

ΜΕΘΟΔΟΛΟΓΙΕΣ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ

Εργασία 1

ΠΑΝΤΕΛΕΗΜΩΝ ΠΡΩΙΟΣ

ice18390023

6ο Εξάμηνο

ice18390023@uniwa.gr

ΜΑΕ04 15:00-17:00



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
UNIVERSITY OF WEST ATTICA

Υπεύθυνοι καθηγητές

ΝΙΚΗΤΑΣ ΚΑΡΑΝΙΚΟΛΑΣ

ΓΕΩΡΓΙΟΣ ΠΡΕΖΕΡΑΚΟΣ

ΠΑΥΛΙΔΗΣ ΘΕΟΔΟΣΙΟΣ

Τμήμα Μηχανικών και Πληροφορικής Υπολογιστών
13 Μαΐου 2021

Περιεχόμενα

1	CelsiusFahrenheit	1
2	Calculator	5
3	Grades	12
4	Stats	17
5	SQLite	34
5.1	Βάση δεδομένων stats.db	34
5.2	Κώδικας Stats_SQLite	34

Κατάλογος σχημάτων

1.1	Ενδεικτικό τρέξιμο του προγράμματος CelsiusFahrenheit	1
2.1	Ενδεικτικό τρέξιμο του προγράμματος Calculator μέρος 1	6
2.2	Ενδεικτικό τρέξιμο του προγράμματος Calculator μέρος 2	7
3.1	Ενδεικτικό τρέξιμο του προγράμματος Grades	13
4.1	Ενδεικτικό τρέξιμο του προγράμματος Stats	19
5.1	Ενδεικτικό τρέξιμο του προγράμματος Stats_sqlite	35

Κώδικες

1.1	Κώδικας CelsiusFahrenheit	1
2.1	Κώδικας Calculator	7
3.1	Κώδικας Grades	13
4.1	Κώδικας Stats	20
5.1	Κώδικας Stats_sqlite	35

1 CelsiusFahrenheit

Το πρόγραμμα CelsiusFahrenheit (κώδικας 1.1) κάνει μετατροπή της θερμοκρασίας από Celsius σε Fahrenheit. Ποίο συγκεκριμένα δέχεται ως πρώτο όρισμα:

- -h για να εμφανίσει την χρήση του προγράμματος
- 0 για να μετατρέψει βαθμούς από Celsius σε Fahrenheit
- 1 για να μετατρέψει βαθμούς από Fahrenheit σε Celsius

στην περίπτωση που δεν δοθεί κανένα όρισμα, εμφανίζει και πάλι την χρήση του προγράμματος και τερματίζει.

```
padelis@MAE-project1$ java CelsiusFahrenheit
Usage: <arg1>

-h:    help
if <arg1> is 0 converts Celsius to Fahrenheit
if <arg1> is 1 converts Fahrenheit to Celsius
padelis@MAE-project1$ java CelsiusFahrenheit -h
Usage: <arg1>

-h:    help
if <arg1> is 0 converts Celsius to Fahrenheit
if <arg1> is 1 converts Fahrenheit to Celsius
padelis@MAE-project1$ java CelsiusFahrenheit 0
Please enter Celsius temprature: 30
30.0 celsius is 86.0 fahrenheit

Goodbye!
padelis@MAE-project1$ java CelsiusFahrenheit 1
Please enter Fahrenheit temprature: 86
86.0 fahrenheit is 30.0 celsius

Goodbye!
padelis@MAE-project1$ java CelsiusFahrenheit 1
Please enter Fahrenheit temprature: this is not a double
Numeric exception: For input string: "this is not a double"

Goodbye!
padelis@MAE-project1$
```

Σχήμα 1.1: Ενδεικτικό τρέξιμο του προγράμματος CelsiusFahrenheit

```
1 import java.util.Scanner;
```

```
2
```

```
3 // @author padelis
4
5 public class CelsiusFahrenheit {
6
7     // @param args <option|help>
8     public static void main(String[] args) {
9
10        // Αν το δοθεί η πρώτη παράμετρος ως "-h" ή δεν υπάρχουν
11        // παράμετροι, τότε θα καλέσει την μέθοδο usage και μετά
12        // θα κλείσει το πρόγραμμα
13        // Επίσης, πρέπει πρώτα να είναι το args.length και μετά
14        // το args[0].equals("-h"), γιατί στην περίπτωση που δεν
15        // δοθεί όρισμα θα πετάξει ArrayOutOfBoundsException
16        // ενώ με αυτήν την σειρά δεν θα υπάρξει πρόβλημα
17        if(args.length == 0 || args[0].equals("-h")) {
18
19            CelsiusFahrenheit.usage();
20            System.exit(0);
21        }
22
23        // Δημιουργεί ένα αντικείμενο scanner στο κανάλι stdin
24        Scanner buffer = new Scanner(System.in);
25        String line = null;
26        double celsius = 0.0;
27        double fahrenheit = 0.0;
28
29        try {
30
31            // Αν το πρώτο όρισμα είναι '0', τότε θα εμφανιστεί
32            // κατάλληλο μήνυμα και η είσοδος που θα δοθεί από
33            // τον χρήστη θα μετατραπεί σε fahrenheit.
34            // Αλλιώς αν το όρισμα είναι 1, τότε θα εμφανιστεί
35            // κατάλληλο μήνυμα και η είσοδος που θα δοθεί από
36            // τον χρήστη θα μετατραπεί σε celsius.
37            // Αλλιώς θα εμφανιστεί το usage και θα τερματίσει
38            if (args[0].equals("0")) {
39
40                System.out.print("Please enter Celsius temprature: ");
41                line = buffer.nextLine();
42
43                celsius = Double.parseDouble(line);
44                fahrenheit = ((double)9/((double)5) * celsius + 32;
45
46                System.out.println(celsius + " celsius is "
```

```
47         + fahrenheit + " fahrenheit");
48
49     }
50     else if(args[0].equals("1")) {
51
52         System.out.print("Please enter Fahrenheit temprature: ");
53         line = buffer.nextLine();
54
55         fahrenheit = Double.parseDouble(line);
56         celsius = (fahrenheit - 32) / (9.0/5.0);
57
58         //System.out.println(fahrenheit + " fahrenheit is "
59         //+ String.format("%.2f",celsius) + " celsius");
60         System.out.println(fahrenheit + " fahrenheit is "
61             + celsius + " celsius");
62
63     }
64     else {
65         CelsiusFahrenheit.usage();
66         System.exit(0);
67     }
68
69     // Μπορεί το Double.parseDouble να πετάξει error
70     // από λάθος format string και προσπαθούμε να
71     // πιάσουμε αυτό το error.
72 }
73 catch(NumberFormatException e) {
74     System.out.println("Numeric exception: " + e.getMessage() );
75
76     // Σε περίπτωση που είναι κλειστό το scanner θα πετάξει exception
77 }
78 catch(IllegalStateException e) {
79     System.out.println("Scanner closed exception: "
80         + e.getMessage() );
81
82     // Οποιοδήποτε άλλο exception
83 }
84 catch(Exception e) {
85     System.out.println(e.toString() );
86 }
87
88 buffer.close();
89
90 System.out.println("\nGoodbye!");
```

```
91 }
92 }
93
94 // Εμφανίζει την χρήση του προγράμματος
95 static void usage() {
96     System.out.println("Usage:\t<arg1>\n");
97     System.out.println("-h:\thelp");
98     System.out.println("if <arg1> is 0 converts Celsius to Fahrenheit");
99     System.out.println("if <arg1> is 1 converts Fahrenheit to Celsius");
100 }
101 }
```

Κώδικας 1.1: Κώδικας CelsiusFahrenheit

2 Calculator

Το πρόγραμμα Calculator (κώδικας 2.1) υπολογίζει το άθροισμα "μη πεπερασμένων" δεδομένων μέχρι να συναντήσει την λέξη "end". Ακόμα, μπορεί να κάνει και διαίρεση. Όπως φαίνεται και στις εικόνες 2.1 και 2.2 δεν δέχεται κανένα όρισμα, όμως εμφανίζει ένα menu επιλογών:

- 1 Πρόσθεση
- 2 Διαίρεση
- 3 Βοήθεια (εμφανίζει την χρήση των επιλογών)
- 0 Εγκατάλειψη

και επαναλαμβάνει μέχρις ότου να δοθεί η επιλογή 0.

```
padelis@MAE-project1$ java Calculator

Options:
0) exit
1) addition
2) division
3) help

Option: 3
Option 1: reads a line with double numbers which will be summarized until keyword "end"
Option 2: reads 2 integers and prints the div and the mod of those two numbers if the divisor
is not zero

Options:
0) exit
1) addition
2) division
3) help

Option: 1
Addition> 1 2 end 3
Result: 3.0

Options:
0) exit
1) addition
2) division
3) help

Option: 1
Addition> 1 2
Word "end" wasn't given, but we reached the last element
Result: 3.0

Options:
0) exit
1) addition
2) division
3) help

Option: 1
Addition> end
Result: 0.0

Options:
0) exit
1) addition
2) division
3) help

Option: 2
Divisible> 3
Divisor> 2
Div is 1 and mod is 1

Options:
```

Σχήμα 2.1: Ενδεικτικό τρέξιμο του προγράμματος Calculator μέρος 1


```
Options:
0) exit
1) addition
2) division
3) help

Option: 2
Divisible> 3
Divisor> 0
Arithemtic error / by zero

Options:
0) exit
1) addition
2) division
3) help

Option: 2
Divisible> not a double
Invalid number java.lang.NumberFormatException: For input string: "not a double"

Options:
0) exit
1) addition
2) division
3) help

Option: 0

Goodbye!
padelis@MAE-project1$
```

Σχήμα 2.2: Ενδεικτικό τρέξιμο του προγράμματος Calculator μέρος 2

```
1 import java.util.Scanner;
2
3 // @author padelis
4
5 public class Calculator {
6
7     public static void main(String[] args) {
8
9         int option = 0;
10        String line = null;
11        String[] inputs = null;
12        boolean flag = true;
13        double result = 0.0;
14        int divisible = 0, divisor = 0;
15
16        // Δημιουργεί ένα αντικείμενο scanner στο κανάλι stdin
17        Scanner scanner = new Scanner(System.in);
18
19        // Το try είναι εντός του βρόγχου επανάληψης διότι
20        // στην περίπτωση που υπάρξει exception θα το χειριστή
21        // κατάλληλα και θα συνεχίσει κανονικά
22        do {
```

```
23
24     try {
25
26         // καλεί το menu με όρισμα το scanner έτσι ώστε να
27         // λάβει input από τον χρήστη και να επιστρέψει την
28         // επιλογή που έκανε από το menu
29         option = Calculator.menu(scanner);
30
31         // Το flag χρησιμοποιείται για να ξέρουμε αν υπήρξε
32         // end ή τελείωσε η γραμμή που δόθηκε
33         flag = true;
34         result = 0.0;
35
36         switch(option) {
37
38             case 1:
39                 // Στην περίπτωση που δοθεί η επιλογή 1, εμφανίζει
40                 // κατάλληλο μήνυμα στην οθόνη και περιμένει ένα
41                 // sequence αριθμών προς πρόσθεση.
42
43                 System.out.print("Addition> ");
44                 line = scanner.nextLine();
45
46                 // Δημιουργεί έναν πίνακα από strings και τα
47                 // χωρίζει βάση των white spaces
48                 inputs = line.split("\\s+");
49
50                 // Το foreach loop κάνει επανάληψη σε όλα τα
51                 // στοιχεία του πίνακα inputs. Αν κάποιο
52                 // στοιχείο είναι end, τότε τοποθετεί στην
53                 // μεταβλητή flag false και βγαίνει από
54                 // τον βρόγχο επανάληψης αλλιώς προσπαθεί
55                 // να μετατρέψει το string σε double και το
56                 // συναθροίζει στην μεταβλητή result
57                 for(String tmp: inputs) {
58                     if( tmp.equals("end") ) {
59                         flag = false;
60                         break;
61                     }
62
63                     result += Double.parseDouble(tmp);
64                 }
65
66                 // Αν δεν έχει δοθεί το end, τότε θα εμφανιστεί
```

```
67 // κατάλληλο μήνυμα και έπειτα το αποτέλεσμα
68 if ( flag ) {
69     System.out.println("Word \"end\" wasn't "+
70         "given, but we reached the last element");
71 }
72
73 System.out.println("Result: " + result);
74
75 break;
76
77 case 2:
78     // Στην περίπτωση που θα δοθεί η επιλογή 2, θα
79     // θα ζητηθεί από τον χρήστη η ο διαιρετέος
80     // και έπειτα ο διαιρέτης
81
82     System.out.print("Divisible> ");
83     line = scanner.nextLine();
84     divisible = Integer.parseInt(line);
85
86     System.out.print("Divisor> ");
87     line = scanner.nextLine();
88     divisor = Integer.parseInt(line);
89
90     // Γινόταν έλεγχος αν ο divisor είναι 0 και
91     // αν ήταν τότε εμφάνιζε κατάλληλο μήνυμα στην οθόνη
92     // και έκανε break από την case, αλλιώς
93     // συνέχιζε κανονικά και εμφάνιζε το αποτέλεσμα
94     // Όμως, πλέον χειρίζεται με το ArithmeticException
95
96     //if (divisor == 0) {
97         //System.out.println("Divisor can not be 0 !!!");
98         //break;
99     //}
100
101     System.out.println("Div is " + divisible / divisor
102         + " and mod is " + divisible % divisor);
103
104     break;
105
106 case 3:
107     // Εμφανίζει την χρήση των επιλογών
108
109     System.out.println("Option 1: reads a line "
110         + "with double numbers which "
```

```
111         + "will be summarized until keyword \"end\"");
112         System.out.println("Option 2: reads 2 integers and "
113         + "prints the div and the mod of those two "
114         + "numbers if the divisor is not zero");
115         break;
116     }
117
118 }
119 catch(NumberFormatException e){
120     // To Double.parseDouble μπορεί να πετάξει τέτοιο exception
121     System.out.println("Invalid number " + e.toString());
122 }
123 catch(ArithmeticException e) {
124     // Η διαίρεση με το 0 μπορεί να πετάξει τέτοιο exception
125     System.out.println("Arithemitc error " + e.getMessage());
126 }
127 catch(IllegalStateException e) {
128     // To scanner αν είναι κλειστό μπορεί να έχει τέτοιο exception
129     System.out.println("Invalid number " + e.toString());
130 }
131 catch(Exception e) {
132     // Οποιοδήποτε άλλο exception
133     System.out.println(e.toString());
134 }
135
136 // όσο δεν επιλέγεται το 0 ως option συνεχίζει την επανάληψη
137 }while(option != 0);
138
139 scanner.close();
140
141 System.out.println("\nGoodbye!");
142
143 }
144
145 // Το menu δέχεται ως παράμετρο ένα αντικείμενο scanner και επιστρέφει
146 // την επιλογή
147 static int menu(Scanner scanner) {
148
149     // Το flag υπάρχει για να εμφανίζει κατάλληλο μήνυμα στην περίπτωση
150     // που δοθεί λάθος επιλογή πάνω από μια φορές
151     int opt = -1;
152     boolean flag = false;
153     String line = null;
154
```

```
155 do {
156
157     // Εμφανίζει κατάλληλο μήνυμα αν δεν είναι η πρώτη φορά
158     if (flag) {
159         System.out.println("Option \" + opt + "\" is not valid!!!");
160     }
161
162     // Εμφανίζει τις επιλογές
163     System.out.println("\nOptions:");
164     System.out.println("0) exit");
165     System.out.println("1) addition");
166     System.out.println("2) division");
167     System.out.println("3) help");
168
169     // Περιμένει είσοδο από τον χρήστη
170     System.out.print("\nOption: ");
171     line = scanner.nextLine();
172     flag = true;
173
174     try {
175         // Το μετατρέπει σε ακέραιο
176         opt = Integer.parseInt(line);
177     }
178     catch (NumberFormatException e){
179         // Αν δεν ήταν ακέραιος τότε εμφανίζει κατάλληλο μήνυμα
180         // Επίσης, εκχωρεί στο flag false για να μην εκτυπωθεί
181         // το μήνυμα λάθους επιλογής εφόσον ήταν πρόβλημα format
182         System.out.println("Invalid input " + e.getMessage());
183         flag = false;
184     }
185
186     // 2 τρόποι επανάληψης μέχρις ότου η επιλογή να είναι 0 <= opt <= 3
187     //}while( !(opt <= 3 && opt >= 0) );
188     }while( opt > 3 || opt < 0 );
189
190     return opt;
191 }
192
193 }
```

Κώδικας 2.1: Κώδικας Calculator

3 Grades

Ο κώδικας 3.1 δεν δέχεται ορίσματα. Περιμένει από τον χρήστη να πληκτρολογήσει τουλάχιστον μία γραμμή, όπου θεωρητικά η πρώτη λέξη θα είναι ο αριθμός μητρώου του φοιτητή, έπειτα θα ακολουθούν τα ζεύγη το όνομα του μαθήματος με τον βαθμό και το τέλος θα υποδεικνύεται με την λέξη "end". Σε περίπτωση που δεν δοθεί το "end" και φτάσει στο τέλος πάλι βγάζει τον μέσο όρο για όλα τα προηγούμενα δεδομένα. Το πρόγραμμα τερματίζει όταν δοθεί ο αριθμός μητρώου "000000".

Κάτι που μπορεί να θεωρείται ως **bug** είναι ότι στην είσοδο π.χ. ice18390023 end, εμφανίζει πως ο μέσος όρος είναι NaN όπου αν θέλουμε να εμφανίζει πως δεν υπάρχουν δεδομένα μπορούμε να βάλουμε ένα flag εντός του βρόγχου που υπολογίζει τον μέσο όρο ώστε να ξέρουμε πως μπήκε τουλάχιστον μία φορά.

```
padelis@MAE-project1$ java Grades
Insert data:
061125 ProgrI 6 dbI 7 mathI 5.5 end
071234 dbII 5 ProgrC++ 4 PSD 4.5 ProgrJava 7 end
051058 MathII 7 PSD 6 end
081092 end
071234 dbI 5.5 mathI 5.5 ProgrJava 5 end
000000
Average grade for 061125 is 6.166666666666667
Average grade for 071234 is 5.125
Average grade for 051058 is 6.5
Average grade for 081092 is NaN
Average grade for 071234 is 5.333333333333333

Goodbye!
padelis@MAE-project1$ java Grades
Insert data:
wrong

Input has only 1 word or
it dosen't end with "end" or
there was a lesson or a grade missing orit was empty line

wrong data input end
Invalid number For input string: "input"

Input has only 1 word or
it dosen't end with "end" or
there was a lesson or a grade missing orit was empty line

000000
There was no records

Goodbye!
padelis@MAE-project1$ java Grades
Insert data:
ice18390023 java 10 math 6 end smth else
000000
Average grade for ice18390023 is 8.0

Goodbye!
padelis@MAE-project1$
```

Σχήμα 3.1: Ενδεικτικό τρέξιμο του προγράμματος Grades

```
1 import java.util.Arrays;
2 import java.util.Scanner;
3
4 // @author padelis
5
6 public class Grades {
7
8     public static void main(String[] args) {
9
10         // Δημιουργεί ένα scanner για το κανάλι stdin
11         Scanner sc = new Scanner(System.in);
12         String line = null;
13         String newStudent = null;
14         String[] results = null;
15         String[] inputs = null;
16         int i = 0;
17         double sum = 0.0;
18
19         // Εμφανίζει κατάλληλο μήνυμα και περιμένει είσοδο από τον χρήστη
20         System.out.println("Insert data: ");
21
22         // Μια δομή επανάληψης για να διαβάξει δεδομένα από τον χρήστη
23         do{
24
25
26             line = sc.nextLine();
27             // Σπάει την είσοδο βάση των white spaces
28             inputs = line.split("\\s+");
29             sum = 0.0;
30
31
32             try {
33
34                 // Αν το πρώτο στοιχείο του πίνακα είναι "000000" τότε
35                 // σημαίνει πως δεν υπάρχουν άλλα δεδομένα
36                 if(inputs[0].equals("000000"))
37                     break;
38
39
40                 // Ξεκινάει από το δεύτερο στοιχείο του πίνακα γιατί
41                 // το πρώτο είναι το AM του φοιτητή
42                 // Κάνει έλεγχο μήπως αντί για όνομα μαθήματος υπάρχει
43                 // η λέξη "end", αν δεν υπάρχει συναθροίζει το επόμενο
44                 // στοιχείο όπου είναι ο αριθμός του μαθήματος και
```



```
45 // συνεχίζει με βήμα 2
46 // Αν υπάρχουν λάθος στοιχεία, τότε συνεχίζει στην
47 // επόμενη γραμμή επειδή το try-catch είναι εντός
48 // του βρόγχου επανάληψης
49 for( i = 1; !inputs[i].equals("end"); i+=2)
50     sum += Double.parseDouble(inputs[i+1]);
51
52 // Δημιουργεί ένα νέο string με τον μέσο όρο και
53 // κατάλληλο μήνυμα
54 newStudent = "Avarage grade for " + inputs[0]
55             + " is " + (sum/((i+1)/2-1));
56
57 // Το τοποθετεί σε έναν πίνακα
58 // Αν ο πίνακας είναι null, τότε κάνει new ένα string
59 // πίνακα μεγέθους ένα, το τοποθετεί εκεί και κάνει
60 // continue
61 if ( results == null ) {
62     results = new String[1];
63     results[0] = newStudent;
64     continue;
65 }
66
67 // Αν δεν γίνει η παραπάνω συνθήκη τότε θα μεγαλώσει το
68 // μέγεθος του πίνακα κατά ένα και θα τοποθετήσουμε
69 // το string στο τέλος
70 results = Arrays.copyOf(results, results.length + 1);
71 results[results.length-1] = newStudent;
72
73
74 }
75 catch(NumberFormatException e){
76     // Το Double.parseDouble μπορεί να πετάξει τέτοιο exception
77     System.err.println("Invalid number " + e.getMessage() + "\n");
78 }
79 catch(ArrayIndexOutOfBoundsException e) {
80     // Μπορεί να δοθεί είσοδος με μόνο μια λέξη και
81     // να μην είναι "000000" ή να μην τελειώνει με "end"
82     // ή να είναι κενή γραμμή ή να μην υπάρχει κάποιο
83     // όνομα μαθήματος ή βαθμός
84     System.err.println("\nInput has only 1 word or\n"
85                       + "it dosen't end with \"end\" or\n"
86                       + "there was a lesson or a grade missing or\n"
87                       + "it was empty line\n");
88 }
```

```
89     catch(Exception e) {
90         System.err.println(e.toString() + "\n");
91     }
92
93     // Γίνεται επανάληψη μέχρι να δοθεί "000000" και να γίνει break
94 }while(true);
95
96
97 // Κλείνει το scanner
98 sc.close();
99
100 try {
101
102     // Εμφανίζει τα αποτελέσματα
103     for(String tmp : results) {
104         System.out.println(tmp);
105     }
106 }
107 catch(NullPointerException e) {
108     // Αν δεν υπήρχαν δεδομένα ή ήταν όλα λάθος
109     // και τα έπιανε το try-catch, τότε το results
110     // θα έχει παραμείνει null
111     System.out.println("There was no records");
112 }
113 catch(Exception e) {
114     System.err.println(e.toString());
115 }
116
117 System.out.println("\nGoodbye!");
118
119 }
120
121 }
```

Κώδικας 3.1: Κώδικας Grades

4 Stats

Ο κώδικας 4.1 υλοποιεί 2 κλάσης. Η πρώτη κλάση είναι η Stats όπου περιέχει την main method, ενώ η δεύτερη είναι η DataSet όπου διαχειρίζεται τα δεδομένα. Η DataSet έχει τα εξής χαρακτηριστικά:

- private String inName, όπου είναι το όνομα του αρχείου εισόδου
- private String outName, όπου είναι το όνομα του αρχείου εξόδου
- private BufferedReader inStream, που είναι το κανάλι όπου θα διαβάσει τα δεδομένα
- private PrintWriter outStream, που είναι το κανάλι όπου θα εκτυπώσει ταξινομημένα τα δεδομένα
- private int[] data, που είναι τα δεδομένα που έχει διαβάσει
- private boolean init, αν υπάρχουν δεδομένα
- private boolean sorted, αν είναι ταξινομημένα τα δεδομένα

Αποτελείται και από 2 constructors:

- public DataSet(), που είναι ο default constructor και βάζει ως όνομα εισόδου το αρχείο numbers.txt και ως αρχείου εξόδου το output.txt
- public DataSet(String inName, String outName), δέχεται τα ονόματα εισόδου και εξόδου

Ακόμα, εκτός από τα getters and setters υπάρχουν οι εξής μέθοδοι:

- public boolean readData(), όπου διαβάζει τα δεδομένα από το αρχείο εισόδου και τα αποθηκεύει στον πίνακα data
- public void export(), όπου εξάγει τα δεδομένα στο αρχείο εξόδου
- public void printStats(), όπου εμφανίζει στατιστικά πάνω στα δεδομένα όπως:
 - τον μικρότερο βαθμό των στοιχείων
 - τον μεγαλύτερο βαθμό των στοιχείων
 - το πλήθος των στοιχείων
 - τον μέσο όρο των στοιχείων
- public int min(), επιστρέφει τον μικρότερο αριθμό των δεδομένων
- public int max(), επιστρέφει τον μέγιστο αριθμό των δεδομένων
- public int sum(), επιστρέφει το άθροισμα των δεδομένων
- public double mean(), επιστρέφει τον μέσο όρο των δεδομένων

- `public double percentiles(double percentile)`, επιστρέφει τον αριθμό που βρίσκεται από κάτω π.χ. 50% των δεδομένων
- `public void addData(int value)`, προσθέτει ένα ακόμα στοιχείο στον πίνακα `data`
- `public void sortData()`, ταξινομεί τα δεδομένα
- `public boolean openInStream()`, ανοίγει το κανάλι εισόδου βάσει του ονόματος `inStream`
- `public boolean closeInStream()`, κλείνει το κανάλι εισόδου
- `public boolean openOutStream()`, ανοίγει το κανάλι εξόδου βάσει του ονόματος `outStream`
- `public void closeOutStream()`, κλείνει το κανάλι εξόδου

Το πρόγραμμα (κώδικας 4.1) δέχεται δύο ορίσματα:

- το πρώτο όρισμα είναι το αρχείο εισόδου όπου θα διαβάσει τα δεδομένα
- το δεύτερο όρισμα είναι το αρχείο εξόδου όπου θα αποθηκεύσει ταξινομημένα τα δεδομένα που διάβασε

