

Chicken Barn - OOP Homework

The problem (classic version)

The chickens

- In a barn **chickens** are bought from a market.
- The number of chickens and their age is configured in an input file.
 - Read the initial information about the chickens from the console or a file (.txt, .xml, .json, .xls – your choice).
 - The file consists of each chicken's age. The number of inputs determine the number of chickens.
 - Example csv input (initial.chicken.ages.csv):

```
initialChickenAgesInWeeks
1
0
1
```

The csv states the barn starts with three chickens - two are 1 week old, one has just hatched (zero weeks old).

- Each chicken lays **0-3 eggs per week depending on the week**
 - A week can be **bad, normal** or **good**
 - a chicken lays 0, 1 or 3 eggs accordingly (bad - 0 eggs, normal - 1 , good - 3)
 - each chicken can lay eggs only if it is **2-4 weeks old** (inclusively - they hatch on the fourth week too)
- Chickens **die at 6 weeks old**.

After 2 weeks the eggs hatch and become chickens.

The weeks

Read the initial information about the "weather" (good or bad weeks) from the console or a file (.txt, .xml, .json, .xls – your choice). The file consists of each week's type (bad, normal, good). The number of week types determine how long we simulate the barn. Example simulation parameters (weeks.csv):

```
WeekType
good
bad
normal
good
normal
good
normal
```

The config specifies 7 weeks, starts with a **good** week and ends with a normal **week**.

The simulation

Run the simulation for as many weeks as are provided in a config file. Output all live chickens and current number of eggs. If you can, print the parent (mother only) of every chicken and how old it is in curly brackets after the name (put **x** if chicken has passed (died)). Example:

- **Chicken0{1}**/ - chicken with index 0 and age 1
- **Chicken1{0}**/ - chicken with index 1 and age 0 (just hatched)
- **Chicken0{x}/Chicken3{4}/Chicken15{0}** - chicken 15 (age 0 - just hatched), child of chicken 3 (age 4), child of chicken 0 (dead)

Chickens are named with a number starting from 0. The three market chickens are 0, 1 and 2 respectively.

Output file

Each chicken and its children as well as the leftover eggs should be printed on the console. After you print the result, ask the user if he wants to save the information in a file. If so, save the information in human-readable format (text based, xml, json).

Example output

This output is based on 3 market chickens (ages 1, 0 and 1 weeks accordingly) and 0 initial eggs.

0-week simulation (basically what we bought from the market, no time has passed)

```
Chicken0{1}
Chicken1{0}
Chicken2{1}
Eggs: 0
```

1-week simulation

```
Chicken0{2}
Chicken1{1}
Chicken2{2}
Eggs: 6
```

3-week simulation - good, bad, normal

```
Chicken0{4}
Chicken1{3}
Chicken2{4}
Chicken0{4}/Chicken3{0}
Chicken0{4}/Chicken4{0}
```

```
Chicken0{4}/Chicken5{0}  
Chicken2{4}/Chicken6{0}  
Chicken2{4}/Chicken7{0}  
Chicken2{4}/Chicken8{0}  
Eggs: 3
```

7-week simulation

Runs for 7 weeks - good, bad, normal, good, normal, good, normal.

```
Chicken0{x}/Chicken3{4}  
Chicken0{x}/Chicken4{4}  
Chicken0{x}/Chicken5{4}  
Chicken2{x}/Chicken6{4}  
Chicken2{x}/Chicken7{4}  
Chicken2{x}/Chicken8{4}  
Chicken0{x}/Chicken9{2}  
Chicken1{x}/Chicken10{2}  
Chicken2{x}/Chicken11{2}  
Chicken1{x}/Chicken12{1}  
Chicken1{x}/Chicken13{1}  
Chicken1{x}/Chicken14{1}  
Chicken0{x}/Chicken3{4}/Chicken15{0}  
Chicken0{x}/Chicken4{4}/Chicken16{0}  
Chicken0{x}/Chicken5{4}/Chicken17{0}  
Chicken2{x}/Chicken6{4}/Chicken18{0}  
Chicken2{x}/Chicken7{4}/Chicken19{0}  
Chicken2{x}/Chicken8{4}/Chicken20{0}  
Eggs: 27
```