

Hobby animals

Documentation

Made by: Illia Takhtamyshev

Contents

Task	3
Plan.....	4
Table animal – day	7
getLowestELAnimals specification.....	7
getLowestELAnimals algorithm.....	8
Testing.....	9
1) Give an input and check exhilLevels of animals in the output after the simulation.....	9
2) Give an input and check getLowestELAnimals output	9

Task

Animals need several things to preserve their exhilaration. Cathy has some hobby animals: fishes, birds, and dogs. Every animal has a name and exhilaration level between 0 and 100 (0 means that the animal dies).

If their keeper is in a good mood, she takes care of everything to cheer up her animals, and their exhilaration level increases: of the fishes by 1, of the birds by 2, and of the dogs by 3.

On an ordinary day, she cares of only the dogs (their exhilaration level does not change), so the exhilaration level of the rest decreases: of the fishes by 3, of the birds by 1.

On a bad day, every animal becomes a bit sadder and their exhilaration level decreases. Off the fishes by 5, of the birds by 3, and of the dogs by 10.

Cathy's mood improves by one if the exhilaration level of every alive animal is at least 5.

Every data is stored in a text file. The first line contains the number of animals. Each of the following lines contain the data of one animal: one character for the type (F – Fish, B – Bird, D – Dog), name of the animal (1 word), and starting level of exhilaration.

In the last line, the daily moods of Cathy are enumerated by a list of characters (g – good, o – ordinary, b – bad). The file is assumed to be correct.

Name the animal of the lowest level of exhilaration which is still alive at the end of the simulation. If there are more, name all.

Possible input file:

3

F Nemo 50

B Hedwig 70

D Lassie 20

ooooggbgbgoogg

Plan

return day

```
switch(d):  
  case 'g':  
    day := GoodDay.Instance()  
    break  
  case 'o':  
    if(everyELevelAt15()) then  
      day := GoodDay.Instance()  
      break  
    endif  
  case 'b':  
    if(everyELevelAt15()) then  
      day := OrdinaryDay.Instance()  
      break  
    endif  
    day := BadDay.Instance()  
    break  
endswitch
```

```
greater := true  
for(i = 0 to |animals|) loop  
  if(animals[i].getExhilLevel() < 5) then  
    greater := false  
  endif  
endloop  
return greater
```

return name

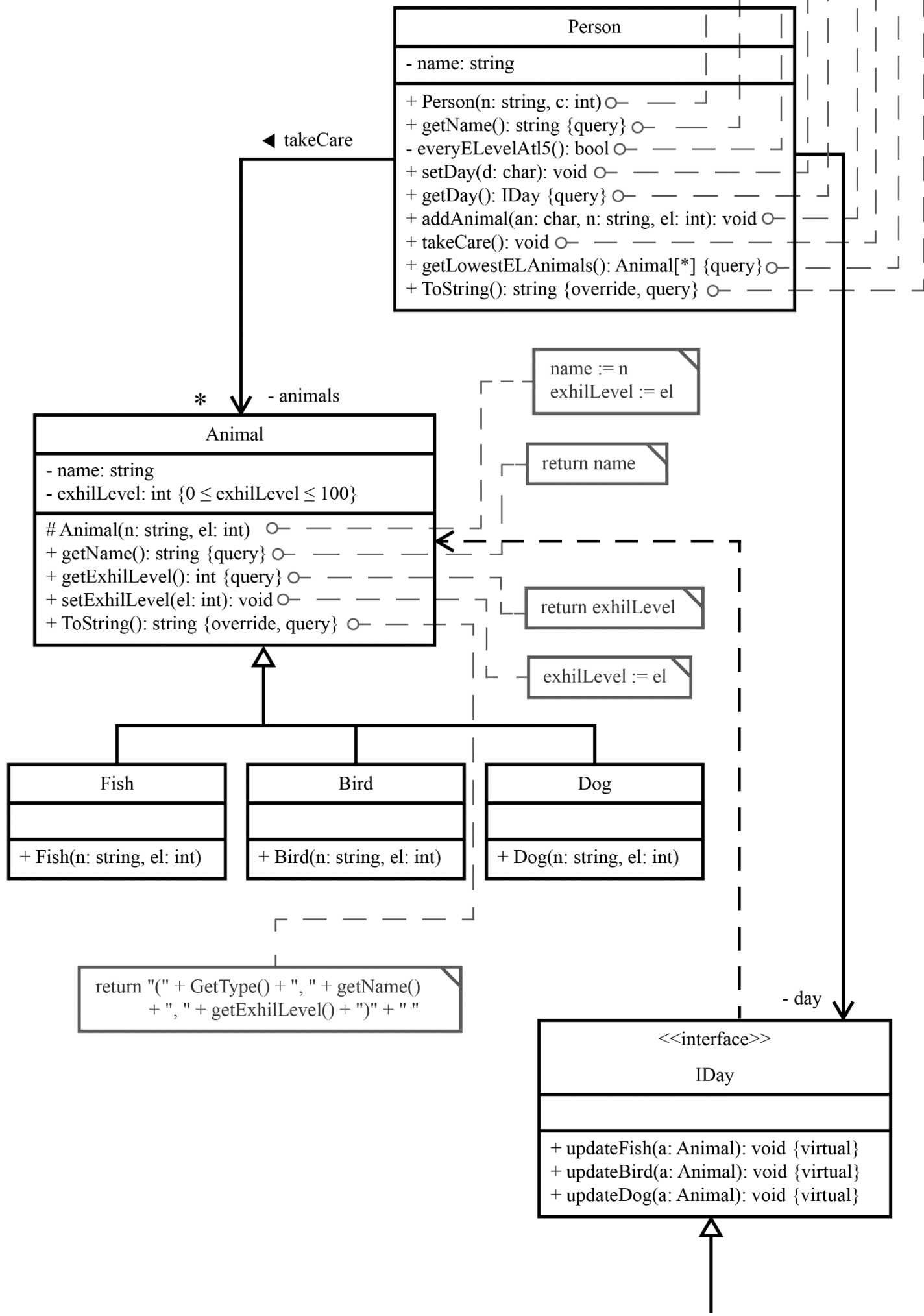
```
name := n  
animals := new List<Animal>(c)
```

```
temp := ""  
foreach(Animal a in animals) loop  
  temp := temp + a.ToString()  
endloop  
return temp
```