Υπολογίζει και εμφανίζει στην οθόνη

- τον μικρότερο βαθμό των στοιχείων
- τον μεγαλύτερο βαθμό των στοιχείων
- το πλήθος των στοιχείων
- τον μέσο όρο των στοιχείων
- το σημείο όπου από κάτω βρίσκεται το 25% των στοιχείων
- το σημείο όπου από κάτω βρίσκεται το 50% των στοιχείων

```
padelis@MAE-project1$ java Stats
Usage: arg1 arg2
arg1: <input file with data numbers>
arg2: <output file>
padelis@MAE-project1$ java Stats numbers.txt out.txt
Min: 823
Max: 16776935
Data length: 100000
Mean: 203.71272
The percentile of 25 is 4170301.5
The percentile of 50 is 8386878.5

Goodbye!
padelis@MAE-project1$ wc -l numbers.txt
100000 numbers.txt
padelis@MAE-project1$ wc -l out.txt
100000 out.txt
padelis@MAE-project1$ head -7 out.txt
823
1797
2040
2062
2117
2160
2175
padelis@MAE-project1$ head -7 numbers.txt
9351807
11706141
10690066
7074694
2034652
8192550
6314209
padelis@MAE-project1$ tail -7 out.txt
16776448
16776642
16776649
16776731
16776774
16776819
16776935
padelis@MAE-project1$ tail -7 numbers.txt
6663713
15388370
4408344
9048569
10516672
9672161
10621243
padelis@MAE-project1$
```

