

## Практическое занятие №1

Приложение – для работы с графами

Выполнил: Калмыков Семён, группа 403

### Ход работы:

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

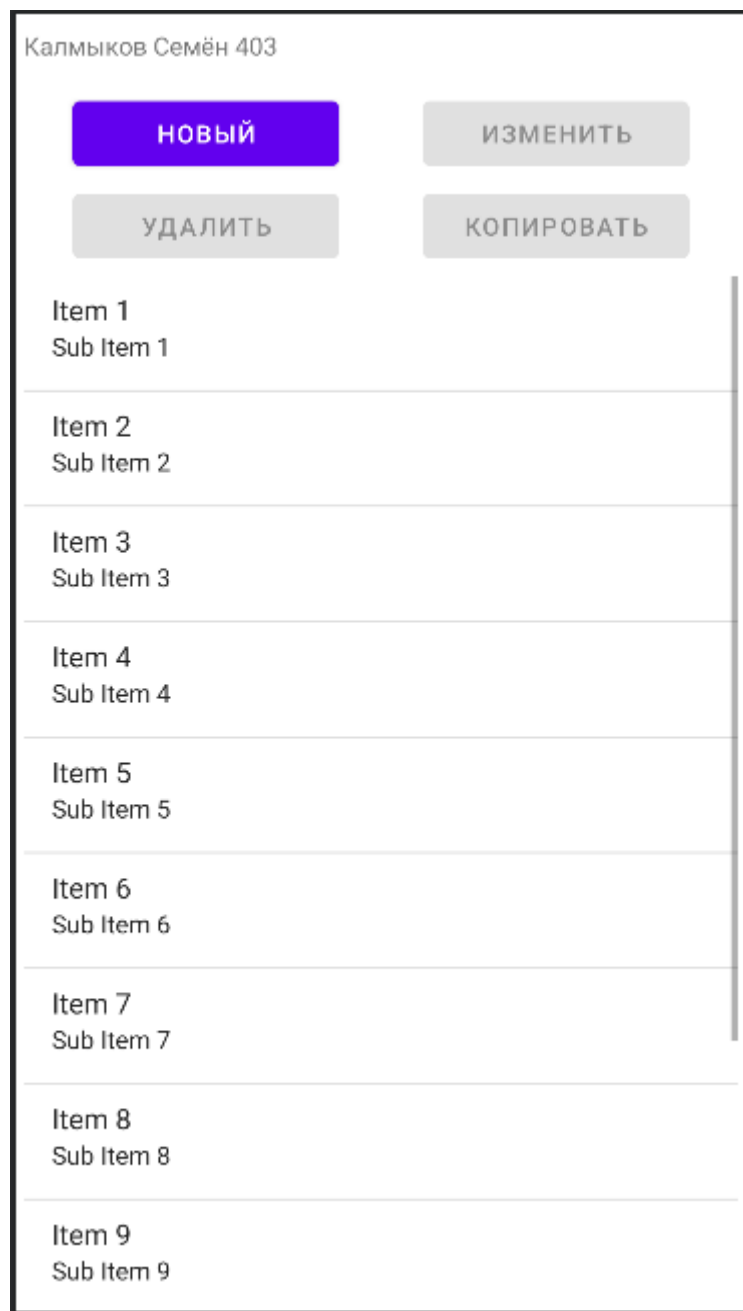


Рисунок 1– Интерфейс activity\_main.xml



Рисунок 2 – Интерфейс activity\_second.xml

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

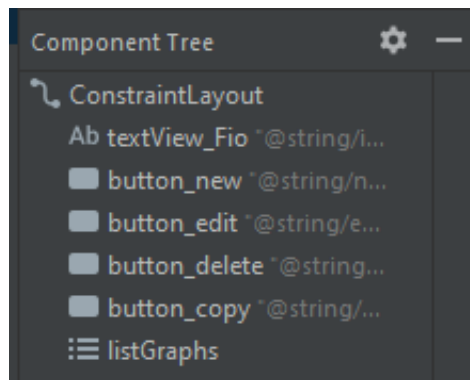


Рисунок 3 – Компоненты приложения activity\_main.xml

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

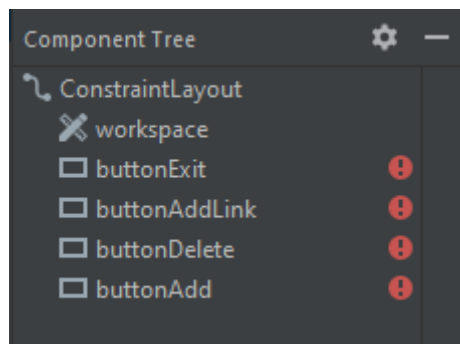


Рисунок 4 – Компоненты приложения activity\_second.xml

4. Создание строковых ресурсов в файле strings.xml, для подписей элементов интерфейса. Это изображено на рисунке 3.

```

1  <resources>
2      <string name="app_name">Графы</string>
3      <string name="add_graph">Создание графа</string>
4      <string name="edit_graph">Редактирование графа</string>
5      <string name="add_node">Добавление точки на граф</string>
6      <string name="add_link">Добавление связи</string>
7      <string name="new_text">Новый</string>
8      <string name="edit_text">Изменить</string>
9      <string name="delete_text">Удалить</string>
10     <string name="copy_text">Копировать</string>
11     <string name="info_FI0">Калмыков Семён 403</string>
12 </resources>

```

Рисунок 5 – Строковые ресурсы

5. В файле MainActivity.java написана логика интерфейса activity\_main.xml. Это изображено на рисунках 6 – 9.

```
1 package example.kalmykov403.graph;
2
3 import ...
4
21
22 public class MainActivity extends AppCompatActivity {
23     private ActivityMainBinding binding = null;
24
25     ArrayAdapter<GraphItem> graphsAdapter;
26     public static LocalDB database;
27
28     private boolean editEnable = false;
29     private boolean deleteEnable = false;
30     private boolean copyEnable = false;
31
32     @Override
33     protected void onCreate(Bundle savedInstanceState) {
34         super.onCreate(savedInstanceState);
35         binding = ActivityMainBinding.inflate(getLayoutInflater());
36         setContentView(binding.getRoot());
37
38         database = new LocalDB( context: this, name: "database2.db", factory: null, version: 1);
39         graphsAdapter = new ArrayAdapter<>( context: this, android.R.layout.simple_list_item_1);
