

ASSIGNMENT 2

To solve the following tasks, more classes have to be used which are inherited from a common base class and override the virtual methods of the base class. The objects of these classes have to be put in a collection, then the collection has to be traversed and the proper methods of the items have to be called. Prepare automatic unit tests. Divide your program to modules (compilation units). Apply exception handling. The solution has to be given based on the tools shown on the lectures. Use design patterns.

1. In the hydrological cycle of the Earth, various areas affect the weather as well as areas are also affected by various weathers. Areas involved in the simulation: plain, grassland, lakes region. Each area has a name, and the amount of water stored in the certain area is also given in km^3 . The humidity of the air over the areas is also given in percentage.

The possible types of weather are the following: sunny, cloudy, rainy, depending on the humidity of the air. In case the humidity exceeds 70%, the weather gets rainy and the humidity decreases to 30%.

In case the humidity is between 40-70%, the calculation of the chance of rainy weather is: $(\text{humidity}-40)*3,3\%$, otherwise the weather is cloudy. Humidity below 40% leads to sunny weather.

In the following, we declare how the certain areas respond to the different type of weathers.

First the amount of water stored by the area varies then the weather will be affected. There is no type of areas with negative amount of water stored.

In case the type is *plain*, if the weather is sunny, the amount of water will be decreased by 3 km^3 ; if cloudy, it will be decreased by 1 km^3 ; for rainy weather it will be increased by 20 km^3 . The humidity of the air is increased by 5%. If the amount of the stored water is greater than 15 km^3 , the plain area changes into grassland.

In case of type *grassland*: in sunny weather, the amount of water is decreased by 6 km^3 , for cloudy it will be decreased by 2 km^3 , but and for rainy, it will be increased by 15 km^3 . The humidity of the air is increased by 10%. The area becomes lakes region obtaining amount of water over 50 km^3 , whereas in case the amount of stored water goes below 16 km^3 , the area changes to plain.

In case of type *lakes region*: in sunny weather, the amount of water is decreased by 10 km^3 , for cloudy it will be decreased by 3 km^3 , for rainy it will be increased by 20 km^3 . The humidity will be increased by 15%. Beyond an amount of water of 51 km^3 the area changes into grassland.

The program reads data from a text file. The first line of the file contains a single integer N indicating the number of areas. Each of the following N lines contains the attributes of an area separated by spaces: the owner of the area, the type of the area, and the amount of water stored by the area. In the last line, the humidity of the air is given in percentage. The type is identified by a character: P – plain, G – grassland, L – lakes region.

After 10 simulation rounds, determine the owner of the area which is storing the greatest amount of water. The amount of water is also required to be determined. The program should print all attributes of the certain areas by simulation rounds!

The program should ask for a filename, then print the content of the input file. You can assume that the input file is correct. Sample input:

```
4
Mr Bean L 86
Mr Green G 26
Mr Dean P 12
Mr Teen G 35
98
```

2. In the hydrological cycle of the Earth, various areas affect the weather as well as areas are also affected by various weathers. Areas involved in the simulation: plain, grassland, lakes region. Each area has a name, and the amount of water stored in the certain area is also given in km^3 . The humidity of the air over the areas is also given in percentage.

The possible types of weather are the following: sunny, cloudy, rainy, depending on the humidity of the air. In case the humidity exceeds 70%, the weather gets rainy and the humidity decreases to 30%.

In case the humidity is between 40-70%, the calculation of the chance of rainy weather is: $(\text{humidity}-30)*3,3\%$, otherwise the weather is cloudy. Humidity below 40% leads to sunny weather.

In the following, we declare how the certain areas respond to the different type of weathers.

First the amount of water stored by the area varies then the weather will be affected. There is no type of areas with negative amount of water stored.

In case the type is *plain*, if the weather is sunny, the amount of water will be decreased by 3 km^3 ; if cloudy, it will be decreased by 1 km^3 ; for rainy weather it will be increased by 20 km^3 . The humidity of the air is increased by 5%. If the amount of the stored water is greater than 15 km^3 , the plain area changes into grassland.

In case of type *grassland*: in sunny weather, the amount of water is decreased by 6 km^3 , for cloudy it will be decreased by 2 km^3 , but and for rainy, it will be increased by 15 km^3 . The humidity of the air is increased by 10%. The area becomes lakes region obtaining amount of water over 50 km^3 , whereas in case the amount of stored water goes below 16 km^3 , the area changes to plain.