Σχήμα 4.1: Ενδεικτικό τρέξιμο του προγράμματος Stats

```
1 import java.io.BufferedReader;
2 import java.io.FileNotFoundException;
3 import java.io.FileReader;
4 import java.io.IOException;
5 import java.io.PrintWriter;
6 import java.util.Arrays;
7
8 // @author padelis
9
10 public class Stats {
11
12     public static void main(String[] args) {
13
14         // Ελέγχει αν υπάρχουν τουλάχιστον 2 ορίσματα
15         // Αν δεν υπάρχουν εμφανίζει κατάλληλο μήνυμα και κλείνει
16         if(args.length < 2) {
17             System.out.println("Usage: arg1 arg2");
18             System.out.println("arg1: <input file with data numbers>");
19             System.out.println("arg2: <output file>");
20             System.exit(0);
21         }
22
23         // Δημιουργεί ένα στοιχείο DataSet και του περνάει στον
24         // constructor τα δύο αρχεία εισόδου και εξόδου
25         DataSet data = new DataSet(args[0], args[1]);
26
27         // Διαβάζει τα δεδομένα από το αρχείο
28         data.readData();
29
30         // Ταξινομεί τα δεδομένα
31         data.sortData();
32
33         // Εκτυπώνει τα δεδομένα στο αρχείο εξόδου
34         data.export();
35
36         // Εμφανίζει τα στατιστικά των δεδομένων
37         data.printStats();
38
39         // Εμφανίζει τα percentile 25 και 50
40         System.out.println("The percentile of 25 is " + data.percentiles(25));
41         System.out.println("The percentile of 50 is " + data.percentiles(50));
42
43         System.out.println("\nGoodbye!");
44     }
```

```
45 }
46
47 }
48
49
50
51
52
53 class DataSet{
54
55     // attributes
56     private String inName = null;
57     private String outName = null;
58     private BufferedReader inStream = null;
59     private PrintWriter outStream = null;
60     private int[] data = null;
61     private boolean init = false;
62     private boolean sorted = false;
63
64
65
66     // constructors
67     public DataSet(String inName, String outName) {
68         // κλήση constructor χωρίς ορίσματα
69         this();
70         this.inName = inName;
71         this.outName = outName;
72     }
73
74     public DataSet() {
75         this.inName = new String("numbers.txt");
76         this.outName = new String("output.txt");
77         this.data = null;
78         this.init = false;
79         this.sorted = false;
80     }
81
82
83
84
85     // getters
86     public String getInName() {
87         return inName;
88     }
```

```
89
90 public int[] getData() {
91     return data;
92 }
93
94 public BufferedReader getInStream() {
95     return inStream;
96 }
97
98 public String getOutName() {
99     return outName;
100 }
101
102 public PrintWriter getOutStream() {
103     return outStream;
104 }
105
106 public boolean isInit() {
107     return init;
108 }
109
110 public boolean getSorted() {
111     return sorted;
112 }
113
114
115
116 // setters
117 public void setInName(String inName) {
118     this.inName = inName;
119 }
120
121 public void setData(int[] data) {
122     this.data = data;
123 }
124
125 public void setInStream(BufferedReader inStream) {
126     this.inStream = inStream;
127 }
128
129 public void setOutStream(PrintWriter outStream) {
130     this.outStream = outStream;
131 }
132
```



```
133 public void setOutName(String outName) {
134     this.outName = outName;
135 }
136
137 public void setInit(boolean init) {
138     this.init = init;
139 }
140
141 public void setSorted(boolean sorted) {
142     this.sorted = sorted;
143 }
144
145
146
147 // methods
148
149 // Αυτή η μέθοδος διαβάζει δεδομένα από το αρχείο που έχει
150 // προσδιορισθεί
151 public boolean readData() {
152
153     String line = null;
154     int value;
155
156     // Καλεί να ανοίξει το κανάλι και ελέγχει αν άνοιξε
157     // αν δεν ανοίξει, εκτυπώνει κατάλληλο μήνυμα στο stderr
158     // και επιστρέφει false.
159     if ( !this.openInStream() ) {
160         System.out.println("Couldn't read data");
161         return false;
162     }
163
164     // Προσπαθούμε να πιάσουμε IO exception από το BufferedReader
165     try {
166
167         // διαβάζει κάθε γραμμή μέχρι το EOF (null)
168         while( (line = inStream.readLine()) != null) {
169
170             // Προσπαθούμε να πιάσουμε NumberFormat exception
171             // και στην περίπτωση που υπάρχει απλώς αγνοούμε
172             // την συγκεκριμένη γραμμή και συνεχίζει κανονικά.
173             // Επίσης, θα μπορούσε να είναι στο προηγούμενο
174             // try-catch, αλλά τότε δεν θα μπορούσαμε να
175             // συνεχίσουμε το διάβασμα τον υπόλοιπων γραμμών
176             try {
```

```
177
178     // Το μετατρέπει σε ακέραιο
179     value = Integer.parseInt(line);
180     // Το προσθέτει στον πίνακα
181     this.addData(value);
182 }
183 catch(NumberFormatException e) {
184     System.err.println("This input was wrong "
185         + "but we will continue "
186         + e.getMessage());
187 }
188 }
189
190 } catch (IOException e) {
191     System.err.println("IO exception " + e.getMessage());
192 }
193
194 this.closeInStream();
195
196 return true;
197 }
198
199
200 // Εκτυπώνει τα δεδομένα σε αρχείο
201 public void export() {
202
203     // Ελέγχει αν υπάρχουν δεδομένα για να εκτυπώσει στο αρχείο
204     // αν δεν υπάρχουν, τότε εκτυπώνει κατάλληλο μήνυμα στο
205     // stderr και επιστρέφει
206     if ( !this.isInit() ) {
207         System.err.println("Data isn't been initialized");
208         return;
209     }
210
211     // Καλεί μέθοδο για να ανοίξει το κανάλι εξόδου και
212     // ελέγχει αν άνοιξε
213     if( this.openOutputStream() ) {
214
215         // εκτυπώνει τα δεδομένα με την foreach
216         for(int tmp: data) {
217             outStream.println(tmp);
218         }
219
220         // κλείνει το κανάλι
```

```
221     this.closeOutputStream();
222   }
223 }
224
225
226 // Εμφανίζει στατιστικά στα δεδομένα
227 public void printStats() {
228
229     // Ελέγχει αν υπάρχουν δεδομένα
230     // Αν δεν υπάρχουν, εμφανίζει κατάλληλο μήνυμα στο
231     // stderr και επιστρέφει
232     if ( !this.isInit() ) {
233         System.err.println("Data isn't been initialized");
234         return;
235     }
236
237     // Εμφανίζει τα δεδομένα
238     System.out.println("Min: " + this.min());
239     System.out.println("Max: " + this.max());
240     System.out.println("Data length: " + this.data.length);
241     System.out.println("Mean: " + this.mean());
242 }
243
244
245 // Βρίσκει το μικρότερο αριθμό πάνω στα δεδομένα
246 public int min() {
247
248     // Ελέγχει αν υπάρχουν δεδομένα
249     // Αν δεν υπάρχουν, εμφανίζει κατάλληλο μήνυμα στο
250     // stderr και επιστρέφει 0
251     if ( !this.isInit() ) {
252         System.err.println("Data isn't been initialized");
253         return 0;
254     }
255
256     // Ελέγχει αν τα δεδομένα είναι ταξινομημένα
257     // Αν είναι τότε δεν υπάρχει λόγος να το αναζητήσει
258     // γιατί είναι το μικρότερο στοιχείο εφόσον γνωρίζουμε
259     // πως η ταξινόμηση είναι με αύξουσα σειρά
260     //if ( this.getSorted() ) {
261         //return data[0];
262     //}
263
264     // Τοποθετεί στο min το πρώτο στοιχείο
```

```
265     int min = data[0];
266
267     // Ελέγχει όλα τα στοιχεία (συμπεριλαμβανομένου και του
268     // στοιχείου που έγινε η αρχικοποίηση) για να βρει
269     // το ελάχιστο στοιχείο του πίνακα
270     for( int tmp : data) {
271
272         if ( min > tmp) {
273             min = tmp;
274         }
275     }
276     return min;
277 }
278
279 // Βρίσκει το μεγαλύτερο αριθμό πάνω στα δεδομένα
280 public int max() {
281
282     // Ελέγχει αν υπάρχουν δεδομένα
283     // Αν δεν υπάρχουν, εμφανίζει κατάλληλο μήνυμα στο
284     // stderr και επιστρέφει 0
285     if ( !this.isInit() ) {
286         System.err.println("Data isn't been initialized");
287         return 0;
288     }
289
290     // Ελέγχει αν τα δεδομένα είναι ταξινομημένα
291     // Αν είναι τότε δεν υπάρχει λόγος να το αναζητήσει
292     // γιατί είναι το μεγαλύτερο στοιχείο εφόσον γνωρίζουμε
293     // πως η ταξινόμηση είναι με αύξουσα σειρά
294     //if ( this.getSorted() ) {
295         //return data[data.length-1];
296     //}
297
298     // Αρχικοποιεί το max με το πρώτο στοιχείο
299     int max = data[0];
300
301     // Ελέγχει όλα τα στοιχεία (συμπεριλαμβανομένου και του
302     // στοιχείου που έγινε η αρχικοποίηση) για να βρει
303     // το μέγιστο στοιχείο του πίνακα
304     for( int tmp : data) {
305
306         if ( max < tmp) {
307             max = tmp;
308         }
309     }
```

```
309     }
310     return max;
311 }
312
313 // Επιστρέφει το άθροισμα των δεδομένων
314 public int sum() {
315
316     // Ελέγχει αν υπάρχουν δεδομένα
317     // Αν δεν υπάρχουν, εμφανίζει κατάλληλο μήνυμα στο
318     // stderr και επιστρέφει 0
