

Group Project 2 – Factory Simulation

The project can be done in a group of ≤ 4 students. Each group must do the project by themselves. **Everyone involved in cheating will get ZERO point**

1. This project uses 1 input file (**config.txt**). The first column of each line indicates the type of input data

1.1 Line 1 starts with D, followed by days of simulation

1.2 Line 2 starts with M, followed by material names

1.3 Next few lines start with S, followed by supplier name and daily supply rate of each material. *For example, Supplier 1 will supply 100 buttons + 100 zippers in each day*

1.4 Next few lines start with F, followed by factory name, product name, daily lot size, and materials required for 1 product. *For example, Factory 1 will try to produce 1 lot of 10 handbags in each day. Each handbag requires 2 button + 6 zippers (so its daily requirement = 20 buttons + 60 zippers)*

config.txt

```
D, 6
M, buttons, zippers
S, Supplier 1, 100, 100
S, Supplier 2, 50, 50
F, Factory 1, Handbags, 10, 2, 6
F, Factory 2, Jackets, 20, 8, 1
F, Factory 3, Pants, 40, 3, 3
```

** Don't hard code these values. I may change some of them to check whether your calculation is correct

- Line order is always as stated in 1.1-1.4 (i.e. lines D, M, a few S's, and a few F's)
- Number of days, supplier rates, factory rates may be changed
- Material names, product names, supplier names, factory names may be changed
- Number of materials, suppliers, and factories may be changed. But #supplied materials in line S and #required materials in line F will always match #materials in line M (as highlighted)
- There won't be any input error (e.g. invalid input, negative number, wrong format, missing columns) in this file. But the program must still handle the case of missing file. Don't let it crash

2. Implement class **Material** that represent each material. Your class should have

- Variable to keep material balance
- Methods to add materials (by suppliers and factories) and retrieve materials (by factories)
- All suppliers and factories must access the same set of material objects

3. Implement class **SupplierThread** that represent each supplier as thread. Thread activities are done in loop. Each iteration of the loop = 1 day. In each day:

- 3.1 Wait until one thread (main, SupplierThread, or FactoryThread) prints day number
- 3.2 Add materials and update material balances. The amount to add depends = its daily supply rates. Print thread activities as in the demo

4. Implement class **FactoryThread** that represent each factory as thread. Your class should have a variable to count product lots. Thread activities are done in loop. Each iteration of the loop = 1 day. In each day:

- 4.1 Wait until one thread (can be main, SupplierThread, or FactoryThread) prints day number and all SupplierThreads finish adding materials
- 4.2 Report materials it is holding from yesterday. All FactoryThreads must wait until all of them finish printing reports, then proceed to 4.3

- 4.3 Retrieve remaining materials to produce 1 lot of products and update material balances. The amount to retrieve = its daily requirement rates. *For example, if Factory 1 is already holding 20 buttons, it only needs to get 60 zippers today.* If the material balance is not enough, it will take as much as it can. *For example, if there are only 50 zippers, it will still take them.* Print thread activities as in the demo
- 4.4 Wait until all FactoryThreads finish retrieving materials, then check whether it can make 1 full lot of products by using the materials it has from 4.2 + 4.3
 - If 1 full lot can be made → update lot count and clear all materials (i.e. they are all used up)
 - If 1 full lot can't be made → keep only materials it has in full amount for tomorrow. Return materials it doesn't have in full amount (i.e. updating material balance)
 - *Example 1: if Factory 1 has 20 buttons + 60 zippers, it will make 1 lot of handbags, hence holding 0 buttons + 0 zippers for tomorrow*
 - *Example 2: if Factory 1 has 20 buttons + 50 zippers, it can't make handbags today. It will keep 20 buttons and return 50 zippers, hence holding 20 buttons + 0 zippers for tomorrow*
 - Print thread activities as in the demo

5. Implement main class with main method

- 5.1 Read data from config.txt
- 5.2 Create Materials, SupplierThreads, FactoryThreads. Start all threads. You are recommended to use ArrayLists to keep Materials, SupplierThreads, FactoryThreads for flexibility
- 5.3 After all threads complete all days of simulation, let main thread report number of lots made by all FactoryThreads, sorted in decreasing order of lots. If lot counts are equal, use alphabetical sorting of product names.

