actual results:

```
Quokka Id: Q001
Quokka is Alive: true
Quokka Food Supply: 2
Quokka has Baby: false
Number of Days Quokka has not Eaten: 0
Number of Consecutive Days Quokka has not Eaten: 0
Baby has been Counted: false
```

Test Passed!

Test 2.1a.1

Create a Quokka object with the non-default constructor with valid field values.

Test data:

- hasBaby: true

Expected results:

- id: "Q002"
- isAlive: true
- foodSupply: 3
- hasBaby: true
- daysNotEaten: 0
- consecutiveDaysNotEaten: 0
- babyCounted: false

Actual results:

```
Quokka Id: Q002
Quokka is Alive: true
Quokka Food Supply: 3
Quokka has Baby: true
Number of Days Quokka has not Eaten: 0
Number of Consecutive Days Quokka has not Eaten: 0
Baby has been Counted: false
```

Test Passed!

Test 2.1a.2

Create a Quokka object with the non-default constructor with valid field values.

Test data:

- hasBaby: false

Expected results:

- id: "Q003"
- isAlive: true
- foodSupply: 2
- hasBaby: false
- daysNotEaten: 0
- consecutiveDaysNotEaten: 0
- babyCounted: false

Actual results:

Quokka Id: Q003
Quokka is Alive: true
Quokka Food Supply: 2
Quokka has Baby: false
Number of Days Quokka has not Eaten: 0
Number of Consecutive Days Quokka has not Eaten: 0
Baby has been Counted: false

Test Passed!

Test 3.1

Test getBabyCounted().

Test data:

- n/a

Expected results:

- babyCounted: false

Actual results:

Test getBabyCounted().
babyCounted: false

Test Passed!

Test 3.2

Test getConsecutiveDaysNotEaten().

Test data:

- n/a

Expected results:

- consecutiveDaysNotEaten: 0

Actual results:

Test getConsecutiveDaysNotEaten().
consecutiveDaysNotEaten: 0

Test Passed!

Test 3.3

Test getDaysNotEaten().

Test data:

- n/a

Expected results:

- daysNotEaten: 0

Actual results:

```
Test getDaysNotEaten().  
daysNotEaten: 0 _
```

Test Passed!

Test 3.4

Test getFoodSupply().

Test data:

- n/a

Expected results:

- foodSupply: 2

Actual results:

```
Test getFoodSupply().  
foodSupply: 2
```

Test Passed!

Test 3.5

Test getHasBaby().

Test data:

- n/a

Expected results:

- hasBaby: false

Actual results:

```
Test getHasBaby().  
hasBaby: false
```

Test Passed!

Test 3.6

Test getId().

Test data:

- n/a

Expected results:

- id: "Q009"

Actual results:

```
Test getId().  
Id: Q009
```

Test Passed!

Test 3.7

Test getIsAlive().

Test data:

- n/a

Expected results:

- isAlive: true

Actual results:

```
Test getIsAlive().  
isAlive: true
```

Test Passed!

Test 4.1a

Test setBabyCounted() with valid values.

Test data:

- babyCounted: true

Expected results:

- babyCounted: true

Actual results:

```
Test setBabyCounted() with valid values.  
babyCounted: true
```

Test Passed!

Test 4.1b

Test setBabyCounted () with invalid values.

Test data:

- babyCounted: false

Expected results:

- babyCounted: true

Actual results:

```
Test setBabyCounted() with invalid values.  
babyCounted: true
```

Test Passed!

Test 4.2a

Test setConsecutiveDaysNotEaten() with valid values.

Test data:

- consecutiveDaysNotEaten: 0

Expected results:

- consecutiveDaysNotEaten: 0

Actual results:

```
Test setConsecutiveDaysNotEaten() with valid values.  
consecutiveDaysNotEaten: 0
```

Test Passed!

Test 4.2b

Test setConsecutiveDaysNotEaten () with invalid values.

Test data:

- consecutiveDaysNotEaten: -1

Expected results:

- consecutiveDaysNotEaten: 0

Actual results:

```
Test setConsecutiveDaysNotEaten() with invalid values.  
consecutiveDaysNotEaten: 0
```

Test Passed!

Test 4.3a

Test setDaysNotEaten() with valid values.