319     if ( !this.isInit() ) {
320         System.err.println("Data isn't been initialized");
321         return 0;
322     }
323
324     // Αρχικοποιεί το sum με 0
325     int sum = 0;
326
327     // Προσθέτει στο sum όλα τα στοιχεία του πίνακα
328     for( int tmp : data) {
329         sum += tmp;
330     }
331     return sum;
332 }
333
334 // Επιστρέφει την μέση τιμή η οποία είναι double
335 public double mean() {
336
337     // Ελέγχει αν υπάρχουν δεδομένα
338     // Αν δεν υπάρχουν, εμφανίζει κατάλληλο μήνυμα στο
339     // stderr και επιστρέφει 0.0
340     if ( !this.isInit() ) {
341         System.err.println("Data isn't been initialized");
342         return 0.0;
343     }
344
345     // Γίνεται κλήση της μεθόδου sum για να είναι ο αριθμητής και
346     // ο παρονομαστής είναι το attribute length του πίνακα
347     // όμως είναι και οι δύο ακέραιοι με αποτέλεσμα να επιστρέφει
348     // ακέραιο και για αυτό κάνουμε cast double τουλάχιστον ένα
349     // από τα δύο
350     return (double) this.sum() / data.length;
351 }
352
```

```
353
354 // Επιστρέφει το percentile των δεδομένων
355 public double percentiles(double percentile) {
356
357     // Ελέγχει αν η τιμή που δόθηκε ως percentile
358     // είναι 0 < p <= 100. Αν δεν είναι εμφανίζεται κατάλληλο
359     // μήνυμα στο stderr και επιστρέφει 0.0
360     if ( percentile <= 0 || percentile > 100 ) {
361         System.err.println("percentile must be 0 < p <= 100");
362         return 0.0;
363     }
364
365     // Ελέγχει αν υπάρχουν δεδομένα
366     // Αν δεν υπάρχουν, εμφανίζει κατάλληλο μήνυμα στο
367     // stderr και επιστρέφει 0.0
368     if ( !this.isInit() ) {
369         System.err.println("Data isn't been initialized");
370         return 0.0;
371     }
372
373     // Αρχικοποιεί το size με το μέγεθος των δεδομένων
374     int size = data.length;
375
376     // Αν το μέγεθος είναι 1, τότε απλά επιστρέφει
377     // το μοναδικό στοιχείο
378     if (size == 1) {
379         return data[0];
380     }
381
382     // Ελέγχει αν είναι ταξινομημένα τα στοιχεία
383     // Αν δεν είναι τότε τα ταξινομεί
384     if ( !this.getSorted() )
385         this.sortData();
386
387     // index είναι το στοιχείο το οποίο εμφανίζεται για
388     // το συγκεκριμένο percentile το οποίο αρχικοποιείται
389     // με την formula
390     // είναι ο πολλαπλασιασμός του percentile
391     // διά 100 με το πλήθος των στοιχείων μείον 1 συν 1
392     double index = (percentile/100) * (size-1) + 1;
393
394     // κάνουμε cast σε int και το αφαιρούμαι από τον εαυτό του
395     // για να δούμε αν υπάρχει τιμή μετά την κινητή υποδιαστολή
396     double diff = index - (int) index;
```

```
397
398 // αν δεν υπάρχει διαφορά, τότε επιστρέφει το στοιχείο
399 if ( diff == 0.0)
400     return data[(int)index-1];
401
402 // Εφόσον το index δεν είναι ακέραιος, πρέπει να βρούμε
403 // μια ενδιάμεση τιμή ανάμεσα στο στοιχείο που δείχνει και
404 // στο επόμενο
405
406 // Αρχικοποιούμε τα στοιχεία με το παρών και το επόμενο
407 // στις μεταβλητές current και next αντίστοιχα
408 int current = data[(int)index-1];
409 int next = data[(int)index];
410
411 // Επιστρέφουμε την τιμή που βρίσκεται από την formula
412 // πολλαπλασιάζοντας την διαφορά με την αφαίρεση των δύο
413 // στοιχείων και προστίθενται το παρών στοιχείο
414 return diff * (next - current) + current;
415 }
416
417
418 // Προσθέτει ένα νέο δεδομένο στο τέλος του πίνακα
419 // Ακριβώς από κάτω από αυτήν την μέθοδο, υπάρχει με σχόλια
420 // η ίδια μέθοδος αλλά με διαφορετικό τρόπο αντιμετώπισης
421 // στην περίπτωση που δεν έχει αρχικοποιηθεί και το data είναι null
422 public void addData(int value) {
423
424     // Ελέγχει αν έχει αρχικοποιηθεί. Αν δεν έχει τότε δημιουργεί
425     // έναν νέο πίνακα μεγέθους ενός ακεραίου, τοποθετεί την τιμή
426     // και αλλάζει την τιμή init σε true
427     // Αλλιώς καλεί το Arrays.copyOf για να μεγαλώσει τον πίνακα
428     // κατά 1 και τοποθετεί στο τέλος την τιμή
429     if ( !this.isInit() ) {
430         data = new int[1];
431         data[0] = value;
432         init = true;
433     }
434     else {
435
436         data = Arrays.copyOf(data, data.length + 1);
437         data[data.length-1] = value;
438     }
439 }
440
```

```
441 //public void addData(int value) {
442
443     //try {
444
445         //data = Arrays.copyOf(data, data.length + 1);
446         //data[data.length-1] = value;
447
448     //} catch (NullPointerException e) {
449         //data = new int[1];
450         //data[0] = value;
451     //}
452 //}
453
454
455 // Ταξινομεί τα δεδομένα αφού ελέγξει πως δεν είναι ήδη ταξινομημένα
456 // υπάρχει σε σχόλια ο τρόπος ταξινόμησης με selection sort, όμως
457 // έκανε τον διπλάσιο χρόνο και για αυτό χρησιμοποιείται
458 // η μέθοδος Arrays.sort
459 public void sortData() {
460
461     // Ελέγχει αν υπάρχουν δεδομένα
462     // Αν δεν υπάρχουν, εμφανίζει κατάλληλο μήνυμα στο
463     // stderr και επιστρέφει
464     if( !this.isInit() ) {
465         System.err.println("Data isn't initialized");
466         return;
467     }
468
469     // η τιμή sorted γίνεται true
470     this.sorted = true;
471
472
473     //real 0m4.595s
474     //user 0m4.557s
475     //sys 0m0.303s
476
477     // Ταξινομείται από την κλάση Arrays
478     Arrays.sort(this.data);
479
480
481     //real 0m8.014s
482     //user 0m7.976s
483     //sys 0m0.316s
484
```



```
485 // Αρχικοποιείται με το μέγεθος του πίνακα
486 //int n = data.length;
487
488 // μεταφέρει ένα προς ένα το μέγεθος του
489 // μη ταξινομημένου υπό πίνακα
490 //for(int i = 0; i < n-1; i++) {
491
492     // βρίσκει το ελάχιστο στοιχείο του
493     // μη ταξινομημένου πίνακα
494     //int min_idx = i;
495     //for(int j = i+1; j < n; j++)
496         //if( data[j] < data[min_idx])
497             //min_idx = j;
498
499     // Αλλάζει το μικρότερο που βρήκε με αυτό που είναι
500     // πρώτο στον μη ταξινομημένο υπό πίνακα
501     //int temp = data[min_idx];
502     //data[min_idx] = data[i];
503     //data[i] = temp;
504 //}
505 }
506
507
508
509
510 // Ανοίγει το κανάλι για είσοδο
511 public boolean openInStream() {
512
513     // Ελέγχει αν υπάρχει όνομα για το αρχείο εισόδου
514     // Αν δεν υπάρχει εμφανίζει κατάλληλο μήνυμα στο
515     // stderr και επιστρέφει false
516     if ( inName == null ) {
517         System.err.println("No inName");
518         return false;
519     }
520
521     // Κλείνει το κανάλι αν είναι ανοιχτό
522     this.closeInStream();
523
524     // Ανοίγει το κανάλι εισόδου και προσπαθεί να πιάσει
525     // FileNotFoundException exception. Αν το ανοίξει θα επιστρέψει
526     // true αλλιώς false
527     try {
528         inStream = new BufferedReader(new FileReader(inName));
```

```
529     } catch (FileNotFoundException e) {
530         System.err.println("File not found " + e.getMessage());
531         return false;
532     }
533     return true;
534 }
535
536
537
538
539 // Κλείνει το κανάλι εισόδου
540 public boolean closeInStream() {
541
542     // Ελέγχει αν υπάρχει κανάλι. Αν δεν υπάρχει τότε θα
543     // επιστρέφει true. Αν υπάρχει θα προσπαθήσει να το κλείσει
544     // και θα εκχωρήσει στο inStream null
545     if (inStream != null) {
546         try {
547             inStream.close();
548             inStream = null;
549         } catch (IOException e) {
550             System.err.println("Stream couldn't close " + e.getMessage());
551             return false;
552         }
553     }
554     return true;
555 }
556
557 // Ανοίγει το κανάλι εξόδου
558 public boolean openOutStream() {
559
560     // Ελέγχει αν υπάρχει όνομα για το αρχείο εξόδου
561     // Αν δεν υπάρχει εμφανίζει κατάλληλο μήνυμα στο
562     // stderr και επιστρέφει false
563     if ( outName == null ) {
564         System.err.println("No outName");
565         return false;
566     }
567
568     // Κλείνει το κανάλι αν είναι ανοιχτό
569     this.closeOutStream();
570
571     // Ανοίγει το κανάλι εξόδου και προσπαθεί να πιάσει
572     // FileNotFoundException. Αν το ανοίξει θα επιστρέψει
```

```
573 // true αλλιώς false
574 try {
575     outputStream = new PrintWriter(outName);
576 } catch (FileNotFoundException e) {
577     System.err.println("File not found " + e.getMessage());
578     return false;
579 }
580
581 return true;
582 }
583
584
585
586
587 // κλείνει το κανάλι εξόδου
588 public void closeOutputStream() {
589
590     // Το close δεν πετάει κάποιο exception και για
591     // αυτό δεν βρίσκεται σε try-catch block
592     // Ελέγχει αν δεν είναι null το outputStream και
593     // αν δεν είναι τότε το κλείνει και εκχωρεί στο
594     // outputStream null.
595     if (outputStream != null) {
596         outputStream.close();
597         outputStream = null;
598     }
599 }
600
601 }
```

