

# TaxpayerAccountsManager.java

```

1 package accountsManager;
2
3 import java.util.ArrayList;
4 import inputManager.InputManager;
5 import taxpayer.Taxpayer;
6 import outputManager.OutputManager;
7 import receipt.Receipt;
8
9 public class TaxpayerAccountsManager {
10     ArrayList<Taxpayer> taxpayersList = new ArrayList<Taxpayer>();
11     InputManager inputManager = new InputManager();
12     OutputManager outputManager = new OutputManager();
13
14     public void deleteTaxpayer(Taxpayer taxpayerForDelete){
15         taxpayersList.remove(taxpayerForDelete);
16     }
17
18     public void deleteTaxpayer(int Afm){
19         for(int i=0; i<taxpayersList.size(); i++){
20             if (taxpayersList.get(i).getAfm() == Afm){
21                 taxpayersList.remove(i);
22             }
23             break;
24         }
25     }
26
27     public void deleteReceiptFromTaxpayer(
28         Taxpayer taxpayerNeedingToDeleteReceipt, int receiptCode){
29         ArrayList<Receipt> receipts =
30             taxpayerNeedingToDeleteReceipt.getReceipts();
31         for(int i=0; i<receipts.size(); i++){
32             if(receipts.get(i).getReceiptCode() == receiptCode){
33                 receipts.remove(i);
34                 return;
35             }
36         }
37         System.out.println("!!!Receipt don't found!!!");
38     }
39
40     public void insertNewReceiptForTaxpayer(
41         Taxpayer taxpayerNeedingNewReceipt, String category,
42         int receiptCode, String dateOfIssue, int amountPaid,
43         String companyName, String country, String city,
44         String street, int streetNumber)
45     {
46         Receipt newReceipt = new Receipt();
47         ArrayList<Receipt> receipts = taxpayerNeedingNewReceipt.
48             getReceipts();
49         newReceipt.initializeReceipt(category, receiptCode,
50             dateOfIssue, amountPaid, companyName, country, city,
51             street, streetNumber);
52         receipts.add(newReceipt);
53         taxpayerNeedingNewReceipt.setReceipts(receipts);
54     }
55
56     public Taxpayer getTaxpayer(String taxpayersName){
57         for(int i=0; i<taxpayersList.size(); i++){
58             if(taxpayersList.get(i).getName().equals(taxpayersName)){
59                 return taxpayersList.get(i);
60             }
61         }
62         System.out.println("Error: Can't find taxpayer !!!");

```

```

63     return null;
64 }
65
66 public Taxpayer getTaxpayer(int taxpayerAfm){
67     for(int i=0;i<taxpayersList.size(); i++){
68         if(taxpayersList.get(i).getAfm() == taxpayerAfm){
69             return taxpayersList.get(i);
70         }
71     }
72     System.out.println("Error: Cant't find taxpayer !!!");
73     return null;
74 }
75
76 public void insertTaxpayerFromFile(String filePath){
77     if(filePath.endsWith(".txt")){
78         inputManager.createTaxpayerDataListFromTextFile(filePath);
79     }else if(filePath.endsWith(".xml")){
80         inputManager.createTaxpayerDataListFromXmlFile(filePath);
81     }
82     taxpayersList.add(inputManager.getTaxpayerFromFileDataList());
83 }
84
85 public void updateInfoFiles(Taxpayer taxp){
86     outputManager.updateInfoFiles(taxp);
87 }
88
89 public void createLogXmlFile(Taxpayer taxp){
90     outputManager.createLogXmlFile(taxp);
91 }
92
93 public void createLogTextFile(Taxpayer taxp){
94     outputManager.createLogTextFile(taxp);
95 }
96
97 public void createBarChart(Taxpayer taxp){
98     outputManager.createBarChart(taxp);
99 }
100
101 public void createPieChart(Taxpayer taxp){
102     outputManager.createPieChart(taxp);
103 }
104
105 public void printTaxpayer(Taxpayer taxpayerForPrint){
106     taxpayerForPrint.toString();
107 }
108
109 public void setDomain(String domain){
110     outputManager.setDomain(domain);
111 }
112 }
113

```