

Test Cases

Black-Box Test Cases

These test cases are defined based on the available documentation and execution of the program. The code was not inspected.

#	Test case (very brief description)	Preconditions (any required setup)	Test steps (steps executed during testing)	Expectation	Observation ("pass" or failure description)
1	Setting sex of pet	-	create -> choose pet -> choose name -> start game	You can give your pet a sex, when you create pet	Failure. Not implemented
2	The pet doesnt have a favorite food and toy	pet is hungry; Pet is bored	feed -> choose food; Play -> choose toy	better stats after feed with the favorite food; Better stats after play with the favorite toy; The favorite toy and food will be written in the stats	Failure. No information given about favorite food and toy. No better stats after playing oder feeding.
3	With which stat values should the pet be created.	-	Create -> choose pet -> choose name -> start game	After pet is created, all 3 stats should be 10 (hunger, dirtiness, boredom)	Failure. All stats 3 stats are 5.
4	How the stats behave	Existing pet	do something	All three stats should be worse when decreasing.	Failure. Pet should die when all 3 stats are 0.

5	When does the pet die	Existing pet	do nothing for hours	The stats should decrease with time.	Failure. The stats dont decrease, so the pet does not die.
6	Load non existing pet	-	Load -> "random string"	Error Message. Try again.	Failure. Programm crashed.
7	Load extern text file	Existing extern text file	Load -> ../„Name text file“	Error Message. Try again. Should not allow reading outside of folder.	Failure. Programm crashed or loaded external file.
8	Empty string as petname	-	Create -> choose pet -> „“	Error Message. Try again.	Failure. Pet with empty name is created.
9	Create pet with existing name	Existing pet	Create -> choose pet -> „Name of existing Pet“	Warning, if I want to override the file.	Failure. The existing pet will be overwritten with new pet.
10	Behavior when asleep	Sleeping pet	Save/Close; Feed/clean/play/sleep;	Lets you save or close game; Does not let you do anything else;	Failure. Doesnt save/close, because pet is sleeping; Pass;
11	Setting pet asleep	Existing pet	Sleep -> „Random string“	Error Message. Try again.	Failure. Infinite Loop.
12	Letting the pet sleep on.	Sleeping pet	Any command -> Back	Pet will sleep on.	Failure. Infinite Loop.

White-Box Test Cases

These additional test cases were defined during inspection of the code.

#	Test case (very brief description)	Preconditions (any required setup)	Test steps (steps executed during testing)	Expectation	Observation ("pass" or failure description)
1	TestFood	Existing pet	Junit tests (testing constructors)	Correct object	pass (4 of 4)
2	TestToy	Existing pet	Junit tests (testing constructors)	Correct object	pass (3 of 3)
3	TestTime	Existing pet	Junit tests (testing constructors)	Correct object	pass (1 of 1)
4	TestPet	Existing pet	Junit tests (testing constructors) (2) TestClean(); Testfeed(); TestPlay();	Correct object. Correct changed stats;	Pass (3 of 5) Failure; Failure;
5	TestGame	-	Junit tests (Input Tests)	Correct object/input	pass (10 of 14)
6	TestGame Beispiel revive()	Existing pet with stats value 10	Start game -> interact with pet	Before Interaction should be output that your pet died.	Failure. Only check_death call after Interaction, so Pet can not die.
7	TestGame Beispiel Wakeup pet	Sleeping Pet	Sleep -> wake up -> yes	The Pet wakes up	Pass
8	Name Pet „None“	-	Create -> choose pet -> „None“	Creating pet with name none	Failure. Warning Message.