Κώδικας 4.1: Κώδικας Stats

5 SQLite

5.1 Βάση δεδομένων stats.db

Για να κατασκευάσει κάποιος μια βάση δεδομένων στην SQLite, αρκεί όταν τρέξει το εκτελέσιμο αρχείο sqlite3 να του δώσει ως όρισμα το όνομα μιας βάσης δεδομένων που δεν υπάρχει.

```
sqlite3 stats.db
```

Για να εισάγουμε τα δεδομένα, πρέπει να φτιάξουμε τον κατάλληλο πίνακα και το επιτυγχάνουμε όπως κάνουμε και σε κάθε άλλη βάση.

```
CREATE TABLE NUMBERS (Number int(16));
```

Έπειτα, με τις ακόλουθες εντολές, δίνετε η οδηγία πως το αρχείο είναι csv και κάνει εισαγωγή των δεδομένων του αρχείου numbers.txt στον πίνακα NUMBERS.

```
.mode csv  
.import numbers.txt NUMBERS
```

Τα queries για την ταξινόμηση, το μέγεθος των δεδομένων, το ελάχιστο στοιχείο, το μέγιστο στοιχείο και το μέσο όρο είναι τα ακόλουθα.

```
# size  
SELECT count(*) FROM NUMBERS;  
  
# min  
SELECT min(Number) FROM NUMBERS;  
  
# max  
SELECT max(Number) FROM NUMBERS;  
  
# avg  
SELECT avg(Number) FROM NUMBERS;
```

5.2 Κώδικας Stats_SQLite

Ο κώδικας 5.1 δέχεται ως όρισμα το όνομα μίας υπάρχουσας βάσης δεδομένων. Το πρόγραμμα συνδέεται με την βάση και υπολογίζει την διάμεση τιμή.

Το ακόλουθο query ταξινομεί τα records του πίνακα numbers σε αύξουσα σειρά βάση την στήλη number με το **ORDER BY number ASC** και με την εντολή **LIMIT n, x** εμφανίζει x στοιχεία από το n-οστό στοιχείο του πίνακα.

```
SELECT * FROM numbers  
ORDER BY number ASC  
LIMIT 5, 1;
```

```
padelis@MAE-project1$ java -cp ".:sqlite-jdbc-3.27.2.1.jar" Stats_sqlite
Usage: <database name>
padelis@MAE-project1$ java -cp ".:sqlite-jdbc-3.27.2.1.jar" Stats_sqlite stats.db
Connected to database: stats.db
The number for percentile 50.0 is: 8386878.5

Goodbye!
padelis@MAE-project1$
```

Σχήμα 5.1: Ενδεικτικό τρέξιμο του προγράμματος Stats_sqlite

```
1 import java.sql.Connection;
2 import java.sql.DriverManager;
3 import java.sql.ResultSet;
4 import java.sql.SQLException;
5 import java.sql.Statement;
6
7 public class Stats_sqlite {
8
9     public static void main(String[] args) {
10
11         // Ελέγχει αν υπάρχει τουλάχιστον ένα όρισμα
12         if (args.length < 1) {
13             System.out.println("Usage: <database name>");
14             System.exit(0);
15         }
16
17         Connection conn = null;
18
19         try {
20
21             // Σύνδεση με την βάση
22             conn = DriverManager.getConnection("jdbc:sqlite:" + args[0]);
23             System.out.println("Connected to database: " + args[0]);
24
25             // Δημιουργία statement obj για αποστολή statement
26             Statement stm = conn.createStatement();
27
28             // Θέτει την αναμονή 1 λεπτό μέχρι να λάβει απάντηση
29             stm.setQueryTimeout(60);
30
31             // Μέγεθος των δεδομένων
32             ResultSet rs = stm.executeQuery("SELECT count(*)" +
33                 " AS foo FROM numbers");
34
35             // Τοποθετεί το μέγεθος των δεδομένων στο size
```

```
36     int size = rs.getInt("foo");
37
38     // Αν δεν υπάρχουν δεδομένα εμφανίζει κατάλληλο μήνυμα
39     // αλλιώς αν υπάρχει μόνο ένα το εμφανίζει
40     if( size == 0) {
41         System.out.println("There are no records");
42         System.exit(0);
43     }
44     else if ( size == 1) {
45         System.out.println("There is only one record");
46         rs = stm.executeQuery("SELECT * FROM numbers");
47         System.out.println(rs.getInt("number"));
48         System.exit(0);
49     }
50
51     // percentile must be 0 < p <= 100
52     double percentile = 50;
53
54     // Η formula για τα percentile όπως στο Stats.java
55     double index = (percentile/100) * (size-1) + 1;
56
57     double diff = index - (int) index;
58
59     if ( diff == 0.0) {
60
61         // Επιστρέφει το nth αντικείμενο με την χρήση του LIMIT
62         rs = stm.executeQuery("SELECT * FROM numbers "
63             + "ORDER BY number ASC LIMIT "
64             + (int) (index -1) + ", 1");
65
66         System.out.println("The number for percentile "
67             + percentile + " is: " + rs.getInt("number"));
68
69         System.exit(0);
70     }
71
72     // Επιστρέφει το nth και nth+1 records με την
73     // χρήση του LIMIT
74     rs = stm.executeQuery("SELECT * FROM numbers "
75         + "ORDER BY number ASC LIMIT "
76         + (int) (index -1) + ", 2");
77
78     // Ξέρουμε πως έχουμε ζητήσει 2 records
79     // και κανονικά πρέπει να έχουν επιστραφεί 2
```

```
80      // αλλιώς θα μπορούσαμε να κάνουμε
81
82      //int sub = 0;
83      //int tmp = 0;
84
85      //while (rs.next()) {
86          //tmp = rs.getInt(1);
87          //sub = tmp - sub;
88      //}
89      //System.out.println("The number for percentile "
90          //+ percentile + " is: "
91          //+ (diff * sub + tmp - sub));
92
93      // Μετακινούμε τον κέρσορα από default position στην
94      // πρώτη γραμμή
95      rs.next();
96
97      int current = rs.getInt(1);
98
99      rs.next();
100
101      int next = rs.getInt(1);
102
103      System.out.println("The number for percentile "
104          + percentile + " is: "
105          + (diff * (next - current) + current));
106  }
107  catch(SQLException e) {
108      System.out.println(e.getMessage());
109  }
110
111  try {
112      if (conn != null) {
113          conn.close();
114      }
115  } catch(SQLException ex) {
116      System.out.println(ex.getMessage());
117  }
118
119  System.out.println("\nGoodbye!");
120
121  }
```

122 }

Κώδικας 5.1: Κώδικας Stats_sqlite