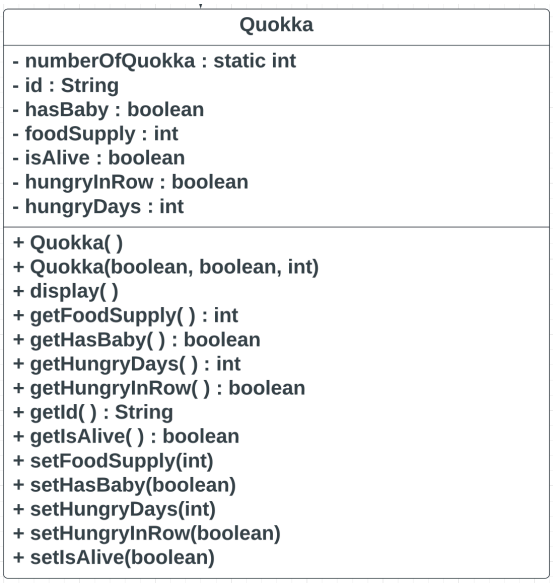

- **Test plan**

1. Create an Quokka object with the default constructor.
2. Create an Quokka object with the non-default constructor:
    - with valid field values
    - with invalid field values
3. Test all get methods:
    - Test getFoodSupply( )
    - Test getHasBaby( )
    - Test getHungryDays( )
    - Test getHungryInRow( )
    - Test getId( )
    - Test getIsAlive( )
4. Test all set methods:
    - Test setFoodSupply( )
        - with valid field values
        - with invalid field values
    - Test setHasBaby( )
        - with valid field values
    - Test setHungryDays( )
        - with valid field values
        - with invalid field values
    - Test setHungryInRow( )
        - with valid field values
    - Test setIsAlive( )
        - with valid field values
        - with invalid field values
5. Test other methods:
    - Test display( )

The actual tests

Test 1

Create a Quokka object with the default constructor.

Test data:

- Id : “Q001”
- hasBaby : false
- foodSupply : 2
- isAlive : true
- hungryInRow : false
- hungryDays : 0

Expected results:

- Quokka ID : Q001
- Quokka has baby : false
- Quokka’s food supply : 2
- Quokka is alive : true
- Quokka is hungry today : false
- The number of days the quokka is hungry : 0

Actual results:

```
Test default constructor:

Quokka ID: Q001
Quokka has baby: false
Quokka's food supply: 2
Quokka is alive: true
Quokka is hungry today: false
The number of days the quokka is hungry: 0
```

Test 2.1.a

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

Test data:

- id : “Q001”
- hasBaby : true
- foodSupply : 3
- isAlive : true
- hungryInRow : false
- hungryDays : 0

Expected results:

- Quokka ID : Q001
- Quokka has baby : true
- Quokka's food supply : 3
- Quokka is alive : true
- Quokka is hungry today : false
- The number of days the quokka is hungry : 0

Actual results:

```
Test non-default constructor with valid field values(1):

Quokka ID: Q001
Quokka has baby: true
Quokka's food supply: 3
Quokka is alive: true
Quokka is hungry today: false
The number of days the quokka is hungry: 0
```

Test 2.1.b

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

Test data:

- Quokka ID : "Q001"
- hasBaby : true
- foodSupply : 3
- isAlive : true
- hungryInRow : false
- hungryDays : 5

Expected results:

- Quokka ID : Q001
- Quokka has baby : true
- Quokka's food supply : 3
- Quokka is alive : true
- Quokka is hungry today : false
- The number of days the quokka is hungry : 5

Actual results:

```
Test non-default constructor with valid field values(2):

Quokka ID: Q001
Quokka has baby: true
Quokka's food supply: 3
Quokka is alive: true
Quokka is hungry today: false
The number of days the quokka is hungry: 5
```

Test 2.2.a

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

Test data:

- id : “Q001”
- hasBaby : true
- foodSupply : 3
- isAlive : true
- hungryInRow : false
- hungryDays : -1

Expected results:

- Quokka ID : Q001
- Quokka has baby : true
- Quokka’s food supply : 3
- Quokka is alive : true
- Quokka is hungry today : false
- The number of days the quokka is hungry : 0

Actual results:

```
Test non-default constructor with invalid field values(1):

Quokka ID: Q001
Quokka has baby: true
Quokka's food supply: 3
Quokka is alive: true
Quokka is hungry today: false
The number of days the quokka is hungry: 0
```

Test 2.2.b

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

Test data:

- id : “Q001”
- hasBaby : true
- foodSupply : 3
- isAlive : true
- hungryInRow : false
- hungryDays : 6

Expected results:

- Quokka ID : Q001
- Quokka has baby : true
- Quokka's food supply : 3
- Quokka is alive : true
- Quokka is hungry today : false
- The number of days the quokka is hungry : 0

Actual results:

```
Test non-default constructor with invalid field values (2):

Quokka ID: Q001
Quokka has baby: true
Quokka's food supply: 3
Quokka is alive: true
Quokka is hungry today: false
The number of days the quokka is hungry: 0
```

Test 3.1

Test accessor method (Food supply)

Test data:

- id : "Q001"
- hasBaby : true
- foodSupply : 3
- isAlive : true
- hungryInRow : false
- hungryDays : 0

Expected results:

- 3

Actual results:

```
Test accessor method (Food supply):
3
```

Test 3.2

Test accessor method (Has baby)

Test data:

- id : “Q001”
- hasBaby : true
- foodSupply : 3
- isAlive : true
- hungryInRow : false
- hungryDays : 0

Expected results:

- true

Actual results:

```
Test accessor method (Has baby):
true
```

Test 3.3

Test accessor method (ID)

Test data:

- id : “Q001”
- hasBaby : true
- foodSupply : 3
- isAlive : true
- hungryInRow : false
- hungryDays : 0

Expected results:

- Q001

Actual results:

```
Test accessor method (ID):
Q001
```

Test 3.4

Test accessor method (Is alive)

Test data:

- id : “Q001”
- hasBaby : true
- foodSupply : 3
- isAlive : true
- hungryInRow : false
- hungryDays : 0

Expected results:

- true

Actual results:

```
Test accessor method (Is alive):
true
```

Test 4.1.a

Test mutator method with valid value(Food supply)

Test data:

- foodSupply : 0

Expected results:

- 0

Actual results:

```
Test mutator method with valid value(Food supply):
0
```

Test 4.1.b

Test mutator method with invalid value(Food supply)

Test data:

- foodSupply : -1

Expected results:

- 0

Actual results:

```
Test mutator method with invalid value(Food supply):
0
```

Test 4.2

Test mutator method with valid value(Has baby)

Test data:

- hasBaby: true

Expected results:

- True

Actual results:

```
Test mutator method with valid value(Has baby):
true
```

Test 4.3.a

Test mutator method with valid value(Hungry days)

Test data:

- hungryDays : 0

Expected results:

- 0

Actual results:

```
Test mutator method with valid value(Hungry days):
0
```

Test 4.3.b

Test mutator method with valid value(Hungry days)

Test data:

- hungryDays : 5

Expected results:

- 5

Actual results:

```
Test mutator method with valid value(Hungry days):
5
```

Test 4.3.c

Test mutator method with invalid value(Hungry days)

Test data:

- hungryDays : -1

Expected results:

- 0

Actual results:

```
Test mutator method with invalid value(Hungry days):
0
```

Test 4.3.d

Test mutator method with invalid value(Hungry days):

Test data:

- hungryDays : 6

Expected results:

- 0

Actual results:

```
Test mutator method with invalid value(Hungry days):
0
```

Test 4.4

Test mutator method with valid value(Hungry in row)

Test data:

- hungryInRow : true

Expected results:

- true

Actual results:

```
Test mutator method with valid value(Hungry in row):
true
```

Test 4.5.a

Test mutator method with valid value(Is alive)

Test data:

- isAlive : true

Expected results:

- true

Actual results:

```
Test mutator method with valid value(Is alive):
true
```

Test 4.5.b

Test mutator method with invalid value(Is alive)

=> If the quokka is dead, it cannot be set alive again.

Test data:

- isAlive : false

Expected results:

- False

Actual results:

```
Test mutator method with invalid value(Is alive):
false
```

Test 4.6

Test display method

Test data:

- id : “Q001”
- hasBaby : true
- foodSupply : 0
- isAlive : false
- hungryInRow : true
- hungryDays : 0

Expected results:

- Quokka ID : Q001
- Quokka has baby : true
- Quokka’s food supply : 0
- Quokka is alive : false
- Quokka is hungry today : true
- The number of days the quokka is hungry : 0
-

Actual results:

```
Test display method:

Quokka ID: Q001
Quokka has baby: true
Quokka's food supply: 0
Quokka is alive: false
Quokka is hungry today: true
The number of days the quokka is hungry: 0
```