

## Практическое занятие №2

Разработка интерактивного графического приложения с использованием REST API

Выполнил: Калмыков Семён, группа 403

### Ход работы:

1. Создание интерфейса приложения. Данный процесс представлен на рисунке 1-2.

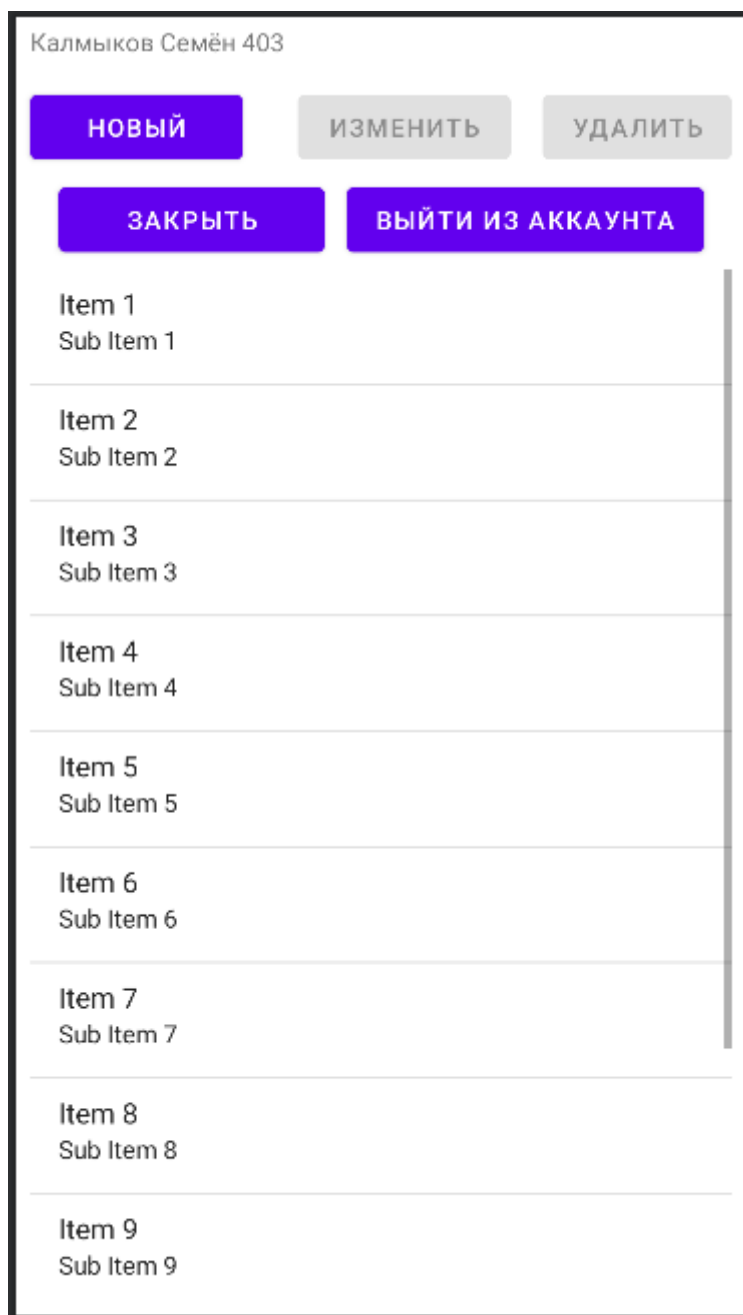


Рисунок 1– Интерфейс activity\_main.xml

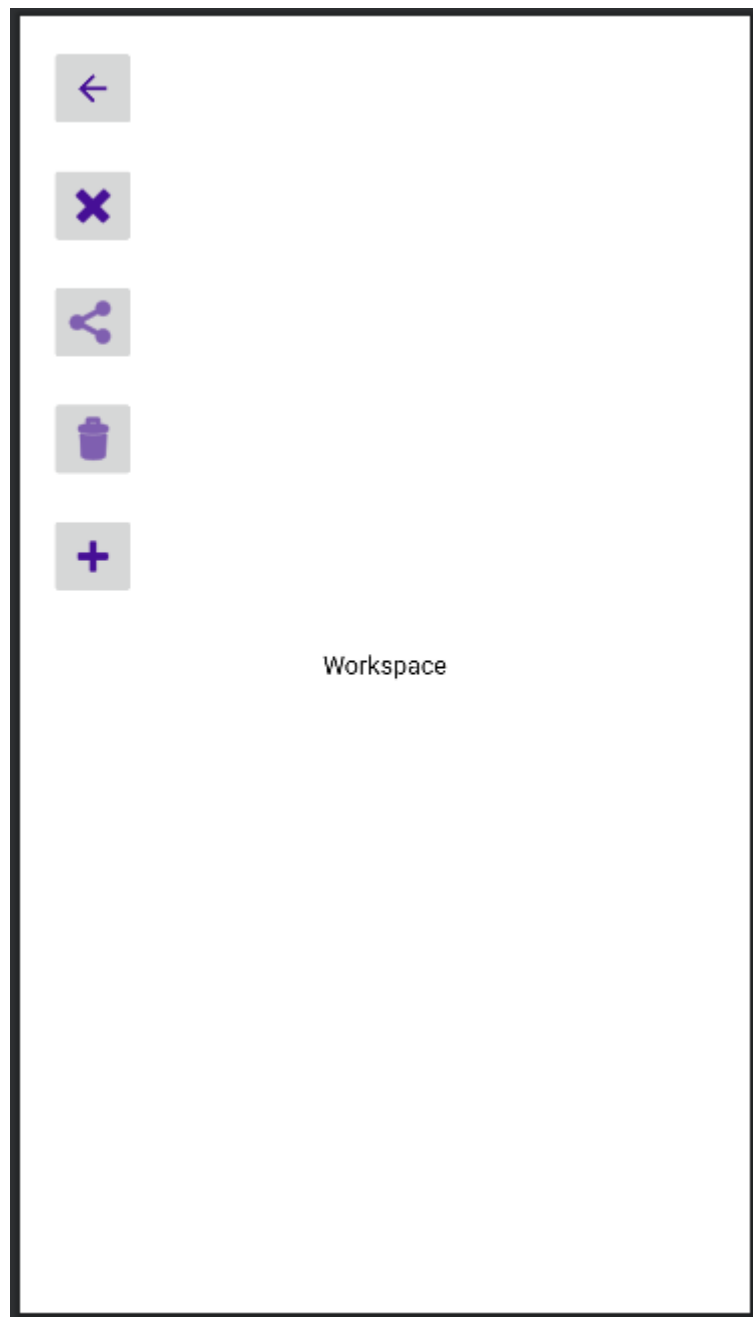


Рисунок 2 – Интерфейс activity\_second.xml

