Aliia Bazarkulova 2.assignment / 10.task June 2, 2023

HZ4BV8
hz4bv8@inf.elte.hu
Group 10

Task

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.**

Aliia Bazarkulova 2.assignment / 10.task June 2, 2023

Analysis

Independent objects in the task are the hobby 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 changed. It can be examined what happens when their keeper's mood is different. Steve's (keeper's) mood affects the hobby animals in the following way:

Tarantula:

Steve's mood	Exhilaration level change
Usual	-2
Joyful	+1
Blue	-3

Hamster:

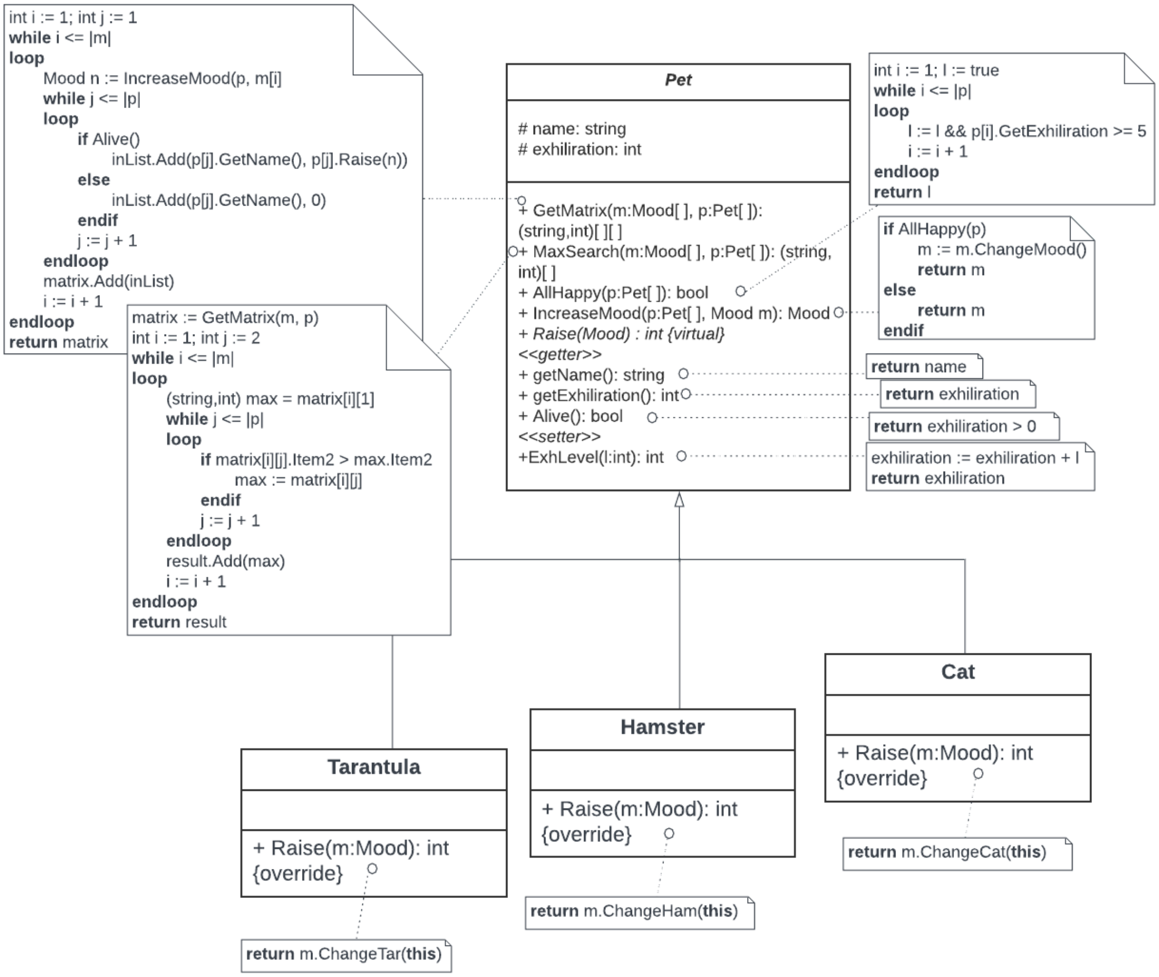
Steve's mood	Exhilaration level change
Usual	-3
Joyful	+2
Blue	-5