In case of type *lakes region*: in sunny weather, the amount of water is decreased by 10 km^3 , for cloudy it will be decreased by 3 km^3 , for rainy it will be increased by 20 km^3 . The humidity will be increased by 15%. Beyond an amount of water of 51 km^3 the area changes into grassland.

The program reads data from a text file. The first line of the file contains a single integer N indicating the number of areas. Each of the following N lines contains the attributes of an area separated by spaces: the owner of the area, the type of the area, and the amount of water stored by the area. In the last line, the humidity of the air is given in percentage. The type is identified by a character: P – plain, G – grassland, L – lakes region.

We continue the simulation until each area has the same type. The program should print all attributes of the certain areas by simulation rounds!

The program should ask for a filename, then print the content of the input file. You can assume that the input file is correct. Sample input:

```
4
Mr Bean L 86
Mr Green G 26
Mr Dean P 12
Mr Teen G 35
98
```

3. Layers of gases are given, with certain type (ozone, oxygen, carbon dioxide) and thickness, affected by atmospheric variables (thunderstorm, sunshine, other effects). When a part of one layer changes into another layer due to an atmospheric variable, the newly transformed layer ascends and engrosses the first identical type of layer of gases over it. In case there is no identical layer above, it creates a new layer on the top of the atmosphere. In the following we declare, how the different types of layers react to the different variables by changing their type and thickness.

No layer can have a thickness less than 0.5 km, unless it ascends to the identical-type upper layer. In case there is no identical one, the layer perishes.

	thunderstorm	sunshine	other
ozone	-	-	5% turns to oxygen
oxygen	50% turns to ozone	5% turns to ozone	10% turns to carbon dioxide
carbon dioxide	-	5% turns to oxygen	-

The program reads data from a text file. The first line of the file contains a single integer N indicating the number of layers. Each of the following N lines contains the attributes of a layer separated by spaces: type and thickness. The type is identified by a character: Z – ozone, X – oxygen, C – carbon dioxide.

The last line of the file represents the atmospheric variables in the form of a sequence of characters: T – thunderstorm, S – sunshine, O – others. In case the simulation is over, it continues from the beginning.

The program should continue the simulation until one gas component totally perishes from the atmosphere. The program should print all attributes of the layers by simulation rounds!

The program should ask for a filename, then print the content of the input file. You can assume that the input file is correct. Sample input:

```
4
Z 5
X 0.8
C 3
X 4
OOOOSSTSSOO
```

4. Layers of gases are given, with certain type (ozone, oxygen, carbon dioxide) and thickness, affected by atmospheric variables (thunderstorm, sunshine, other effects). When a part of one layer changes into another layer due to an atmospheric variable, the newly transformed layer ascends and engrosses the first identical type of layer of gases over it. In case there is no identical layer above, it creates a new layer on the top of the atmosphere. In the following we declare, how the different types of layers react to the different variables by changing their type and thickness.

No layer can have a thickness less than 0.5 km, unless it ascends to the identical-type upper layer. In case there is no identical one, the layer perishes.

	thunderstorm	sunshine	other
ozone	-	-	5% turns to oxygen
oxygen	50% turns to ozone	5% turns to ozone	10% turns to carbon dioxide
carbon dioxide	-	5% turns to oxygen	-

The program reads data from a text file. The first line of the file contains a single integer N indicating the number of layers. Each of the following N lines contains the attributes of a layer separated by spaces: type and thickness. The type is identified by a character: Z – ozone, X – oxygen, C – carbon dioxide.

The last line of the file represents the atmospheric variables in the form of a sequence of characters: T – thunderstorm, S – sunshine, O – others. In case the simulation is over, it continues from the beginning.

The program should continue the simulation until the number of layers is the triple of the initial number of layers or is less than three. The program should print all attributes of the layers by simulation rounds!

The program should ask for a filename, then print the content of the input file. You can assume that the input file is correct. Sample input:

```
4
Z 5
X 0.8
C 3
X 4
OOOOSSTSSOO
```

5. We are simulating the animals of the tundra. There are colonies of prey and predator animals. The number of animals in a colony affect the number of animals in other colonies. There are three predator species: the snowy owl, the arctic fox and the wolf. There are three kinds of prey: the lemming, the arctic hare and the gopher.

If the number of prey animals increase, predators can reproduce more quickly. If the number of prey is very large, most of them will wander away because they cannot find enough food. If the number of predators is large, the number of the prey decreases quicker as they are preyed upon.

Each colony has a name, a species, and the number of animals in the colony. The prey species are affected by the different predator species as follows. The number of animals in their own colony changes first, then they influence the predators.

