

ViewModel.java

```
1 package expenses.manager;
2
3 import static expenses.manager.Log.logger;
4 /**
5  *
6  * @author Avital Chen
7  * ViewModel class
8  */
9
10 public class ViewModel implements IViewModel {
11     private IModel model;
12     private IView view;
13     @Override
14     public void setIncome(String date, double amount) {
15         new Thread( new Runnable() {
16
17             @Override
18             public void run() {
19                 try {
20                     model.setIncome(date, amount);
21                 } catch (Exception e) {
22                     e.printStackTrace();
23                 }
24             }
25         } ).start();
26     }
27
28     @Override
29     public void setExpenses(String type, String date, double
30 amount) {
31         new Thread( new Runnable() {
32             @Override
33             public void run() {
34                 try {
35                     model.setExpenses(type, date, amount);
36                 } catch (Exception e) {
37                     e.printStackTrace();
38                 }
39             }
40         } ).start();
41     }
42
43     @Override
44     public void getTotalExpensesByTypeCurrMonth() {
45         new Thread( new Runnable() {
```

ViewModel.java

```

53
54         @Override
55         public void run() {
56
57             try {
58                 Vector data = model.getTotalExpensesByType();
59                 SwingUtilities.invokeLater(new Runnable() {
60
61                     @Override
62                     public void run() {
63                         try {
64
65                             view.getExpendsesByTypeCurrYear(data);
66                             } catch (Exception e) {
67
68                                 e.printStackTrace();
69                             }
70                         }));
71                     } catch (Exception e) {
72                         e.printStackTrace();
73                     }
74                 }
75             } ).start();
76
77     }
78
79     @Override
80     public void setModel(IModel m) {
81         this.model = m;
82
83     }
84
85     @Override
86     public void setView(IView v) {
87         this.view = v;
88
89     }
90
91     @Override
92     public void getTableInfo() {
93         new Thread( new Runnable() {
94
95             @Override
96             public void run() {
97                 try {
98                     ArrayList<Double> infoArray = new
99 ArrayList<Double>();

```

ViewModel.java

```
100         infoArray.add(0,
model.getTotalIncomeOverall());
101
102         infoArray.add(1,
model.getTotalExpensesOverall());
103         infoArray.add(2,
model.getTotalIncomeCurrYear());
104         infoArray.add(3,
model.getTotalExpensesCurrYear());
105
106
107         SwingUtilities.invokeLater(new Runnable() {
108
109             @Override
110             public void run() {
111                 try {
112                     view.getTableInfo(infoArray);
113                 } catch (Exception e) {
114                     e.printStackTrace();
115                 }
116
117             }
118         });
119     } catch (Exception e) {
120         e.printStackTrace();
121     }
122 }
123 }.start();
124
125 }
126
127 @Override
128 public void getInclomeAndExpensesByMonth() {
129     new Thread( new Runnable() {
130
131         @Override
132         public void run() {
133             try {
134                 Vector data = new Vector();
135                 data.add(0, model.getTotalInclomeByMonth());
136                 data.add(1, model.getTotalExpensesByMonth());
137                 SwingUtilities.invokeLater(new Runnable() {
138
139                     @Override
140                     public void run() {
141                         try {
142                             view.getIncomeAndExpensesByMonth(data);
143                         } catch (Exception e) {
```

ViewModel.java

```

144         e.printStackTrace();
145     }
146     });
147
148     } catch (Exception e) {
149         e.printStackTrace();
150     }
151 }
152 } ).start();
153
154 }
155
156
157 @Override
158 public void getTotalExpensesByTypeCurrYear() {
159     new Thread( new Runnable() {
160
161         @Override
162         public void run() {
163             try {
164                 Vector data =
model.getTotalExpensesByTypeCurrMonth();
165                 SwingUtilities.invokeLater(new Runnable() {
166
167                     @Override
168                     public void run() {
169                         try {
170
view.getExpensesByTypeCurrMonth(data);
171                             } catch (Exception e) {
172                                 e.printStackTrace();
173                             }
174                         });
175
176                     } catch (Exception e) {
177                         e.printStackTrace();
178                     }
179                 }
180             } ).start();
181
182
183         }
184
185 }
186

```