**** Everything printed to the screen must be labelled by the name of the thread who prints it. Don't hard code thread's name but use `Thread.currentThread().getName()`**

6. Package and folder structure must be correct

- 6.1 Your source files (.java) must be in folder Project2_XXX where XXX = ID of the group representative, assuming that this folder is under Maven's "src/main/java" structure. The first lines of all source files must be comments containing names & IDs of all members.
- 6.2 Input files must be read from Project2_XXX. Don't use absolute path that is valid only on your PC.
- 6.3 Add readme.txt containing names & IDs of all members in Project2_XXX.

Submission

1. Group representative zips and submits Project2_XXX to Google classroom
2. Other members submit only readme.txt to Google classroom

Grading

- | | |
|----------|---|
| 1 point | correct steps + results by SupplierThread (adding materials) |
| 3 points | correct steps + results by FactoryThread (holding, retrieving, returning materials) |
| 1 point | correct summary by main thread (sorted) |
| 1 point | other requirements (thread names, missing file handling) |
| 4 points | design & programming in proper OOP and multithreading style |

```

--- exec-maven-plugin:3.0.0:exec (default-cli) @ projectsol_2 ---
java.io.FileNotFoundException: src\main\java\Project2\configs.txt (The system cannot find the file specified)
Thread main >> Enter config file for simulation =
config
java.io.FileNotFoundException: src\main\java\Project2\config (The system cannot find the file specified)
Thread main >> Enter config file for simulation =
config.txt
main      >> read configs from src/main/java/Project2/config.txt

main      >> simulation days = 6
main      >> Supplier 1 daily supply rates = 100 buttons    100 zippers
main      >> Supplier 2 daily supply rates = 50 buttons     50 zippers
main      >> Factory 1  daily use  rates = 20 buttons        60 zippers    producing 10 Handbags
main      >> Factory 2  daily use  rates = 160 buttons       20 zippers    producing 20 Jackets
main      >> Factory 3  daily use  rates = 120 buttons       120 zippers   producing 40 Pants

```

Handle missing file

```

main      >> -----
main      >> Day 1
Supplier 2 >> Put      50 buttons      balance = 50 buttons
Supplier 2 >> Put      50 zippers      balance = 50 zippers
Supplier 1 >> Put     100 buttons      balance = 150 buttons
Supplier 1 >> Put     100 zippers      balance = 150 zippers
Factory 2  >> Holding  0 buttons      0 zippers
Factory 1  >> Holding  0 buttons      0 zippers
Factory 3  >> Holding  0 buttons      0 zippers
Factory 3  >> Get     120 buttons      balance = 30 buttons
Factory 3  >> Get     120 zippers      balance = 30 zippers
Factory 1  >> Get      20 buttons      balance = 10 buttons
Factory 1  >> Get      30 zippers      balance = 0 zippers
Factory 2  >> Get      10 buttons      balance = 0 buttons
Factory 2  >> Get       0 zippers      balance = 0 zippers
Factory 2  >> Jackets production fails
Factory 2  >> Put      10 buttons      balance = 10 buttons
Factory 1  >> Handbags production fails
Factory 1  >> Put      30 zippers      balance = 30 zippers
Factory 3  >> Pants  production succeeds, lot 1

```

config.txt

Demo 1

D, 6
 M, buttons, zippers
 S, Supplier 1, 100, 100
 S, Supplier 2, 50, 50
 F, Factory 1, Handbags, 10, 2, 6
 F, Factory 2, Jackets, 20, 8, 1
 F, Factory 3, Pants, 40, 3, 3

```

main      >> -----
main      >> Day 2
Supplier 1 >> Put     100 buttons      balance = 110 buttons
Supplier 1 >> Put     100 zippers      balance = 130 zippers
Supplier 2 >> Put      50 buttons      balance = 160 buttons
Supplier 2 >> Put      50 zippers      balance = 180 zippers
Factory 3  >> Holding  0 buttons      0 zippers
Factory 2  >> Holding  0 buttons      0 zippers
Factory 1  >> Holding  20 buttons      0 zippers
Factory 1  >> Get      60 zippers      balance = 120 zippers
Factory 3  >> Get     120 buttons      balance = 40 buttons
Factory 3  >> Get     120 zippers      balance = 0 zippers
Factory 2  >> Get      40 buttons      balance = 0 buttons
Factory 2  >> Get       0 zippers      balance = 0 zippers
Factory 2  >> Jackets production fails
Factory 2  >> Put      40 buttons      balance = 40 buttons
Factory 1  >> Handbags production succeeds, lot 1
Factory 3  >> Pants  production succeeds, lot 2

