OOP – HOBBY ANIMALS

JAMALA KHALIGOVA (A0JCKR)

2nd May 2020. 3rd assignment/ 9th task

Task

Hobby animals need several things to preserve their exhilaration. Cathy has some hobby animals: fishes, birds, and dogs. Every animal has a name and their exhilaration level is between 0 and 100 (0 means that the animals 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, Cathy takes care 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: of 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 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 (one word), and the initial 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 of them!

Analysis[1]

Main and independent objects in the task are the hobby animals.There are 3 groups of hobby animals: Fishes, Dogs and Birds.

Each one of these animals has name and exhilaration level.Exhilaration level of these animals depends on Catchy's daily mood. We can see effects of this below:

Fishes:

mood	*exhilaration change*
good	*+1*
ordinary	*-3*
bad	*-5*

Dogs:

mood	*exhilaration change*
good	*+3*
ordinary	0
bad	*-10*

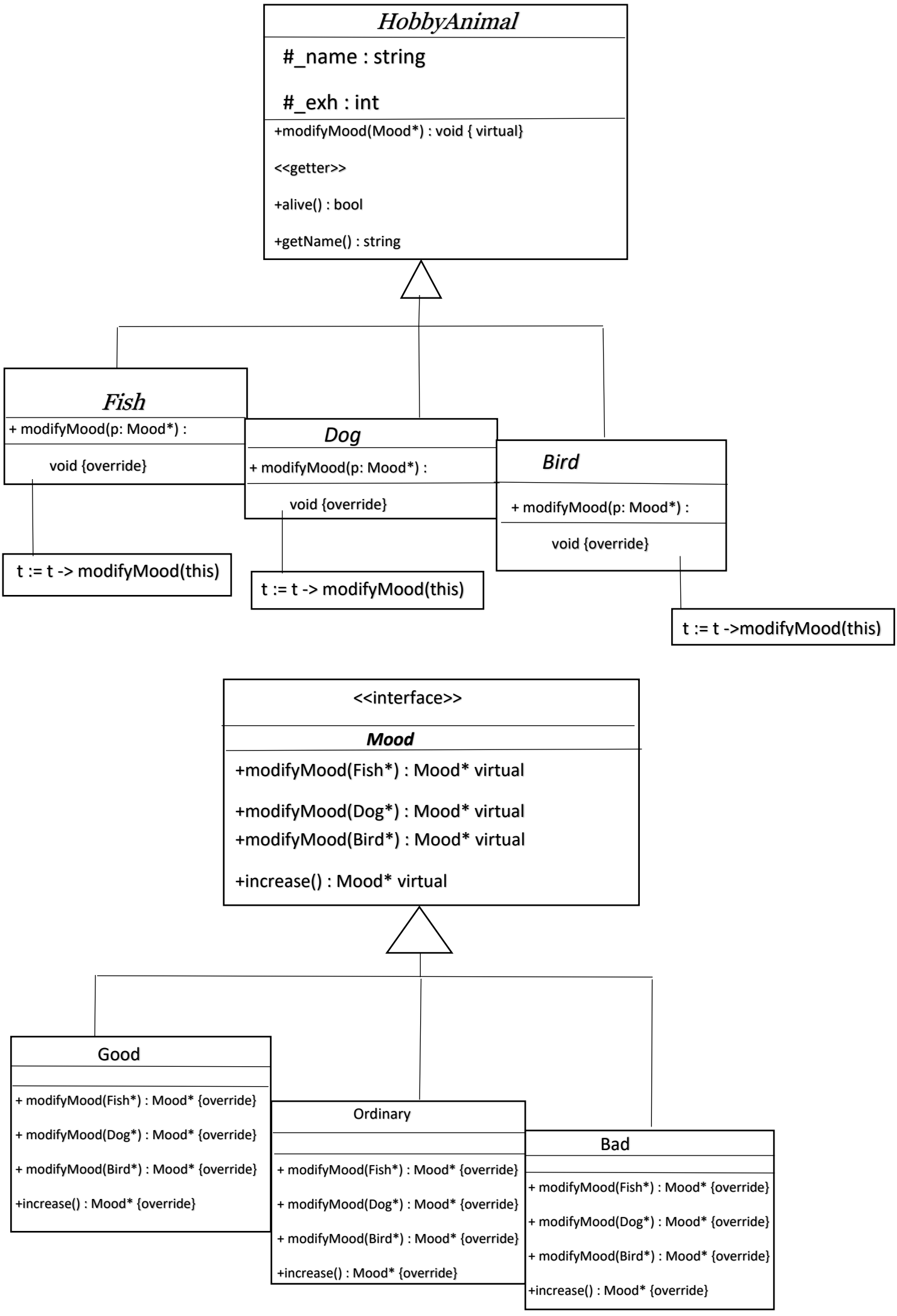
Birds:

mood	*exhilaration change*
good	*+2*
ordinary	*-1*
bad	*-3*

Plan[2]