2. На рисунке 3 представлены компоненты приложения из activity\_main.xml.

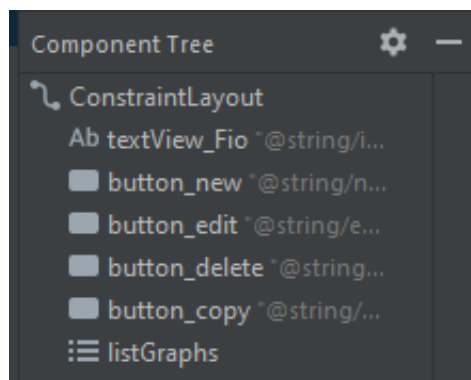


Рисунок 3 – Компоненты приложения activity\_main.xml

3. На рисунке 4 представлены компоненты приложения из activity\_second.xml

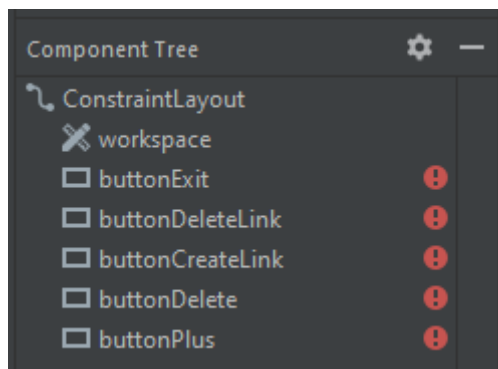


Рисунок 4 – Компоненты приложения activity\_second.xml

4. Создание строковых ресурсов в файле strings.xml, для подписей элементов интерфейса. Это изображено на рисунке 5.