Cat:

Steve's mood	Exhilaration level change
Usual	+3
Joyful	+3
Blue	-7

Aliia Bazarkulova 2.assignment / 10.task June 2, 2023

Plan


Pet
# name: string
# exhiliration: int
+ GetMatrix(m:Mood[ ], p:Pet[ ]): (string,int)[ ][ ]
+ MaxSearch(m:Mood[ ], p:Pet[ ]): (string, int)[ ]
+ AllHappy(p:Pet[ ]): bool
+ IncreaseMood(p:Pet[ ], Mood m): Mood
+ Raise(Mood) : int {virtual}
<<getter>>
+ getName(): string
+ getExhiliration(): int
+ Alive(): bool
<<setter>>
+ExhLevel(l:int): int
Tarantula
+ Raise(m:Mood): int {override}
Hamster
+ Raise(m:Mood): int {override}
Cat
+ Raise(m:Mood): int {override}

```
int i := 1; int j := 1
while i <= |m|
loop
    Mood n := IncreaseMood(p, m[i]
    while j <= |p|
    loop
        if Alive()
            inList.Add(p[j].GetName(), p[j].Raise(n))
        else
            inList.Add(p[j].GetName(), 0)
        endif
        j := j + 1
    endloop
    matrix.Add(inList)
    i := i + 1
endloop
return matrix
```

```
matrix := GetMatrix(m, p)
int i := 1; int j := 2
while i <= |m|
loop
    (string,int) max = matrix[i][1]
    while j <= |p|
    loop
        if matrix[i][j].Item2 > max.Item2
            max := matrix[i][j]
        endif
        j := j + 1
    endloop
    result.Add(max)
    i := i + 1
endloop
return result
```

```
int i := 1; l := true
while i <= |p|
loop
    l := l && p[i].GetExhiliration >= 5
    i := i + 1
endloop
return l
```

```
if AllHappy(p)
    m := m.ChangeMood()
    return m
else
    return m
endif
```

```
return name
```

```
return exhiliration
```

```
return exhiliration > 0
```

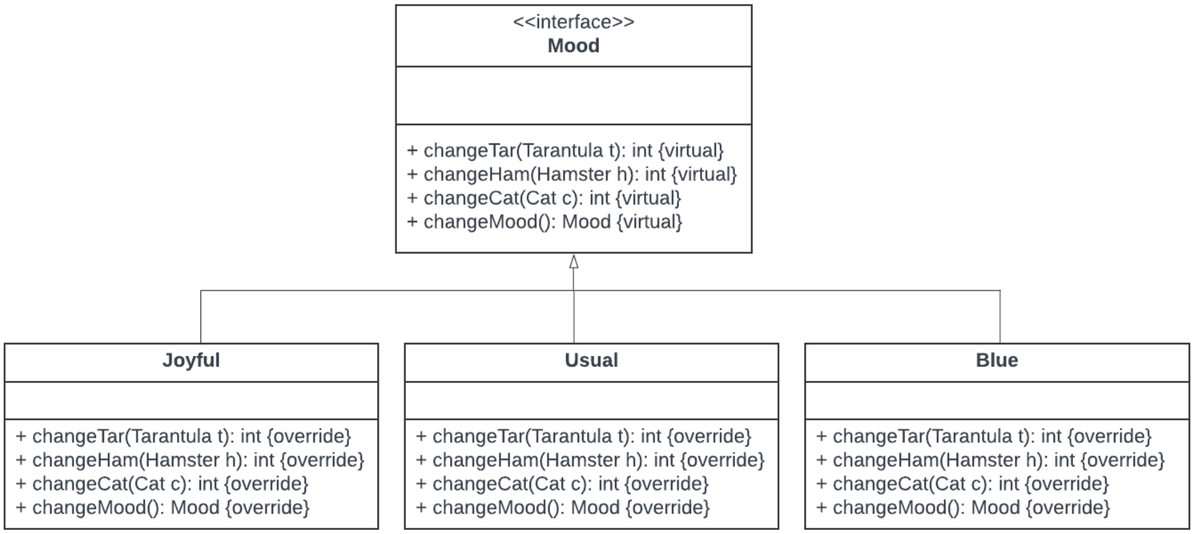
```
exhiliration := exhiliration + l
return exhiliration
```

```
return m.ChangeTar(this)
```

```
return m.ChangeHam(this)
```

```
return m.ChangeCat(this)
```