To describe the hobby animals, base class is implemented (HobbyAnimals) which describes common properties and 3 children for the concrete hobby animals: Bird, Dog, and Fish. Regardless the type of the hobby animals, they have several common properties, like the name (_name) and the exhilaration (_exh), the getter of its name (getName()), if it is alive (alive()) and it can be examined by exhilaration based on Cathy's mood. This latter operation (increase()) modifies daily mood if all animals' _exh are more than 5. Operations alive() and getName() may be implemented in the base class already, but modifyMood() is on the level of the concrete classes as its effect depends on the type of hobby animal. Therefore, the general class HobbyAnimal is going to be abstract, as method modifyMood() is abstract and we do not wish to instantiate such class. General description of the moods is done by the base class Mood from which concrete moods are inherited: Good, Ordinary, and Bad. Every concrete mood has three methods that show how a Fish, a Dog, or a Bird changes its exhilaration. Objects are referred by pointers. The specific hobby animal classes initialize the name and the exhilaration through the constructor of the base class and override the operation modifyMood() and increase() in a unique way. Initialization and the override are explained in Section Analysis. According to the tables, in method modifyMood(), conditionals have to be used in which mood is examined.

Note: Visitor is applied where the mood classes are going to have the role of visitor.

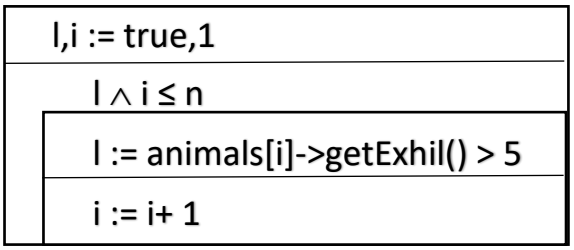

HobbyAnimal
#_name : string
#_exh : int
+modifyMood(Mood*) : void { virtual}
<<getter>>
+alive() : bool
+getName() : string
Fish
+ modifyMood(p: Mood*) :
void {override}
Dog
+ modifyMood(p: Mood*) :
void {override}
Bird
+ modifyMood(p: Mood*) :
void {override}
t := t -> modifyMood(this)
t := t -> modifyMood(this)
t := t ->modifyMood(this)
<<interface>>
Mood
+modifyMood(Fish*) : Mood* virtual
+modifyMood(Dog*) : Mood* virtual
+modifyMood(Bird*) : Mood* virtual
+increase() : Mood* virtual
Good
+ modifyMood(Fish*) : Mood* {override}
+ modifyMood(Dog*) : Mood* {override}
+ modifyMood(Bird*) : Mood* {override}
+increase() : Mood* {override}
Ordinary
+ modifyMood(Fish*) : Mood* {override}
+ modifyMood(Dog*) : Mood* {override}
+ modifyMood(Bird*) : Mood* {override}
+increase() : Mood* {override}
Bad
+ modifyMood(Fish*) : Mood* {override}
+ modifyMood(Dog*) : Mood* {override}
+ modifyMood(Bird*) : Mood* {override}
+increase() : Mood* {override}

Program allHappy()

A = *animals*: vector<HobbyAnimal* ^{n} >

Pre = *animals = animals*$_0$

Post = Post = (Pre $\wedge$ I = $\forall SEARCH_{i=1..n}$ animals(i)-->getExhil()>5)

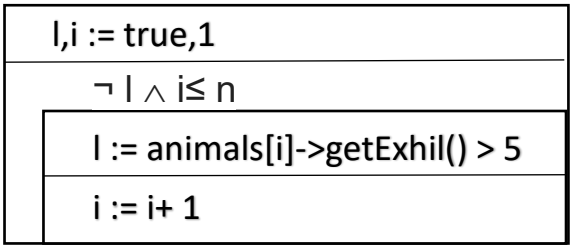


Program anyAlive()

A = *animals*: vector<HobbyAnimal* ^{n} >

Pre = *animals = animals*$_0$

Post = Post = (Pre $\wedge$ I = $\forall SEARCH_{i=1..n}$ animals(i)-->alive())



Main Program

A = mood:Mood^{m} , animals : HobbyAnimals^{n},min_alive: String^{k}.

