10. Hobby animals need several things to preserve their exhilaration. Steve has some hobby animals: tarantulas, hamsters, and cats. Every animal has a name and their exhilaration level is between 0 and 70 (0 means that the animals dies). If their keeper is joyful, he takes care of everything to cheer up his animals, and their exhilaration level increases: of the tarantulas by 1, of the hamsters by 2, and of the cats by 3.

On a usual day, Steve takes care of only the cats (their exhilaration level increases by 3), so the exhilaration level of the rest decreases: of the tarantulas by 2, and of the hamsters by 3. On a blue day, every animal becomes a bit sadder and their exhilaration level decreases: of the tarantulas by 3, of the hamsters by 5, of the cats by 7.

Steve'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 (T – Tarantula, H – Hamster, C – Cat), name of the animal (one word), and the initial level of exhilaration.

In the last line, the daily moods of Steve are enumerated by a list of characters (j – joyful, u – usual, b – blue). The file is assumed to be correct.

List the animals of the highest exhilaration level at the end of each day.

Analysis[1]

Independent objects in the task are the animals. They can be divided into 3 different groups: Tarantulas, Hamsters and Cats. All of them have a name and an Exhilaration level that can be got. It can be examined what happens when there keeper's mood change. Steve's mood effects the animal in the following way:

If all animals exhilaration levels are greater then 5 then:

Tarantulas

mood	exhilaration change	mood change
joyfull	+1	-
usual	-2	joyfull
blue	-3	usual

Hamsters

mood	exhilaration change	mood change
joyfull	+2	-
usual	-3	joyfull
blue	-5	usual

Cats

mood	exhilaration change	mood change
joyfull	+3	-
usual	+3	joyfull
blue	-7	usual

If any animals exhilaration levels is less or equal to 5 then:

Tarantulas

mood	exhilaration change	mood change
joyfull	+1	-
usual	-2	-
blue	-3	-

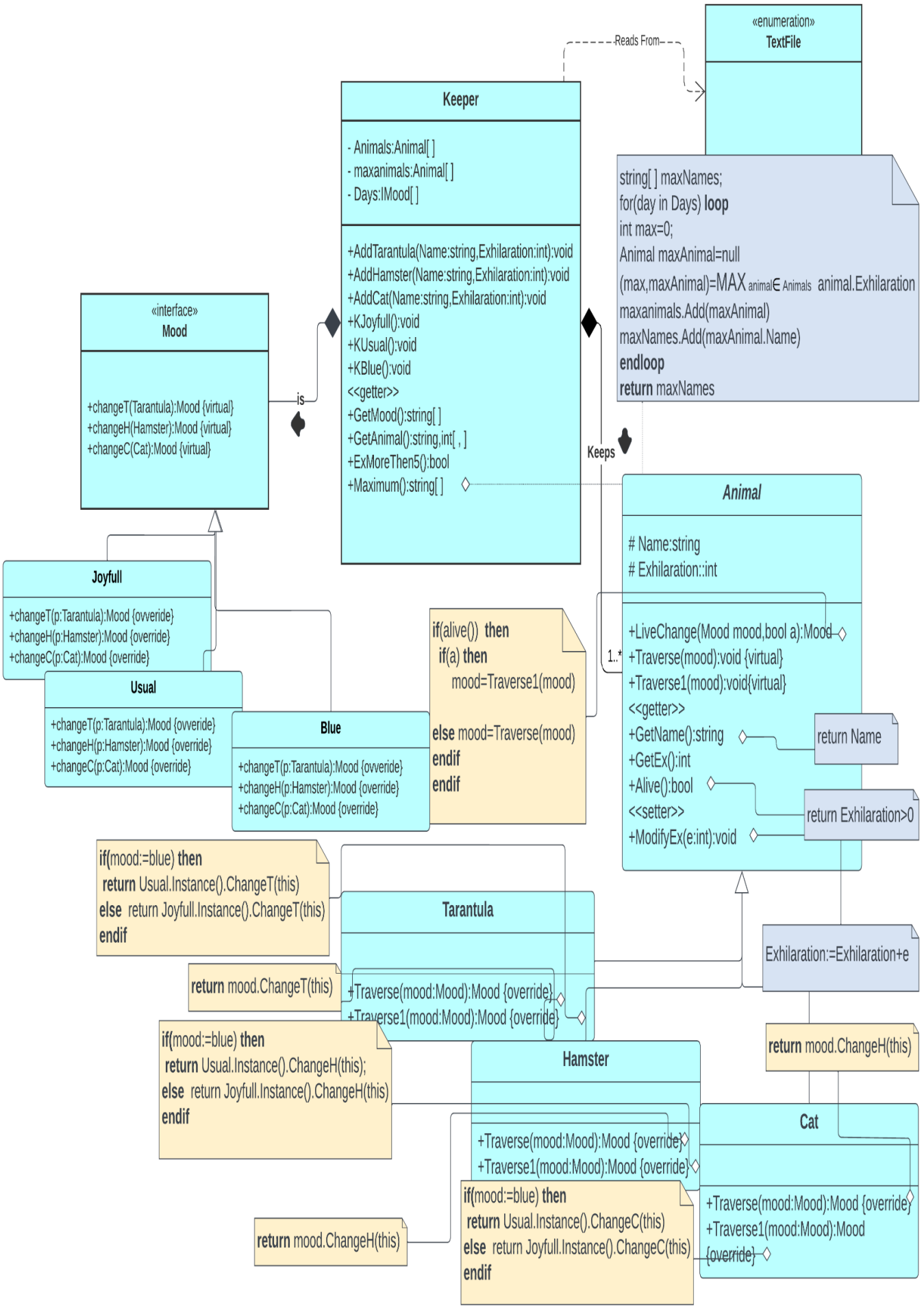
Hamsters

mood	exhilaration change	mood change
joyfull	+2	-
usual	-3	-
blue	-5	-

Cats

mood	exhilaration change	mood change
joyfull	+3	-
usual	+3	-
blue	-7	-

Plan:


«enumeration»
TextFile
Reads From
Keeper
- Animals:Animal[ ]
- maxanimals:Animal[ ]
- Days:IMood[ ]
+AddTarantula(Name:string,Exhilaration:int):void
+AddHamster(Name:string,Exhilaration:int):void
+AddCat(Name:string,Exhilaration:int):void
+KJoyfull():void
+KUsual():void
+KBlue():void
<<getter>>
+GetMood():string[ ]
+GetAnimal():string,int[,]
+ExMoreThen5():bool
+Maximum():string[ ]
string[ ] maxNames;
for(day in Days) loop
int max=0;
Animal maxAnimal=null
(max,maxAnimal)=MAX animal∈ Animals animal.Exhilaration
maxanimals.Add(maxAnimal)
maxNames.Add(maxAnimal.Name)
endloop
return maxNames
«interface»
Mood
+changeT(Tarantula):Mood {virtual}
+changeH(Hamster):Mood {virtual}
+changeC(Cat):Mood {virtual}
is
Keeps
Animal
Name:string
Exhilaration::int
+LiveChange(Mood mood,bool a):Mood
+Traverse(mood):void {virtual}
+Traverse1(mood):void{virtual}
<<getter>>
+GetName():string
+GetEx():int
+Alive():bool
<<setter>>
+ModifyEx(e:int):void
return Name
return Exhilaration>0
1..*
Joyfull
+changeT(p:Tarantula):Mood {ovveride}
+changeH(p:Hamster):Mood {override}
+changeC(p:Cat):Mood {override}
Usual
+changeT(p:Tarantula):Mood {ovveride}
+changeH(p:Hamster):Mood {override}
+changeC(p:Cat):Mood {override}
Blue
+changeT(p:Tarantula):Mood {ovveride}
+changeH(p:Hamster):Mood {override}
+changeC(p:Cat):Mood {override}
if(alive()) then
if(a) then
mood=Traverse1(mood)
else mood=Traverse(mood)
endif
endif
if(mood:=blue) then
return Usual.Instance().ChangeT(this)
else return Joyfull.Instance().ChangeT(this)
endif
Tarantula
return mood.ChangeT(this)
+Traverse(mood:Mood):Mood {override}
+Traverse1(mood:Mood):Mood {override}
Exhilaration:=Exhilaration+e
return mood.ChangeH(this)
if(mood:=blue) then
return Usual.Instance().ChangeH(this);
else return Joyfull.Instance().ChangeH(this)
endif
Hamster
+Traverse(mood:Mood):Mood {override}
+Traverse1(mood:Mood):Mood {override}
Cat
+Traverse(mood:Mood):Mood {override}
+Traverse1(mood:Mood):Mood {override}
return mood.ChangeH(this)
if(mood:=blue) then
return Usual.Instance().ChangeC(this)
else return Joyfull.Instance().ChangeC(this)
endif

Specification:

A= (Days:Mood^n, Animals:Animal^m ,max:N , maxnames:S^n ,More5:Bool)

Pre= (Days = Days$_0$ AND Animals = Animals$_0$ AND |Days|>0 AND |Animals|>0)

Post= ($\forall$i$\in$[1..n]: (More5=($\forall$j$\in$[1..m] : (Animals[j].Exhilaration>5) $\wedge$ Days[i] = animal.LiveChange(Days[i],More5) $\wedge$ (max, maxAnimal) = $MAX_{\text{animal}\in\text{Animals}}$

Alive(animal) $\wedge$ 0 ≤animal.Exhilaration ≤ 70

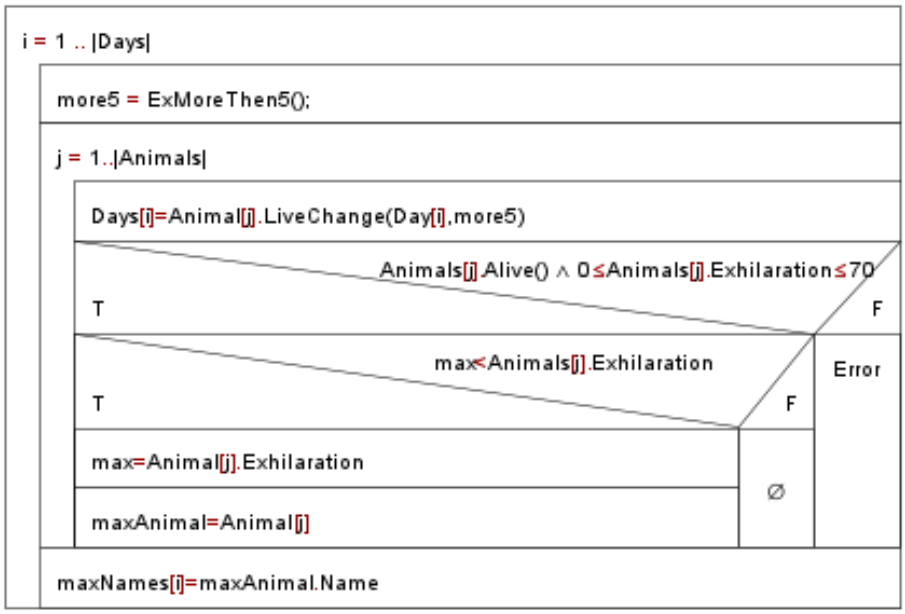
animal.Exhilaration $\wedge$ maxnames[i]=maxAnimal.Name)

Analogy:

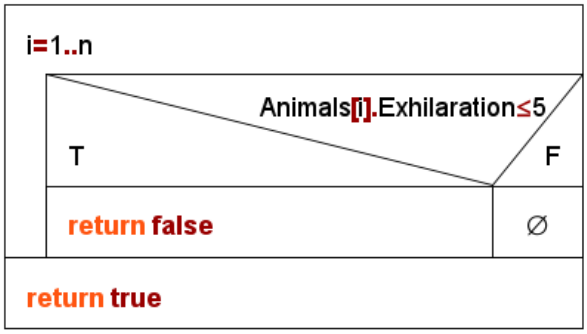
enor(E)	i=1..n
f(e)	LiveCange(Days[i],More5
s	Day
H,+,0	Mood*, ⊜, Day

enor(E)	i=1..n
f(e)	animal.Exhilaration
cond(e)	Alive(animal)
max	max
elem	maxAnimal
H, +, 0	Animal*,MAX ,Animal

Lets Merge the above to the same algorithm.



ExMoreThen5()



Testing:

Testing the operations (Black box Testing)

1. **Adding Elements**
    a. **Adding an Animal**
    b. **Adding Exhilaration of a Animal**
    c. **Adding a Mood**
    d. **Checking if the mood and animal are added**
    e. **Checking Exception if Exhilaration is more then 70 or less then 0**
2. **Checking getExhilaration and getName**

3. **Checking the Maximum Element of each day**
    a. **Adding some animals and moods**
    b. **Checking if the Maximum gets the correct element**
4. **Checking Mood Change**
    a. **Adding Animals with Ex more then 5**
    b. **Add some Moods (Other then Joyfull)**
    c. **Check mood before and after MoodChange**
5. **Checking Exceptions**
    a. **Check if there in no animal nor mood added**
    b. **Check if all Animals died**
    c. **Check for illegal input in moods and in animals**