Aliia Bazarkulova 2.assignment / 10.task June 2, 2023



<<interface>>
Mood
+ changeTar(Tarantula t): int {virtual}
+ changeHam(Hamster h): int {virtual}
+ changeCat(Cat c): int {virtual}
+ changeMood(): Mood {virtual}
Joyful
+ changeTar(Tarantula t): int {override}
+ changeHam(Hamster h): int {override}
+ changeCat(Cat c): int {override}
+ changeMood(): Mood {override}
Usual
+ changeTar(Tarantula t): int {override}
+ changeHam(Hamster h): int {override}
+ changeCat(Cat c): int {override}
+ changeMood(): Mood {override}
Blue
+ changeTar(Tarantula t): int {override}
+ changeHam(Hamster h): int {override}
+ changeCat(Cat c): int {override}
+ changeMood(): Mood {override}

Methods Raise() of the concrete hobby animals expect a Mood object as an input parameter as a visitor and call the methods which correspond to the type of a pet.

The class Mood is an interface, so the methods are not implemented. All the classes of the Moods are realized based on the Singleton design pattern, as it is enough to create one object for each class.

Specification

A = moods: Mood^{m} , pets: Pet^{n} , result:$(\text{String, int})^{*}$

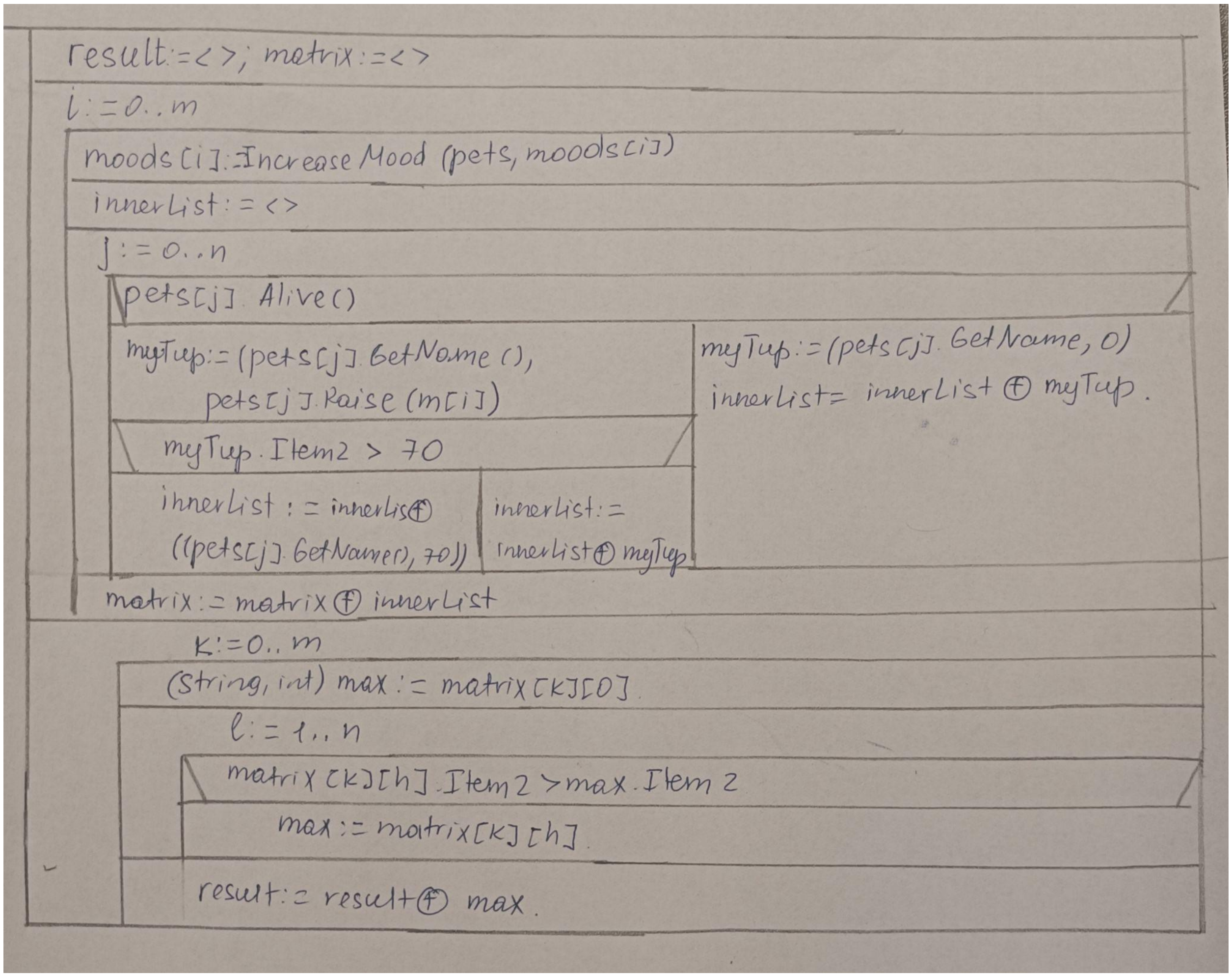
Pre = moods = moods_0 Λ pets = pets_0

Post = $\forall$ iϵ[1..m]: moods[i] = IncreaseMood(pets, moods_0[i]) Λ

$\forall$ jϵ[1..n]: pets[j].Raise(moods[i]) Λ

(elem, max) = MAX pets[j].exhilaration Λ

result = $\underset{i = 1..m}{\oplus}$ <(elem.getName(), elem.getExhilaration())>



```
result:=<>; matrix:=<>
i:=0..m
    moods[i]:=Increase Mood (pets, mooods[i])
    innerList:=<>
    j:=0..n
        pets[j].Alive()
        yes:
            myTup:=(pets[j].GetName(),
                  pets[j].Raise(m[i]))
            myTup.Item2 > 70
            yes:
                innerList:= innerlis⊕
                  ((pets[j].GetName(), 70))
            no:
                innerList:=
                  innerList⊕ myTup
        no:
            myTup:=(pets[j].GetName, 0)