Lemming: If they are preyed upon by a predator colony, the number of animals in their colony decreases by four times the number of animals in the predator colony. The number of animals in their colony doubles every second turn. If there are more than 200 animals in the colony, the number of animals in the colony decreases to 30.

Hare: If they are preyed upon by a predator colony, the number of animals in their colony decreases by double the number of animals in the predator colony. The number of animals in their colony grows by 50 percent (to one and a half times their previous number) every second turn. If there are more than 100 animals in the colony, the number of animals in the colony decreases to 20.

Gopher: If they are preyed upon by a predator colony, the number of animals in their colony decreases by double the number of animals in the predator colony. The number of animals in their colony doubles every fourth turn. If there are more than 200 animals in the colony, the number of animals in the colony decreases to 40.

Predators choose and attack a prey colony randomly in each turn. If there are not enough animals in the attacked colony (for example, there are not four times the number of predators in a lemming colony), the number of predators also decreases: every fourth predator out of the ones who didn't get prey perishes. Predators have offsprings every eighth turn. Normally, the snow owls have 1 offspring per 4 animals, the foxes have 3 offsprings per 4 animals, and the wolves have 2 offsprings per 4 animals.

The program should read the colonies from a text file. The first line contains the number of prey and predator colonies separated by a space. Each of the next lines contains the data of one colony separated by space: their name, their species, their starting number of animals. The species can be: o - owl, f - fox, w - wolf, l - lemming, h - hare, g - gopher.

Simulate the process until the number of animals in each predator colony decreases to below 4, or the number of predators doubles compared to its starting value. Print the data of each colony in each turn.

The program should ask for the name of the input file and display its contents. You can assume that the input file is correct. A possible input:

```
3 3
lem1 l 86
lem2 l 90
hare1 h 26
go g 12
hungry w 12
feathery o 6
```

6. We are simulating the animals of the tundra. There are colonies of prey and predator animals. The number of animals in a colony affect the number of animals in other colonies. There are three predator species: the snowy owl, the arctic fox and the wolf. There are three kinds of prey: the lemming, the arctic hare and the gopher.

If the number of prey animals increase, predators can reproduce more quickly. If the number of prey is very large, most of them will wander away because they cannot find enough food. If the number of predators is large, the number of the prey decreases quicker as they are preyed upon.

Each colony has a name, a species, and the number of animals in the colony. The prey species are affected by the different predator species as follows. The number of animals in their own colony changes first, then they influence the predators.

Lemming: If they are preyed upon by a predator colony, the number of animals in their colony decreases by four times the number of animals in the predator colony. The number of animals in their colony doubles every second turn. If there are more than 200 animals in the colony, the number of animals in the colony decreases to 30.

Hare: If they are preyed upon by a predator colony, the number of animals in their colony decreases by double the number of animals in the predator colony. The number of animals in their colony grows by 50 percent (to one and a half times their previous number) every second turn. If there are more than 100 animals in the colony, the number of animals in the colony decreases to 20.

Gopher: If they are preyed upon by a predator colony, the number of animals in their colony decreases by double the number of animals in the predator colony. The number of animals in their colony doubles every fourth turn. If there are more than 200 animals in the colony, the number of animals in the colony decreases to 40.

Predators choose and attack a prey colony randomly in each turn. If there are not enough animals in the attacked colony (for example, there are not four times the number of predators in a lemming colony), the number of predators also decreases: every fourth predator out of the ones who didn't get prey perishes. Predators have offsprings every eighth turn. Normally, the snow owls have 1 offspring per 4 animals, the foxes have 3 offsprings per 4 animals, and the wolves have 2 offsprings per 4 animals.

The program should read the colonies from a text file. The first line contains the number of prey and predator colonies separated by a space. Each of the next lines contains the data of one colony separated by space: their name, their species, their starting number of animals. The species can be: o - owl, f - fox, w - wolf, l - lemming, h - hare, g - gopher.

Simulate the process until each of the prey colonies becomes extinct or the number of prey animals quadruples compared to its starting value. Print the data of each colony in each turn.

The program should ask for the name of the input file and display its contents. You can assume that the input file is correct. A possible input:

```
3 3
lem1 l 86
lem2 l 90
hare1 h 26
go g 12
hungry w 12
feathery o 6
```

7. Different kinds of plants live on a planet. If the nutrient of a plant runs out (its nutrient level becomes zero), the plant wastes away. There are three kinds of radiation on the planet: alpha, delta, no radiation. The different species of plants react to radiation differently. The reaction involves a change in the nutrient level of the plant and the radiation the next day. The radiation of the next day will be alpha radiation if the sum of the demand for alpha radiation over all plants is greater than the sum of the demand for delta radiation by at least three. If the demand for delta radiation is greater by at least three than the demand for alpha radiation, the radiation will be delta. If the difference is less than three, there will be no radiation. There is no radiation the first day.