Test data:

- daysNotEaten: 0

Expected results:

- daysNotEaten: 0

Actual results:

```
Test setDaysNotEaten() with valid values.  
daysNotEaten: 0
```

Test Passed!

Test 4.3b

Test setDaysNotEaten() with invalid values.

Test data:

- daysNotEaten: -1

Expected results:

- daysNotEaten: 0

Actual results:

```
Test setDaysNotEaten() with invalid values.  
daysNotEaten: 0
```

Test Passed!

Test 4.4a

Test setFoodSupply() with valid values.

Test data:

- foodSupply: 0

Expected results:

- foodSupply: 0

Actual results:

```
Test setFoodSupply() with valid values.  
foodSupply: 0
```

Test Passed!

Test 4.4b

Test setFoodSupply() with invalid values.

Test data:

- foodSupply: -1

Expected results:

- foodSupply: 0

Actual results:

```
Test setFoodSupply() with invalid values.  
foodSupply: 0
```

Test Passed!

Test 4.5a

Test setHasBaby() with valid values.

Test data:

- hasBaby: true

Expected results:

- hasBaby: true

Actual results:

Test setHasBaby() with valid values.
hasBaby: true

Test Passed!

Test 4.5b

Test setHasBaby() with invalid values.

Test data:

- hasBaby: false

Expected results:

- hasBaby: true

Actual results:

Test setHasBaby() with invalid values.
hasBaby: true

Test Passed!

Test 4.6a

Test setId() with valid values.

Test data:

- id: "1"

Expected results:

- id: "Q001"

Actual results:

Test setId() with valid values.
id: Q001

Test Passed!

Test 4.6b

Test setId() with invalid values.

Test data:

- id: "-1"

Expected results:

- id: "Q023"

Actual results:

Test setId() with invalid values.

id: Q023

Test Passed!

Test 4.7a

Test setIsAlive() with valid values.

Test data:

- isAlive: false

Expected results:

- isAlive: false

Actual results:

Test setIsAlive() with valid values.

isAlive: false

Test Passed!

Test 4.7b

Test setIsAlive() with invalid values.

Test data:

- isAlive: true

Expected results:

- isAlive: false

Actual results:

Test setIsAlive() with invalid values.

isAlive: false

Test Passed!

Test 5.1a

Test decrementFoodSupply() with valid values.

Test data:

- amountEaten: 2

Expected results:

- foodSupply: 0

Actual results:

Test decrementFoodSupply() with valid values.
foodSupply: 0

Test Passed!

Test 5.1b

Test decrementFoodSupply() with invalid values.

Test data:

- amountEaten: -1

Expected results:

- foodSupply: 2

Actual results:

Test decrementFoodSupply() with invalid values.
foodSupply: 2

Test Passed!

Test 5.2

Test incrementConsecutiveDaysNotEaten().

Test data:

- n/a

Expected results:

- consecutiveDaysNotEaten: 1

Actual results:

Test incrementConsecutiveDaysNotEaten().
consecutiveDaysNotEaten: 1

Test Passed!

Test 5.3

Test incrementDaysNotEaten().

Test data:

- n/a

Expected results:

- daysNotEaten: 1

Actual results:

```
Test incrementDaysNotEaten().  
daysNotEaten: 1
```

Test Passed!

Test 5.4a

Test incrementFoodSupply() with valid values.

Test data:

- touristPayment: 1

Expected results:

- foodSupply: 3

Actual results:

```
Test incrementFoodSupply() with valid values.  
foodSupply: 3
```

Test Passed!

Test 5.4b

Test incrementFoodSupply() with invalid values.

Test data:

- touristPayment: -1

Expected results:

- foodSupply: 2

Actual results:

Test incrementFoodSupply() with invalid values.
foodSupply: 2

Test Passed!

Test 5.5

Test toString().

Test data:

- n/a

Expected results:

- id: "Q031"
- isAlive: true
- foodSupply: 2
- hasBaby: false
- daysNotEaten: 0
- consecutiveDaysNotEaten: 0
- babyCounted: false

Actual results:

Test toString().

Quokka Id: Q031

Quokka is Alive: true

Quokka Food Supply: 2

Quokka has Baby: false

Number of Days Quokka has not Eaten: 0

Number of Consecutive Days Quokka has not Eaten: 0

Baby has been Counted: false

Test Passed!

Note: Test Script = TestingQuokka.java