```

10 buttons from yesterday

30 zippers from yesterday

Factory 1 is holding 20 buttons from yesterday.

It will try to get only zippers today

get 0 zippers, so no need to return zippers

```

main      >> -----
main      >> Day 3
Supplier 2 >> Put      50 buttons      balance = 90 buttons
Supplier 2 >> Put      50 zippers      balance = 50 zippers
Supplier 1 >> Put     100 buttons      balance = 190 buttons
Supplier 1 >> Put     100 zippers      balance = 150 zippers
Factory 3  >> Holding  0 buttons      0 zippers
Factory 2  >> Holding  0 buttons      0 zippers
Factory 1  >> Holding  0 buttons      0 zippers
Factory 1  >> Get      20 buttons      balance = 170 buttons
Factory 1  >> Get      60 zippers      balance = 90 zippers
Factory 2  >> Get     160 buttons      balance = 10 buttons
Factory 2  >> Get      20 zippers      balance = 70 zippers
Factory 3  >> Get      10 buttons      balance = 0 buttons
Factory 3  >> Get      70 zippers      balance = 0 zippers
Factory 3  >> Pants  production fails
Factory 3  >> Put      10 buttons      balance = 10 buttons
Factory 3  >> Put      70 zippers      balance = 70 zippers
Factory 1  >> Handbags production succeeds, lot 2
Factory 2  >> Jackets production succeeds, lot 1

```

```

main      >> -----
main      >> Day 4
Supplier 1 >> Put    100 buttons      balance = 110 buttons
Supplier 1 >> Put    100 zippers     balance = 170 zippers
Supplier 2 >> Put    50 buttons      balance = 160 buttons
Supplier 2 >> Put    50 zippers     balance = 220 zippers
Factory 1  >> Holding 0 buttons      0 zippers
Factory 2  >> Holding 0 buttons      0 zippers
Factory 3  >> Holding 0 buttons      0 zippers
Factory 3  >> Get     120 buttons     balance = 40 buttons
Factory 3  >> Get     120 zippers     balance = 100 zippers
Factory 2  >> Get     40 buttons     balance = 0 buttons
Factory 2  >> Get     20 zippers     balance = 80 zippers
Factory 1  >> Get      0 buttons     balance = 0 buttons
Factory 1  >> Get     60 zippers     balance = 20 zippers
Factory 1  >> Handbags production fails
Factory 3  >> Pants   production succeeds, lot 3
Factory 2  >> Jackets production fails
Factory 2  >> Put     40 buttons     balance = 40 buttons

main      >> -----
main      >> Day 5
Supplier 2 >> Put     50 buttons     balance = 90 buttons
Supplier 2 >> Put     50 zippers     balance = 70 zippers
Supplier 1 >> Put    100 buttons     balance = 190 buttons
Supplier 1 >> Put    100 zippers     balance = 170 zippers
Factory 3  >> Holding 0 buttons      0 zippers
Factory 2  >> Holding 0 buttons     20 zippers
Factory 1  >> Holding 0 buttons     60 zippers
Factory 1  >> Get     20 buttons     balance = 170 buttons
Factory 2  >> Get    160 buttons     balance = 10 buttons
Factory 3  >> Get     10 buttons     balance = 0 buttons
Factory 3  >> Get    120 zippers     balance = 50 zippers
Factory 3  >> Pants   production fails
Factory 2  >> Jackets production succeeds, lot 2
Factory 1  >> Handbags production succeeds, lot 3
Factory 3  >> Put     10 buttons     balance = 10 buttons

main      >> -----
main      >> Day 6
Supplier 1 >> Put    100 buttons     balance = 110 buttons
Supplier 1 >> Put    100 zippers     balance = 150 zippers
Supplier 2 >> Put     50 buttons     balance = 160 buttons
Supplier 2 >> Put     50 zippers     balance = 200 zippers
Factory 1  >> Holding 0 buttons      0 zippers
Factory 2  >> Holding 0 buttons      0 zippers
Factory 3  >> Holding 0 buttons     120 zippers
Factory 3  >> Get     120 buttons     balance = 40 buttons
Factory 2  >> Get     40 buttons     balance = 0 buttons
Factory 2  >> Get     20 zippers     balance = 180 zippers
Factory 1  >> Get      0 buttons     balance = 0 buttons
Factory 1  >> Get     60 zippers     balance = 120 zippers
Factory 1  >> Handbags production fails
Factory 3  >> Pants   production succeeds, lot 4
Factory 2  >> Jackets production fails
Factory 2  >> Put     40 buttons     balance = 40 buttons

main      >> -----
main      >> Summary
main      >> Total Pants   = 4 lots
main      >> Total Handbags = 3 lots
main      >> Total Jackets = 2 lots

```

1. Print new day

2. Supply materials

3. Check holding materials

4. Retrieve materials

5. Success or failure (+ return materials)

Sorted in decreasing order of lots

All output lines are labeled with thread name
 (Thread.currentThread().getName())

```

--- exec-maven-plugin:3.0.0:exec (default-cli) @ projectsol_2 ---
main      >> read configs from src/main/java/Project2/config_1.txt

main      >> simulation days = 4
main      >> Supplier 1 daily supply rates = 50 buttons      50 zippers      50 velcros
main      >> Supplier 2 daily supply rates = 100 buttons     100 zippers     100 velcros
main      >> Factory 1  daily use   rates = 20 buttons      60 zippers      20 velcros      producing 10 Handbags
main      >> Factory 2  daily use   rates = 160 buttons     20 zippers      40 velcros      producing 20 Jackets
main      >> Factory 3  daily use   rates = 120 buttons     120 zippers     80 velcros      producing 40 Pants

```

```

main      >> -----
main      >> Day 1
Supplier 1 >> Put      50 buttons      balance = 50 buttons
Supplier 1 >> Put      50 zippers      balance = 50 zippers
Supplier 1 >> Put      50 velcros      balance = 50 velcros
Supplier 2 >> Put     100 buttons      balance = 150 buttons
Supplier 2 >> Put     100 zippers      balance = 150 zippers
Supplier 2 >> Put     100 velcros      balance = 150 velcros
Factory 1  >> Holding  0 buttons      0 zippers      0 velcros
Factory 3  >> Holding  0 buttons      0 zippers      0 velcros
Factory 2  >> Holding  0 buttons      0 zippers      0 velcros
Factory 2  >> Get     150 buttons      balance = 0 buttons
Factory 1  >> Get      0 buttons      balance = 0 buttons
Factory 2  >> Get      20 zippers      balance = 130 zippers
Factory 3  >> Get      0 buttons      balance = 0 buttons
Factory 1  >> Get      60 zippers      balance = 70 zippers
Factory 2  >> Get      40 velcros      balance = 110 velcros
Factory 3  >> Get      70 zippers      balance = 0 zippers
Factory 1  >> Get      20 velcros      balance = 90 velcros
Factory 3  >> Get      80 velcros      balance = 10 velcros
Factory 3  >> Pants   production fails
Factory 3  >> Put      70 zippers      balance = 70 zippers
Factory 2  >> Jackets production fails
Factory 2  >> Put     150 buttons      balance = 150 buttons
Factory 1  >> Handbags production fails