below

```
foreach(Animal a in animals) loop  
  if(a is Fish) then  
    day.updateFish(a)  
  elseif(a is Bird) then  
    day.updateBird(a)  
  elseif(a is Dog) then  
    day.updateDog(a)  
  endif  
endloop
```

```
switch(an):  
  case 'F':  
    animals.Add(new Fish(n, el))  
    break  
  case 'B':  
    animals.Add(new Bird(n, el))  
    break  
  case 'D':  
    animals.Add(new Dog(n, el))  
    break  
endswitch
```



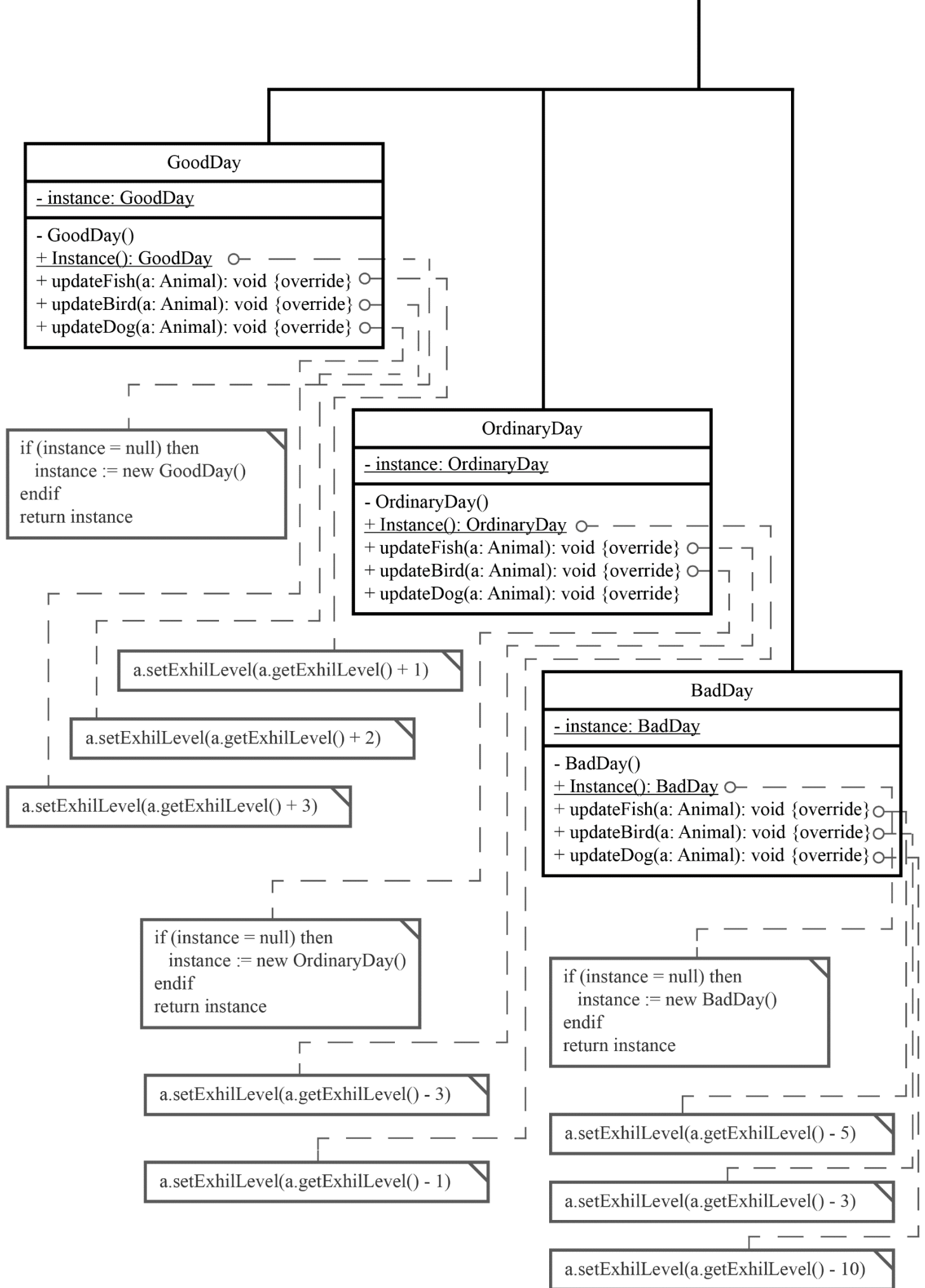


Table animal – day

	Fish	Bird	Dog
GoodDay	+1	+2	+3
OrdinaryDay	-3	-1	0
BadDay	-5	-3	-10

getLowestELAnimals specification

$A = (\text{animals} : \text{Animal}^n, \text{found} : \mathbb{L}, i : \mathbb{Z}, \text{lowestEL} : \mathbb{Z}, \text{lowests} : \text{Animal}^n)$

$\text{Pre} = (\text{animals} = \text{animals}')$

$\text{Post} = (\text{Pre} \wedge (\text{found}, i, \text{lowestEL}, \text{lowests}) = \text{SEARCH}_{a \in \text{animals}'_{1..n}} (a.\text{getExhilLevel}() > 0))$

$\wedge (\text{lowestEL}, \text{lowests}) = \text{MIN}_{\substack{a \in \text{animals}'_{i..n} \\ (a.\text{getExhilLevel}() > 0)}} a.\text{getExhilLevel}()$

getLowestELAnimals algorithm

found := false			