Each plant has a name (string), a nutrient level (int), and a boolean that denotes whether it's alive. The plant species are wombleroot, wittentoot and woreroot. The different plant species react to the different radiations as follows. The level of nutrients changes first. After that, the plant can influence the radiation of the next day if it's still alive.

Wombleroot: Alpha radiation makes the nutrient level increase by 2, no radiation makes it decrease by 1, and delta radiation makes it decrease by 2. It demands alpha radiation by a strength of 10 regardless of the current radiation. This plant also wastes away if its nutrient level increases above 10.

Wittentoot: Alpha radiation makes the nutrient level decrease by 3, no radiation makes it decrease by 1, delta radiation makes it increase by 4. This plant demands delta radiation with strength 4 if its nutrient level is less than 5, with strength 1 if its nutrient level is between 5 and 10, and doesn't influence the radiation if its nutrient level is greater than 10.

Woreroot: Its nutrient level increases by 1 if there is alpha or delta radiation, and decreases by 1 if there is no radiation. Doesn't influence the radiation of the next day.

Simulate the ecosystem of plants and give the name of the strongest plant which is still alive after n days. Print all the data of the plants and the level of radiation on each day.

The program should read the data of the simulation from a text file. The first line contains the number of plants. Each of the next lines contains the data of one plant: its name, its species, and its starting nutrient level. The species can be: wom - wombleroot, wit - wittentoot, wor - woreroot. The last line of the file contains n, the number of days as an int. The program should ask for the filename and display the contents of the file. You can assume that the input file is correct. A possible input file:

```
4
Hungry wom 7
Lanky wit 5
Big wor 4
Tall wit 3
10
```

8. Different kinds of plants live on a planet. If the nutrient of a plant runs out (its nutrient level becomes zero), the plant wastes away. There are three kinds of radiation on the planet: alpha, delta, no radiation. The different species of plants react to radiation differently. The reaction involves a change in the nutrient level of the plant and the radiation the next day. The radiation of the next day will be alpha radiation if the sum of the demand for alpha radiation over all plants is greater than the sum of the demand for delta radiation by at least three. If the demand for delta radiation is greater by at least three than the demand for alpha radiation, the radiation will be delta. If the difference is less than three, there will be no radiation. There is no radiation the first day.

Each plant has a name (string), a nutrient level (int), and a boolean that denotes whether it's alive. The plant species are wombleroot, wittentoot and woreroot. The different plant species react to the different radiations as follows. The level of nutrients changes first. After that, the plant can influence the radiation of the next day if it's still alive.

Wombleroot: Alpha radiation makes the nutrient level increase by 2, no radiation makes it decrease by 1, and delta radiation makes it decrease by 2. It demands alpha radiation by a strength of 10 regardless of the current radiation. This plant also wastes away if its nutrient level increases above 10.

Wittentoot: Alpha radiation makes the nutrient level decrease by 3, no radiation makes it decrease by 1, delta radiation makes it increase by 4. This plant demands delta radiation with strength 4 if its nutrient level is less than 5, with strength 1 if its nutrient level is between 5 and 10, and doesn't influence the radiation if its nutrient level is greater than 10.

Woreroot: Its nutrient level increases by 1 if there is alpha or delta radiation, and decreases by 1 if there is no radiation. Doesn't influence the radiation of the next day.

Simulate the ecosystem of plants until there is no radiation on two consecutive days. Print all the data of the plants and the level of radiation on each day.

The program should read the data of the simulation from a text file. The first line contains the number of plants. Each of the next lines contains the data of one plant: its name, its species, and its starting nutrient level. The species can be: wom - wombleroot, wit - wittentoot, wor - woreroot. The program should ask for the filename and display the contents of the file. You can assume that the input file is correct. A possible input file:

```
4
Hungry wom 7
Lanky wit 5
Big wor 4
Tall wit 3
```


9. 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 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 (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!

A possible input file:

```
3
F Nemo 50
B Hedwig 70
D Lassie 20
ooooggbgbgoogg
```

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.

A possible input file:

```
3
T Webster 20
H Butterscotch 30
C Cat-man-do 50
uuuujjbjbjjuujj
```