Pre = animals = animals$_{0}$ $_{0}$ $\wedge$ mood = mood$_{0}$

Post = mood = mood$_{n}$ $\wedge$ $\forall i \in [1..n]$: *animals*[*i*], mood = mood -> increase() $\wedge$
allHappy(animals)

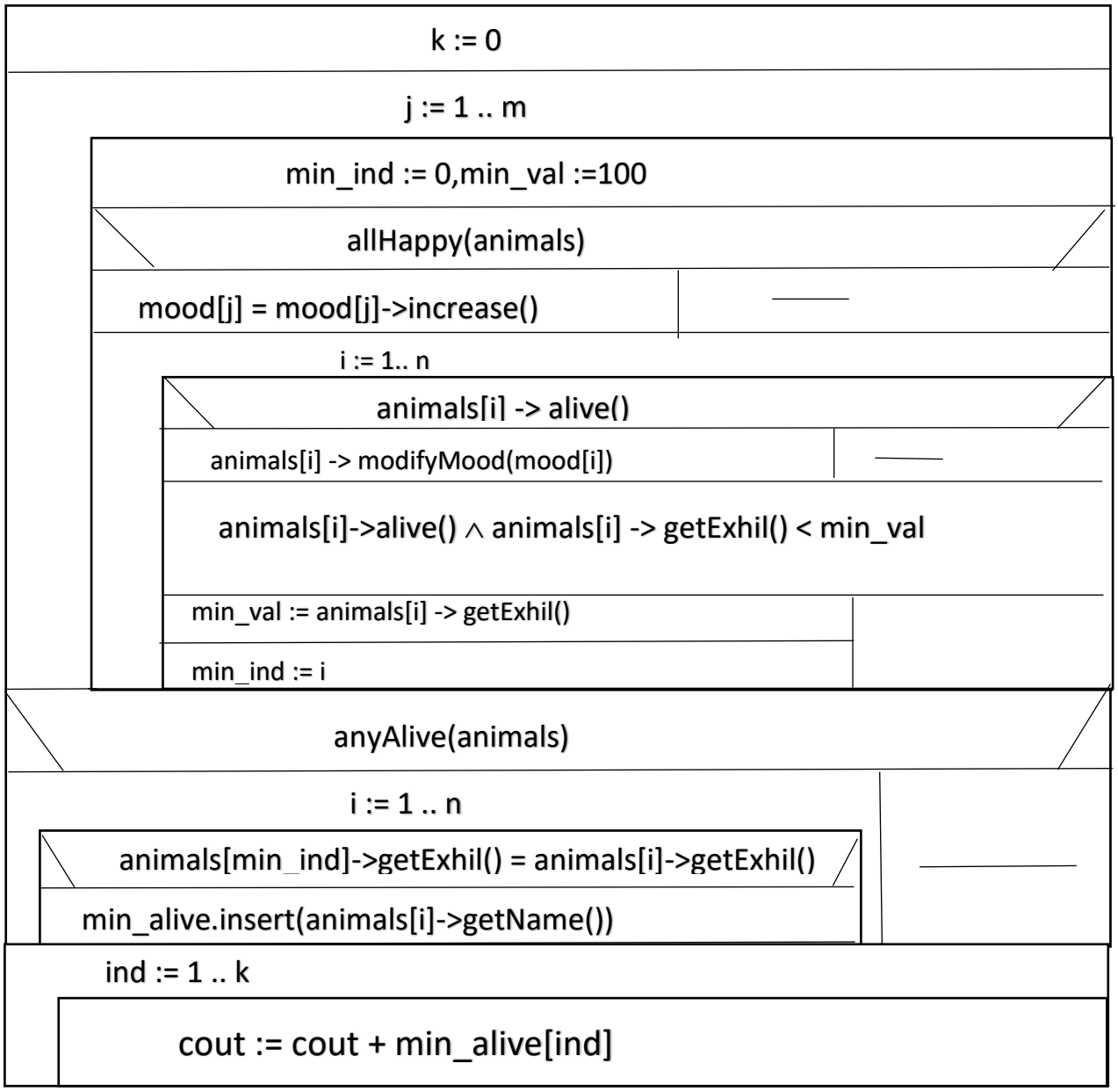
$\wedge$ *min_ind* = *M*IN$_{i=1..n}$ *animals(i).getExhil()*

min_alive = $\bigoplus_{i=1\,..\,n}$< MIN >
animals[i]->alive()

enor(E)	i = 1 .. n
f(e)	modifyMood(animals[i],mood)$_{1}$
s	animals
H,+,0	HobbyAnimal*,$\bigoplus$,<>

enor(E)	i = 1 .. n
f(e)	modifyMood(animals[i],mood)$_{2}$
s	mood
H,+,0	HobbyAnimal*,$\ominus$,<>

enor(E)	i = 1 .. n
f(e)	<animals[i]> if animals[i].alive() $\wedge$ animals[i].getExhil() < =mini
s	min_alive
H,+,0	HobbyAnimal*,$\bigoplus$,<>



Testing

Grey box test cases:

Outer loop (Summation)

1. length-based:

- zero day

- one day

- more days

2. pattern based:

- all good

- all bad

- all ordinary

Inner loop

1.length-based:

-one animal

-more animals

2. first and last:

- first animal has least exhilaration

-last animal has least exhilaration

Examination of function modifyMood()

Nine different cases depending on the animal and the mood.

Exhilaration level doesn't change if animal is dead.