# **Fruit shop**

Let's imagine that we have a fruit store. Every day in the store there are a number of activities, information about which is recorded in a file during the day. The current input file is sent to the program in CSV format (it is recommended to use standard libraries for parsing).

Your task is:

* validate the data from file to make sure that we won't process incorrect input
* generate a report based on the input file.

There are four activities at the store:

b - balance, the remnants of fruits at the beginning of the working day s - supply, means you are receiving new fruits from suppliers p - purchase, means someone has bought some fruit r - return, means someone who have bought the fruits now returns them back

Let's check in details all types of activities:

1. Balance. Fruit balance at the beginning of the work shift. The following line in the file will look like:
2. b,banana,100
3. The line above means there are 100 bananas at the beginning of the work shift.
4. Supply. You are accepting new fruits from suppliers. The following line in the file will look like:
5. s,banana,100
6. The line above means you receive 100 bananas.
7. Purchase. Buyers can visit your shop and buy some fruits. In this case you will have the following line in the file:
8. p,banana,13
9. The line above means someone has bought 13 bananas.
10. Return. Buyers can return you some fruits. In this case you will have the following line in the file:
11. r,banana,10
12. The line above means someone has returned you 10 bananas.

### **Input file example**

type,fruit,quantity b,banana,20 b,apple,100 s,banana,100 p,banana,13 r,apple,10 p,apple,20 p,banana,5 s,banana,50

### **Expecting report file example**

We are expecting to see how many fruits are available today after the work shift in your Fruit store.

fruit,quantity banana,152 apple,90

The line above means you have 152 bananas, and 90 apples in your Fruit store after the work shift. NOTE: if while processing transactions you detect some business logic mistakes (e.g not enough fruits to buy) throw an exception from service or handlers.

### **Validation**

For validation create separate class Validator with corresponding method. You can perform it after File has been read during parsing of raw data. If the file has mistakes, you should throw an exception. Example of incorrect input file:

type,fruit,quantity b,banana,20 p, s,banana,50

or

type,fruit,quantity b,banana,20 p,banana,-10 // Buyers will not be able to buy -10 bananas. -10 is incorrect input. s,banana,50

You can change by yourself.

1. package com.company;
3. import java.util.Scanner;
5. public class FruitShop {
6. public static void main ( String[] args ) {
7. Scanner scanner = new Scanner ( System.in );
8. String fruit = scanner.nextLine ( );
9. String day = scanner.nextLine ( );
10. double quantity = Double.parseDouble ( scanner.nextLine ( ) );
11. double price = 0.00;
13. boolean isError = false;
14. switch (day) {
15. case "Monday":
16. case "Tuesday":
17. case "Wednesday":
18. case "Thursday":
19. case "Friday":
20. if (day.equals ( "Monday" ) || day.equals ( "Tuesday" ) || day.equals ( "Wednesday" ) || day.equals ( "Thursday" ) || day.equals ( "Friday" ) || price != 0.00) {
21. if (fruit.equals ( "banana" )) {
22. price = 2.50;
23. } else if (fruit.equals ( "apple" )) {
24. price = 1.20;
26. } else if (fruit.equals ( "orange" )) {
27. price = 0.85;
29. } else if (fruit.equals ( "grapefruit" )) {
30. price = 1.45;
32. } else if (fruit.equals ( "kiwi" )) {
33. price = 2.70;
35. } else if (fruit.equals ( "pineapple" )) {
36. price = 5.50;
37. } else if (fruit.equals ( "grapes" )) {
38. price = 3.85;
39. } else {
40. isError = true;
42. }
43. } else {
44. System.out.println ( "error" );
45. ;
47. }

50. break;
51. case "Saturday":
52. case "Sunday":
54. if (day.equals ( "Saturday" ) || day.equals ( "Sunday" ) || price != 0.00) {
55. if (fruit.equals ( "banana" )) {
56. price = 2.70;
57. } else if (fruit.equals ( "apple" )) {
58. price = 1.25;
60. } else if (fruit.equals ( "orange" )) {
61. price = 0.90;
63. } else if (fruit.equals ( "grapefruit" )) {
64. price = 1.60;
66. } else if (fruit.equals ( "kiwi" )) {
67. price = 3.00;
69. } else if (fruit.equals ( "pineapple" )) {
70. price = 5.60;
71. } else if (fruit.equals ( "grapes" )) {
72. price = 4.20;
73. } else {
74. isError = true;
76. }
77. } else {
78. System.out.println ( "error" );
79. ;
81. }
83. break;

86. }
87. if (price != 0.00) {
88. double totalprice = price \* quantity;
89. System.out.printf ("%.2f",totalprice);
90. } else {
91. System.out.printf ( "error");
92. {

95. }
96. }
97. }
98. }