40
41         binding.listGraphs.setAdapter(graphsAdapter);
42         binding.listGraphs.setOnItemClickListener((parent, view, position, id) -> {
43             if (!editEnable && !deleteEnable && !copyEnable) {
44                 Intent i = new Intent(getApplicationContext(), SecondActivity.class);
45                 i.putExtra( name: "GRAPH_ID", graphsAdapter.getItem(position).id);
46                 startActivity(i);
47             }
48             else if (editEnable) {
49                 openEditGraphDialog(graphsAdapter.getItem(position).id);
50                 editEnable = false;
51             }
52         });
53     }
54 }
```

Рисунок 6 – Код MainActivity.java

```

52     else if (copyEnable) {
53         database.addGraph(graphsAdapter.getItem(position).name);
54         refreshGraphs();
55         GraphItem newGraph = graphsAdapter.getItem(position: graphsAdapter.getCount() - 1);
56         ArrayList<NodeItem> oldNodes = database.getNodesInGraph(graphsAdapter.getItem(position).id);
57         for(NodeItem node : oldNodes) {
58             database.addNode(newGraph.id, node.name, node.x, node.y);
59         }
60         ArrayList<NodeItem> newNodes = database.getNodesInGraph(newGraph.id);
61         Map<Integer, NodeItem> nodeMap = new HashMap<>();
62         for(int i = 0; i < oldNodes.size(); i++) {
63             nodeMap.put(oldNodes.get(i).id, newNodes.get(i));
64         }
65         ArrayList<LinkItem> oldLinks = database.getLinks(graphsAdapter.getItem(position).id);
66         for(LinkItem link : oldLinks) {
67             database.addLink(newGraph.id, link.text, nodeMap.get(link.firstNodeId), nodeMap.get(link.secondNodeId));
68         }
69         copyEnable = false;
70     }
71     else {
72         database.deleteGraph(graphsAdapter.getItem(position).id);
73         deleteEnable = false;
74         refreshGraphs();
75     }
76 });
77
78 refreshGraphs();
79
80 binding.buttonNew.setOnClickListener(v -> {
81     openAddGraphDialog();
82 });
83
84 binding.buttonEdit.setOnClickListener(v -> {
85     deleteEnable = false;
86     editEnable = true;
87     copyEnable = false;
88 });
89
90 binding.buttonDelete.setOnClickListener(v -> {
91     deleteEnable = true;

```

Рисунок 7 – Код MainActivity.java

```

90         binding.buttonDelete.setOnClickListener(v -> {
91             deleteEnable = true;
92             editEnable = false;
93             copyEnable = false;
94         });
95
96         binding.buttonCopy.setOnClickListener(v -> {
97             deleteEnable = false;
98             editEnable = false;
99             copyEnable = true;
100         });
101     }
102
103     5 usages
104     private void refreshGraphs() {
105         graphsAdapter.clear();
106         graphsAdapter.addAll(database.getGraphs());
107
108         binding.buttonEdit.setEnabled(!graphsAdapter.isEmpty());
109         binding.buttonDelete.setEnabled(!graphsAdapter.isEmpty());
110         binding.buttonCopy.setEnabled(!graphsAdapter.isEmpty());
111     }
112
113     1 usage
114     private void openAddGraphDialog() {
115         AlertDialog dialog = new AlertDialog.Builder(context, this).create();
116
117         View dialogView = getLayoutInflater().inflate(R.layout.add_node_dialog, root, null);
118         AddNodeDialogBinding dialogBinding = AddNodeDialogBinding.bind(dialogView);
119
120         dialogBinding.tvTitle.setText("Создание графа");
121
122         dialogBinding.btnAdd.setOnClickListener(view -> {
123             if (!dialogBinding.etName.getText().toString().isEmpty()) {
124                 database.addGraph(dialogBinding.etName.getText().toString());
125                 refreshGraphs();
126                 dialog.dismiss();
127             }
128         });
129
130         dialog.setView(dialogView);
131         dialog.getWindow().setLayout(width: 300, height: 300);

```

Рисунок 8 – Код MainActivity.java

```

128     dialog.getWindow().setLayout( width: 300, height: 300);
129     dialog.show();
130 }
131
132 1 usage
133 private void openEditGraphDialog(Integer graphId) {
134     AlertDialog dialog = new AlertDialog.Builder( context: this).create();
135
136     View dialogView = getLayoutInflater().inflate(R.layout.add_node_dialog, root: null);
137     AddNodeDialogBinding dialogBinding = AddNodeDialogBinding.bind(dialogView);
138
139     dialogBinding.tvTitle.setText("Редактирование графа");
140
141     dialogBinding.btnAdd.setOnClickListener(view -> {
142         if (!dialogBinding.etName.getText().toString().isEmpty()) {
143             database.updateGraphName(graphId, dialogBinding.etName.getText().toString());
144             refreshGraphs();
145             dialog.dismiss();
146         }
147     });
148     dialog.setView(dialogView);
149     dialog.getWindow().setLayout( width: 300, height: 300);
150     dialog.show();
151 }

```

Рисунок 9 – Код MainActivity.java

6. В файле SecondActivity.java написана логика интерфейса activity\_second.xml. Это изображено на рисунках 10 – 13.

```

1 package example.kalmykov403.graph;
2
3 import ...
4
18
19 public class SecondActivity extends AppCompatActivity {
20     private ActivitySecondBinding binding = null;
21     private Integer graphId;
22     private boolean isConnection = false;
23     private boolean isDelete = false;
24     private NodeItem firstForConnect = null;
25
26     @Override
27     protected void onCreate(Bundle savedInstanceState) {
28         super.onCreate(savedInstanceState);
29         binding = ActivitySecondBinding.inflate(getLayoutInflater());
30         setContentView(binding.getRoot());
31         graphId = getIntent().getIntExtra("GRAPH_ID", 0);
32         binding.workspace.addListeners(
33             (newNode, newLinks) -> {
34                 database.updateNodePosition(newNode.id, newNode.x, newNode.y);
35                 for (LinkItem link: newLinks) {
36                     database.updateLink(link);
37                 }
38             },
39             node -> {
40                 if (isConnection) {
41                     if (firstForConnect == null) {
42                         firstForConnect = node;
43                     } else {
44                         if (!binding.workspace.isLinkExists(firstForConnect, node)) {
45                             openAddLinkDialog(firstForConnect, node);
46                         }
47                         isConnection = false;
48                         firstForConnect = null;
49                     }
46
47
48
49

```

Рисунок 10 – Код SecondActivity.java



```

49     }
50 }
51
52     if (isDelete) {
53         database.deleteNode(node.id);
54         ArrayList<LinkItem> links = database.getLinks(graphId);
55         for (LinkItem link : links) {
56             if(link.firstNodeId == node.id || link.secondNodeId == node.id) {
57                 database.deleteLink(link.id);
58             }
59         }
60         refreshNodes();
61     }
62 });
63
64 refreshNodes();
65
66 binding.buttonExit.setOnClickListener( v -> {
67     Intent i = new Intent(getApplicationContext(), MainActivity.class);
68     startActivity(i);
69 });
70
71 binding.buttonAdd.setOnClickListener(v -> {
72     openAddNodeDialog();
73 });
74
75 binding.buttonDelete.setOnClickListener(v -> {
76     isDelete = true;
77     isConnection = false;
78 });
79
80 binding.buttonAddLink.setOnClickListener( v -> {
81     isConnection = true;
82     isDelete = false;
83 });
84 }

```

Рисунок 11 – Код SecondActivity.java

```

86      4 usages
87      private void refreshNodes() {
88          binding.workspace.refreshPoints(database.getNodesInGraph(graphId), database.getLinks(graphId));
89          binding.workspace.invalidate();
90      }
91
92      1 usage
93      private void openAddNodeDialog() {
94          AlertDialog dialog = new AlertDialog.Builder(context: this).create();
95
96          View dialogView = getLayoutInflater().inflate(R.layout.add_node_dialog, root: null);
97          AddNodeDialogBinding dialogBinding = AddNodeDialogBinding.bind(dialogView);
98
99          dialogBinding.tvTitle.setText("Добавление точки на граф");
100
101          dialogBinding.btnAdd.setOnClickListener(view -> {
102              if (!dialogBinding.etName.getText().toString().isEmpty()) {
103                  String name = dialogBinding.etName.getText().toString();
104                  database.addNode(graphId, name, x: 300f, y: 300f);
105                  refreshNodes();
106                  dialog.dismiss();
107              }
108          });
109          dialog.setView(dialogView);
110          dialog.show();
111      }
112
113      1 usage
114      private void openAddLinkDialog(NodeItem firstNode, NodeItem secondNode) {
115          AlertDialog dialog = new AlertDialog.Builder(context: this).create();
116
117          View dialogView = getLayoutInflater().inflate(R.layout.add_node_dialog, root: null);
118          AddNodeDialogBinding dialogBinding = AddNodeDialogBinding.bind(dialogView);
119
120          dialogBinding.tvTitle.setText("Добавление связи");
121
122          dialogBinding.btnAdd.setOnClickListener(view -> {
123              if (!dialogBinding.etName.getText().toString().isEmpty()) {

```

Рисунок 12 – Код SecondActivity.java

```

120          if (!dialogBinding.etName.getText().toString().isEmpty()) {
121              String name = dialogBinding.etName.getText().toString();
122              database.addLink(graphId, name, firstNode, secondNode);
123              refreshNodes();
124              dialog.dismiss();
125          }
126      });
127
128      dialog.setView(dialogView);
129      dialog.show();
130  }
131  }

```

Рисунок 13 – Код SecondActivity.java