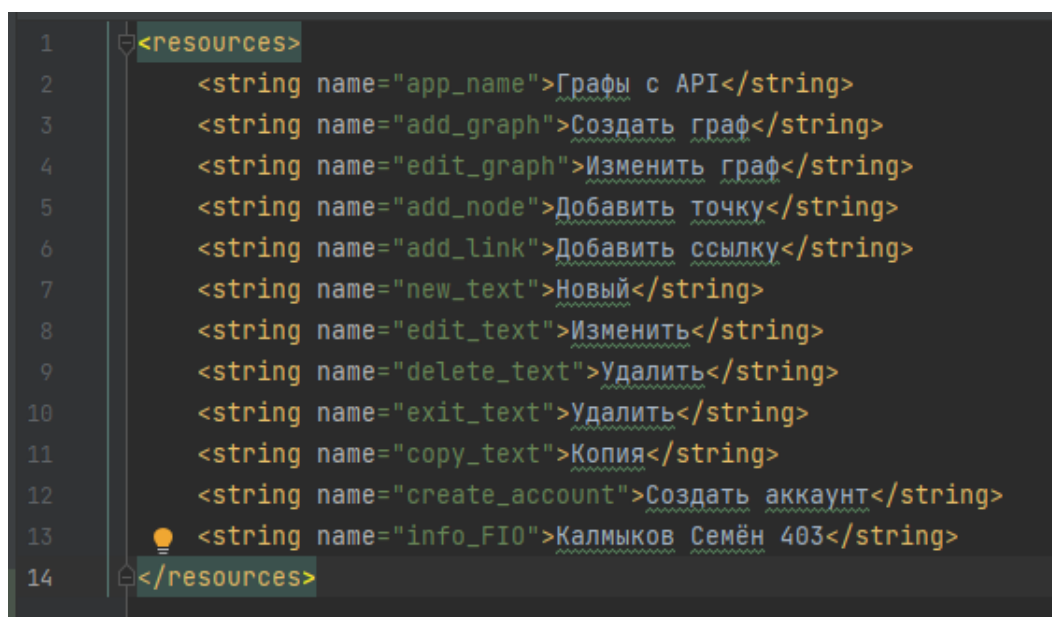


Рисунок 5 – Строковые ресурсы

5. В файле MainActivity.java написана логика интерфейса activity\_main.xml. Это изображено на рисунках 6 – 10.

```
1 package example.kalmykov403.graphs;
2
3 import ...
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23 public class MainActivity extends AppCompatActivity {
24     private ActivityMainBinding binding = null;
25
26     private ArrayAdapter<GraphItem> graphsAdapter;
27     private GraphAPI api = ApiBuilder.getAPI();
28     private SharedPreferencesRepository repository;
29     private MainActivity mainActivity;
30
31     private boolean editEnable = false;
32     private boolean deleteEnable = false;
33     private boolean copyEnable = false;
34
35     @Override
36     protected void onCreate(Bundle savedInstanceState) {
37         super.onCreate(savedInstanceState);
38         binding = ActivityMainBinding.inflate(getLayoutInflater());
39         setContentView(binding.getRoot());
40         repository = new SharedPreferencesRepository(this.getApplicationContext());
41         mainActivity = this;
42         graphsAdapter = new ArrayAdapter<>(context: this, android.R.layout.simple_list_item_1);
43
44         binding.listGraphs.setAdapter(graphsAdapter);
45         binding.listGraphs.setOnItemClickListener((parent, view, position, id) -> {
46             if (!editEnable && !deleteEnable && !copyEnable) {
47                 Intent i = new Intent(getApplicationContext(), SecondActivity.class);
48                 i.putExtra(name: "GRAPH_ID", graphsAdapter.getItem(position).id);
49                 startActivity(i);
50             } else if (editEnable) {
```

Рисунок 6 – Код MainActivity.java

```

50     } else if (editEnable) {
51         openEditGraphDialog(graphsAdapter.getItem(position).id);
52         editEnable = false;
53     } else {
54         api.graphDelete(repository.getToken(), graphsAdapter.getItem(position).id).enqueue(new Callback<Void>() {
55             @Override
56             public void onResponse(Call<Void> call, Response<Void> response) {
57                 deleteEnable = false;
58                 refreshGraphs();
59             }
60
61             @Override
62             public void onFailure(Call<Void> call, Throwable t) {}
63         });
64     }
65
66 });
67
68 refreshGraphs();
69
70 binding.buttonNew.setOnClickListener(v -> {
71     openAddGraphDialog();
72 });
73
74 binding.buttonEdit.setOnClickListener(v -> {
75     deleteEnable = false;
76     editEnable = true;
77     copyEnable = false;
78 });
79
80 binding.buttonDelete.setOnClickListener(v -> {
81     deleteEnable = true;
82     editEnable = false;
83     copyEnable = false;
84 });
85
86 binding.buttonExit.setOnClickListener(v -> {
87     api.sessionDelete(repository.getToken()).enqueue(new Callback<Void>() {
88         @Override
89         public void onResponse(Call<Void> call, Response<Void> response) {

```

Рисунок 7 – Код MainActivity.java

```

89  public void onResponse(Call<Void> call, Response<Void> response) {
90      MainActivity.finishAffinity();
91  }
92
93  @Override
94  public void onFailure(Call<Void> call, Throwable t) {
95
96  }
97  });
98  });
99  binding.buttonDeleteAccount.setOnClickListener(v -> {
100      api.accountDelete(repository.getToken()).enqueue(new Callback<Void>() {
101          @Override
102          public void onResponse(Call<Void> call, Response<Void> response) {
103              repository.saveAccountName("");
104              repository.saveAccountSecret("");
105              repository.saveToken("");
106              Intent i = new Intent(getApplicationContext(), RegistrationActivity.class);
107              startActivity(i);
108          }
109
110          @Override
111          public void onFailure(Call<Void> call, Throwable t) {
112
113          }
114      });
115  });
116  }
117
118  4 usages
119  private void refreshGraphs() {
120      graphsAdapter.clear();
121      api.graphList(repository.getToken()).enqueue(new Callback<List<GraphItem>>() {
122          @Override
123          public void onResponse(Call<List<GraphItem>> call, Response<List<GraphItem>> response) {
124              graphsAdapter.addAll(response.body());
125
126              binding.buttonEdit.setEnabled(!graphsAdapter.isEmpty());
127              binding.buttonDelete.setEnabled(!graphsAdapter.isEmpty());
128          }
129      });
130  }

```

Рисунок 8 – Код MainActivity.java

```

127     }
128
129     @Override
130     public void onFailure(Call<List<GraphItem>> call, Throwable t) {
131     }
132 }
133 });
134 }
135
136 1 usage
137 private void openAddGraphDialog() {
138     AlertDialog dialog = new AlertDialog.Builder( context: this).create();
139
140     View dialogView = getLayoutInflater().inflate(R.layout.add_node_dialog, root: null);
141     AddNodeDialogBinding dialogBinding = AddNodeDialogBinding.bind(dialogView);
142
143     dialogBinding.tvTitle.setText("Создать граф");
144
145     dialogBinding.btnAdd.setOnClickListener(view -> {
146         if (!dialogBinding.etName.getText().toString().isEmpty()) {
147             api.graphCreate(repository.getToken(), dialogBinding.etName.getText().toString()).enqueue(new Callback<IdDTO>() {
148                 @Override
149                 public void onResponse(Call<IdDTO> call, Response<IdDTO> response) {
150                     refreshGraphs();
151                     dialog.dismiss();
152                 }
153
154                 @Override
155                 public void onFailure(Call<IdDTO> call, Throwable t) {}
156             });
157         }
158     });
159     dialog.setView(dialogView);
160     dialog.getWindow().setLayout( width: 300, height: 300);
161     dialog.show();
162 }
163
164 1 usage
165 private void openEditGraphDialog(Integer graphId) {

```

Рисунок 9 – Код MainActivity.java

```

163 private void openEditGraphDialog(Integer graphId) {
164     AlertDialog dialog = new AlertDialog.Builder( context: this).create();
165
166     View dialogView = getLayoutInflater().inflate(R.layout.add_node_dialog, root: null);
167     AddNodeDialogBinding dialogBinding = AddNodeDialogBinding.bind(dialogView);
168
169     dialogBinding.tvTitle.setText("Изменить граф");
170
171     dialogBinding.btnAdd.setOnClickListener(view -> {
172         if (!dialogBinding.etName.getText().toString().isEmpty()) {
173             api.graphUpdate(repository.getToken(), graphId, dialogBinding.etName.getText().toString()).enqueue(new Callback<Void>() {
174                 @Override
175                 public void onResponse(Call<Void> call, Response<Void> response) {
176                     refreshGraphs();
177                     dialog.dismiss();
178                 }
179
180                 @Override
181                 public void onFailure(Call<Void> call, Throwable t) {}
182             });
183         }
184     });
185     dialog.setView(dialogView);
186     dialog.getWindow().setLayout( width: 300, height: 300);
187     dialog.show();
188 }
189 }