i := 1			
i < animals && ¬found			
animals[i].getExhilLevel() > 0			
lowestEL := animals[i].getExhilLevel()		—	
lowests.Add(animals[i])			
found := true			
i := i + 1			
¬found			
throw error			
i < animals			
animals[i].getExhilLevel() > 0			
animals[i].getExhilLevel() < lowestEL			
lowests := <>		animals[i].getExhilLevel() = lowestEL	
lowestEL := animals[i].getExhilLevel()		lowests.Add(animals[i])	—
lowests.Add(animals[i])			
i := i + 1			
return lowests			

Testing

1) Give an input and check exhilLevels of animals in the output after the simulation

- a) Test 1 with input: 4 animals: fish, dog, bird, fish; 3 days: bad, good, ordinary
- b) Test 2 with input: 3 animals: bird, fish, bird; 4 days: bad, bad, ordinary, good
- c) Test 3 with input: 3 animals: dog, fish, bird; 5 days: bad, good, ordinary, ordinary, ordinary

2) Give an input and check getLowestELAnimals output

- a) Test 4 with input: 3 animals: fish, dog, bird; 10 days: ooggbgbgog
- b) Test 5 with input: 4 animals: dog, dog, fish, bird; 8 days: ogbgbogg
- c) Test 6 with input: 4 animals: bird, fish, dog, fish; 12 days: ggbgooogbgbb
- d) Test 7 with input: 3 animals: bird, dog, fish; 7 days: bbbbbbb