Whale class test strategy

We have been provided with the following specification details:

- Species is is a string from the set {Humpback, Minke, Southern Right, Blue, Orca}
- Type is a string from the set {adult,calf}.
- Direction is a string from the set {north,south}.
- Location is a string from the set {Eden, JervisBay, Byron Bay, Hervey Bay}.
- Injured is either true to indicate whale is injured, or false to indicate whale is healthy.

Test plan

The following steps outline an approach to a test plan:

1. Create an Whale object with the default constructor.
2. Create an Whale object with the non-default constructor:
    - with valid field values
    - with invalid field values
3. Test all get methods:
    - Test getSpecies()
    - Test getType()
    - getLocation()
    - getDirection()
    - getInjured()
4. Test all set methods:
    - Test setSpecies(String species)
        - with valid field values
        - with invalid field values
    - Test setType(String type)
        - with valid field values
        - with invalid field values
    - setDirection(String direction)
        - with valid field values
        - with invalid field values
    - setLocation(String location)

        - with valid field values
        - with invalid field values
    - setInjured(boolean injured)
        - with valid field values
        - with invalid field values

It is important to note the following:

- In the test plan, every method which has a formal parameter must do both positive and negative testing.
- If a class has multiple non-default constructors, then each constructor should be specified as an independent test case.
- A negative test could involve multiple values. When this happens, each value being tested is shown as an independent subcase under the same test heading.

The actual tests

Test 1

Create an Whale object with the default constructor.

Test data:

- species: " unknown "
- type: "unknown"
- Direction: "unknown"
- Location: "unknown"
- Injured: "false"

Expected results:

- species: "unknown"
- type: "unknown"
- Direction: "unknown"
- Location: "unknown"
- Injured: "false"

Actual results:

Test 2.1

Create an Whale object with the non-default constructor with valid field values.

Test data:

- species: "Humpback"
- type: "adult"
- Direction: "north"
- Location: " Eden "
- Injured: "false"

Expected results:

- species: "Humpback "
- type: "adult"
- Direction: " north"
- Location: " Eden"
- Injured: "false"

Actual results:

Test 2.2

Create an Whale object with the non-default constructor with invalid field values.

Test data:

- species: "abc"
- type: 'sdf'
- Direction: " sds"
- Location: " aaa"
- Injured: "true"

Expected results:

- species: "unknown"
- type: "unknown"
- Direction: " unknown "
- Location: "unknown "

- Injured: "false"

Actual results:

Test 3

Test all get methods.

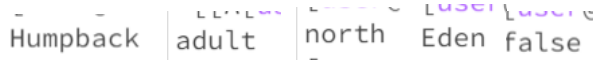
Test data:

- species: "Humpback"
- type: "adult"
- Direction: "north"
- Location: " Eden "
- Injured: "false"

Expected results:

- species: "Humpback"
- type: "adult"
- Direction: "north"
- Location: " Eden "
- Injured: "false"

Actual results:



Test 4.1

Test all set methods with valid field values.

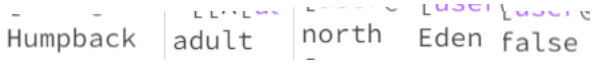
Test data:

- species: "Humpback"
- type: "adult"
- Direction: "north"
- Location: " Eden "
- Injured: "false"

Expected results:

- species: "Humpback"
- type: "adult"
- Direction: "north"
- Location: " Eden "
- Injured: "false"

Actual results:



Test 4.2

Test all set methods with invalid field values.

Test data:

- species: "abc"
- type: 'sdf'
- Direction: " sds"
- Location: " aaa"
- Injured: "true"

Expected results:

- species: "unknown"
- type: "unknown"
- Direction: " unknown "
- Location: "unknown "
- Injured: "false"

Actual results: