

**Developer**: Aidan Farhi

**Date**: 11/19/2023

# IT 145 Global Rain Summary Report Template

## Pseudocode

method checkInPet() {

CALL getCheckInInfoFromClient()

CALL checkBoardingSpace(pet)

IF there is no boarding space

PRINT 'Sorry, there is not enough room for your pet'

RETURN false

END IF

CALL assignPetSpace(pet)

RETURN true

}

method getCheckInInfoFromClient() {

IF pet is a returning pet

CALL handleReturningPet(pet)

ELSE

CALL handleNewPet(pet)

END IF

GET length of stay, and grooming preference info from the client

IF the owner wants the pet to be groomed

IF the pet is a cat

PRINT 'Sorry, only dogs can be groomed'

ELSE IF the pet is a dog and the length of stay is less than two days

PRINT 'Sorry, the length of stay must be at least two days for grooming'

CALL pet.setGrooming(false)

ELSE

CALL pet.setGrooming(true)

END IF

END IF

}

method handleNewPet(Pet pet) {

GET pet information from client

SET pet information

}

method handleReturningPet(Pet pet) {

IF pet info needs to be updated

GET updated information from client

SET pet information

END IF

}

method checkBoardingSpace(Pet pet) {

IF the type of pet is a dog

IF there are not enough spaces for dogs

RETURN false

END IF

ELSE

IF there are not enough spaces for cats

RETURN false

END IF

END IF

RETURN true

}

method assignPetSpace(Pet pet) {

IF the type of pet is a dog

GET an available dog space

CALL pet.setDogSpaceNumber(available space)

SET dog space as occupied

ELSE

GET an available cat space

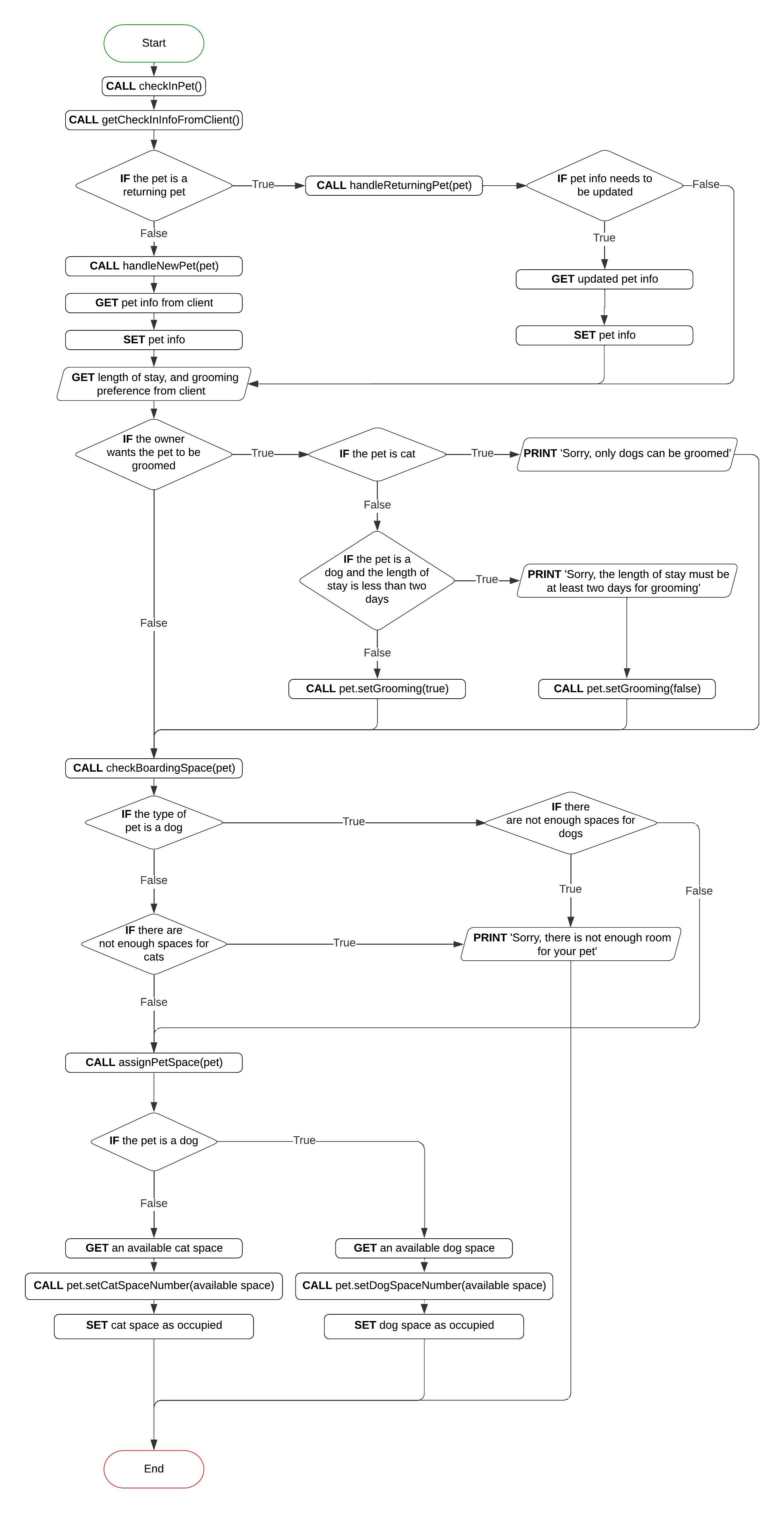
CALL pet.setCatSpaceNumber(available space)

SET cat space as occupied

END IF

}

## Flowchart



## OOP Principles Explanation

The Pet class that I developed was designed using object-oriented programming principles. All of the data and methods to operate on the data related to a pet are bundled together. The attributes of the class are all private, which prevents direct access to the data fields. If a user wishes to manipulate the state of the object, they must use the public getter and setter methods. These characteristics of my class exemplify the concept of encapsulation, one of the four pillars of object-oriented programming.