```

Рисунок 10 – Код MainActivity.java

6. В файле SecondActivity.java написана логика интерфейса activity\_second.xml. Это изображено на рисунках 11 – 16.

```

1 package example.kalmykov403.graphs;
2
3 import ...
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22 public class SecondActivity extends AppCompatActivity {
23     12 usages
24     private ActivitySecondBinding binding = null;
25     4 usages
26     private Integer graphId;
27     5 usages
28     private boolean isConnection = false;
29     5 usages
30     private boolean isDelete = false;
31     5 usages
32     private boolean isDeleteLink = false;
33     10 usages
34     private NodeItem firstNode = null;
35
36     7 usages
37     private GraphAPI api = ApiBuilder.getAPI();
38     8 usages
39     private SharedPreferencesRepository repository;
40
41     @Override
42     protected void onCreate(Bundle savedInstanceState) {
43         super.onCreate(savedInstanceState);
44         binding = ActivitySecondBinding.inflate(getLayoutInflater());
45         setContentView(binding.getRoot());
46         repository = new SharedPreferencesRepository(this.getApplicationContext());
47
48         graphId = getIntent().getIntExtra("GRAPH_ID", defaultValue: 0);
49         binding.workspace.addListeners(
50             (newNode, newLinks) -> {
51                 api.nodeUpdate(repository.getToken(), newNode.id, newNode.x, newNode.y, newNode.name).enqueue(new Callback<Void>() {
52                     @Override
53                     public void onResponse(Call<Void> call, Response<Void> response) {
54
55                     }
56                 })
57             }
58         );
59     }
60
61     @Override

```

Рисунок 11 – Код SecondActivity.java



```

49         @Override
50         public void onFailure(Call<Void> call, Throwable t) {
51
52         }
53     });
54 },
55     node -> {
56         if (isConnection) {
57             if (firstNode == null) {
58                 firstNode = node;
59             } else {
60                 if (!binding.workspace.isLinkExists(firstNode, node)) {
61                     openAddLinkDialog(firstNode, node);
62                 }
63                 isConnection = false;
64                 firstNode = null;
65             }
66         }
67
68         if (isDelete) {
69             api.nodeDelete(repository.getToken(), node.id).enqueue(new Callback<Void>() {
70                 @Override
71                 public void onResponse(Call<Void> call, Response<Void> response) {
72                     refreshNodes();
73
74                     isDelete = false;
75                 }
76
77                 @Override
78                 public void onFailure(Call<Void> call, Throwable t) {
79
80                 }
81             });
82         }
83
84         if (isDeleteLink) {
85             if (firstNode == null) {
86                 firstNode = node;
87             } else {
88                 if (binding.workspace.isLinkExists(firstNode, node)) {

```

Рисунок 12 – Код SecondActivity.java

```

88         if(binding.workspace.isLinkExists(firstNode, node)) {
89             api.linkDelete(repository.getToken(), binding.workspace.findLink(firstNode, node).id).enqueue(new CallBack<Void>() {
90                 @Override
91                 public void onResponse(Call<Void> call, Response<Void> response) {
92                     isDeleteLink = false;
93                     firstNode = null;
94                     refreshNodes();
95                 }
96             }
97             @Override
98             public void onFailure(Call<Void> call, Throwable t) {
99                 }
100         });
101     }
102 }
103
104 };
105
106
107 refreshNodes();
108
109 binding.buttonExit.setOnClickListener( v -> {
110     Intent i = new Intent(getApplicationContext(), MainActivity.class);
111     startActivity(i);
112 });
113
114 binding.buttonPlus.setOnClickListener(v -> {
115     openAddNodeDialog();
116 });
117
118 binding.buttonDelete.setOnClickListener(v -> {
119     isDelete = true;
120     isConnection = false;
121     isDeleteLink = false;
122 });
123
124 binding.buttonCreateLink.setOnClickListener( v -> {
125     isConnection = true;
126     isDelete = false;
127     isDeleteLink = false;