```

```

main      >> -----
main      >> Day 2
Supplier 2 >> Put     100 buttons      balance = 250 buttons
Supplier 2 >> Put     100 zippers      balance = 170 zippers
Supplier 2 >> Put     100 velcros      balance = 110 velcros
Supplier 1 >> Put      50 buttons      balance = 300 buttons
Supplier 1 >> Put      50 zippers      balance = 220 zippers
Supplier 1 >> Put      50 velcros      balance = 160 velcros
Factory 3  >> Holding  0 buttons      0 zippers      80 velcros
Factory 1  >> Holding  0 buttons      60 zippers      20 velcros
Factory 2  >> Holding  0 buttons      20 zippers      40 velcros
Factory 2  >> Get     160 buttons      balance = 140 buttons
Factory 3  >> Get     120 buttons      balance = 20 buttons
Factory 3  >> Get     120 zippers      balance = 100 zippers
Factory 1  >> Get      20 buttons      balance = 0 buttons
Factory 1  >> Handbags production succeeds, lot 1
Factory 2  >> Jackets production succeeds, lot 1
Factory 3  >> Pants   production succeeds, lot 1

```

```

main      >> -----
main      >> Day 3
Supplier 1 >> Put      50 buttons      balance = 50 buttons
Supplier 1 >> Put      50 zippers      balance = 150 zippers
Supplier 1 >> Put      50 velcros      balance = 210 velcros
Supplier 2 >> Put     100 buttons      balance = 150 buttons
Supplier 2 >> Put     100 zippers      balance = 250 zippers
Supplier 2 >> Put     100 velcros      balance = 310 velcros
Factory 2  >> Holding  0 buttons      0 zippers      0 velcros
Factory 3  >> Holding  0 buttons      0 zippers      0 velcros
Factory 1  >> Holding  0 buttons      0 zippers      0 velcros
Factory 1  >> Get      20 buttons      balance = 130 buttons
Factory 1  >> Get      60 zippers      balance = 190 zippers
Factory 3  >> Get     120 buttons      balance = 10 buttons
Factory 3  >> Get     120 zippers      balance = 70 zippers
Factory 1  >> Get      20 velcros      balance = 290 velcros
Factory 2  >> Get      10 buttons      balance = 0 buttons
Factory 2  >> Get      20 zippers      balance = 50 zippers
Factory 3  >> Get      80 velcros      balance = 210 velcros
Factory 2  >> Get      40 velcros      balance = 170 velcros
Factory 2  >> Jackets production fails
Factory 2  >> Put      10 buttons      balance = 10 buttons
Factory 1  >> Handbags production succeeds, lot 2
Factory 3  >> Pants   production succeeds, lot 2

```

config 1.txt

Demo 2

D, 4
 M, buttons, zippers, velcros
 S, Supplier 1, 50, 50, 50
 S, Supplier 2, 100, 100, 100
 F, Factory 1, Handbags, 10, 2, 6, 2
 F, Factory 2, Jackets, 20, 8, 1, 2
 F, Factory 3, Pants, 40, 3, 3, 2

```
main      >> -----
main      >> Day 4
Supplier 2 >> Put      100 buttons      balance = 110 buttons
Supplier 2 >> Put      100 zippers     balance = 150 zippers
Supplier 2 >> Put      100 velcros     balance = 270 velcros
Supplier 1 >> Put       50 buttons     balance = 160 buttons
Supplier 1 >> Put       50 zippers     balance = 200 zippers
Supplier 1 >> Put       50 velcros     balance = 320 velcros
Factory 1  >> Holding   0 buttons      0 zippers    0 velcros
Factory 2  >> Holding   0 buttons     20 zippers   40 velcros
Factory 3  >> Holding   0 buttons      0 zippers    0 velcros
Factory 3  >> Get       120 buttons     balance = 40 buttons
Factory 3  >> Get       120 zippers     balance = 80 zippers
Factory 2  >> Get       40 buttons     balance = 0 buttons
Factory 3  >> Get       80 velcros     balance = 240 velcros
Factory 1  >> Get        0 buttons     balance = 0 buttons
Factory 1  >> Get       60 zippers     balance = 20 zippers
Factory 1  >> Get       20 velcros     balance = 220 velcros
Factory 1  >> Handbags production fails
Factory 2  >> Jackets production fails
Factory 2  >> Put       40 buttons      balance = 40 buttons
Factory 3  >> Pants    production succeeds, lot 3

main      >> -----
main      >> Summary
main      >> Total Pants   =    3 lots
main      >> Total Handbags =    2 lots
main      >> Total Jackets =    1 lots
```

```
-----
BUILD SUCCESS
-----
```