            innerList= innerList ⊕ myTup.
    matrix:= matrix ⊕ innerList
k:=0..m
    (string, int) max := matrix[k][0].
    l:=1..n
        matrix[k][h].Item2 >max.Item 2
            max:= matrix[k][h].
    result:= result⊕ max.
```

Testing

1. Test Constructors, getters and setters
    a. One tarantula, one Hamster, one Cat
    b. GetName(), GetExhiliration(), ExhLevel()
2. Test Zero Pet and Zero Mood
    a. No element in pets and moods arrays
3. Test GetMatrix() (my function which helps to find the max later)
    a. Insert one tarantula, one hamster, one cat into pets and Blue Instance, Usual Instance, Joyful instance into moods
    b. Test matrix[0] and each element in it if the name is correct and exhilaration level was changed accordingly
    c. Test matrix[1] and each element in it if the name is correct and exhilaration level was changed accordingly
    d. Test matrix[2] and each element in it if the name is correct and exhilaration level was changed accordingly
4. Test only Blue mood input and the MaxSearch()
    a. Test if AllHappy() and IncreaseMood() are working correctly
5. Test only Usual mood input and the MaxSearch()
    a. Test if AllHappy() and IncreaseMood() are working correctly
6. Test only Joyful mood input and the MaxSearch()
    a. Test if AllHappy() and IncreaseMood() are working correctly
7. Test different inputs
    a. First file with 3 pets and 14 moods. Check if the first and last lines of the output are the expected values
    b. Second file with 2 pets and 15 moods. Check if the 1st, 5th, 10th and 15th lines of the output are the expected values
    c. Third file with 5 pets and 7 moods. Check if the first, middle and last lines of the output are the expected values

8. Test Exceptions
    a. Test file with 0 animals
    b. Test file with 0 moods
    c. Test empty file
    d. Test incorrect inputs for exhilaration level
    e. Test incorrect inputs for the type of mood