```

Рисунок 13 – Код SecondActivity.java

```

127         isDeleteLink = false;
128     });
129
130     binding.buttonDeleteLink.setOnClickListener( v -> {
131         isDeleteLink = true;
132         isDelete = false;
133         isConnected = false;
134     });
135     }
136
137     5 usages
138     private void refreshNodes() {
139         ArrayList<LinkItem> links = new ArrayList<>();
140         ArrayList<NodeItem> nodes = new ArrayList<>();
141         api.nodeList(repository.getToken(), graphId).enqueue(new Callback<List<NodeItem>>() {
142             @Override
143             public void onResponse(Call<List<NodeItem>> call, Response<List<NodeItem>> response) {
144                 nodes.addAll(response.body());
145                 api.getLinks(repository.getToken(), graphId).enqueue(new Callback<List<LinkItem>>() {
146                     @Override
147                     public void onResponse(Call<List<LinkItem>> call, Response<List<LinkItem>> response) {
148                         links.addAll(response.body());
149                     }
150                 });
151                 binding.workspace.refreshPoints(nodes, links);
152             }
153             @Override
154             public void onFailure(Call<List<LinkItem>> call, Throwable t) {}
155         });
156     }
157
158     @Override
159     public void onFailure(Call<List<NodeItem>> call, Throwable t) {}
160
161     });
162
163     1 usage
164     private void openAddNodeDialog() {
165         AlertDialog dialog = new AlertDialog.Builder( context: this).create();

```

Рисунок 14 – Код SecondActivity.java

```

163     private void openAddNodeDialog() {
164         AlertDialog dialog = new AlertDialog.Builder(context: this).create();
165
166         View dialogView = getLayoutInflater().inflate(R.layout.add_node_dialog, root: null);
167         AddNodeDialogBinding dialogBinding = AddNodeDialogBinding.bind(dialogView);
168
169         dialogBinding.tvTitle.setText(R.string.add_node);
170
171         dialogBinding.btnAdd.setOnClickListener(view -> {
172             if (!dialogBinding.etName.getText().toString().isEmpty()) {
173                 String name = dialogBinding.etName.getText().toString();
174                 api.nodeCreate(repository.getToken(), graphId, x: 300f, y: 300f, name).enqueue(new Callback<IdDTO>() {
175                     @Override
176                     public void onResponse(Call<IdDTO> call, Response<IdDTO> response) {
177                         refreshNodes();
178                         dialog.dismiss();
179                     }
180
181                     @Override
182                     public void onFailure(Call<IdDTO> call, Throwable t) {
183                     }
184                 });
185             }
186         });
187         dialog.setView(dialogView);
188         dialog.show();
189     }
190
191     1 usage
192     private void openAddLinkDialog(NodeItem firstNode, NodeItem secondNode) {
193         AlertDialog dialog = new AlertDialog.Builder(context: this).create();
194
195         View dialogView = getLayoutInflater().inflate(R.layout.add_node_dialog, root: null);
196         AddNodeDialogBinding dialogBinding = AddNodeDialogBinding.bind(dialogView);
197
198         dialogBinding.tvTitle.setText(R.string.add_link);
199
200         dialogBinding.btnAdd.setOnClickListener(view -> {
201             if (!dialogBinding.etName.getText().toString().isEmpty()) {
202                 Float name = Float.parseFloat(dialogBinding.etName.getText().toString());
203                 api.linkCreate(repository.getToken(), firstNode.id, secondNode.id, name).enqueue(new Callback<IdDTO>() {

```

Рисунок 15 – Код SecondActivity.java

```

202         api.linkCreate(repository.getToken(), firstNode.id, secondNode.id, name).enqueue(new Callback<IdDTO>() {
203             @Override
204             public void onResponse(Call<IdDTO> call, Response<IdDTO> response) {
205                 refreshNodes();
206                 dialog.dismiss();
207             }
208
209             @Override
210             public void onFailure(Call<IdDTO> call, Throwable t) {
211             }
212         });
213     }
214
215     });
216
217     dialog.setView(dialogView);
218     dialog.show();
219 }
220 }

```

Рисунок 16 – Код SecondActivity.java