

```
1 package com.dimedrol.newcalc.ui.mainui
2
3 import
36
37 @Composable
38 fun Calculator(paddingValues: PaddingValues) {
39     var input by remember { mutableStateOf<String>("") }
40     var result by remember { mutableStateOf<String>("") }
41     var isBigResult by remember { mutableStateOf<Boolean>{ false } }
42
43     val buttons = listOf<String>{
44         "(", ")", "C", "+", "-",
45         "9", "8", "7", "*", "/",
46         "6", "5", "4", "=",
47         "3", "2", "1", "0",
48         ".", "0.1", "0.01", "0.001"
49     }
50
51     LaunchedEffect(input) {
52         result = try {
53             calculation(input)
54         } catch (e: Exception) {
55             ""
56         }
57     }
58
59     Box(
60         modifier = Modifier
61             .fillMaxSize()
62             .padding(paddingValues)
63     ) {
64
65         Box(
66             modifier = Modifier
67                 .fillMaxWidth()
68         ) {
69             Column(
70                 modifier = Modifier
71                     .fillMaxWidth(),
72                 verticalArrangement = Arrangement.Top
73             ) {
74                 // ...
75             }
76         }
77     }
78 }
```

```
38 fun Calculator(paddingValues: PaddingValues) {
39     // ...
40     // ...
41     // ...
42     // ...
43     // ...
44     // ...
45     // ...
46     // ...
47     // ...
48     // ...
49     // ...
50     // ...
51     // ...
52     // ...
53     // ...
54     // ...
55     // ...
56     // ...
57     // ...
58     // ...
59     // ...
60     // ...
61     // ...
62     // ...
63     // ...
64     // ...
65     // ...
66     // ...
67     // ...
68     // ...
69     // ...
70     // ...
71     // ...
72     // ...
73     // ...
74     // ...
75     // ...
76     // ...
77     // ...
78     // ...
79     // ...
80     // ...
81     // ...
82     // ...
83     // ...
84     // ...
85     // ...
86     // ...
87     // ...
88     // ...
89     // ...
90     // ...
91     // ...
92     // ...
93     // ...
94     // ...
95     // ...
96     // ...
97     // ...
98     // ...
99     // ...
100     // ...
101     // ...
102     // ...
103     // ...
104     // ...
105     // ...
106     // ...
107     // ...
108     // ...
109     // ...
110     // ...
111     // ...
112     // ...
113     Card(
114         modifier = Modifier
115             .fillMaxWidth()
116             .align(Alignment.BottomCenter),
117         colors = CardDefaults.cardColors(containerColor = MaterialTheme.colorScheme.onPrimary.copy(alpha = 0.3f))
118     ) {
119         Box(
120             modifier = Modifier
121                 .fillMaxWidth()
122         ) {
123             Column(
124                 modifier = Modifier
125                     .fillMaxWidth()
126                 // ...
127             ) {
128                 // ...
129             }
130         }
131     }
132 }
```

```
fun Calculator(paddingValues: PaddingValues) {
    63
    118
    122
    128
    149
    150
    151
    152
    153
    154
    155
    156
    157
    158
    159
    160
    161
    162
    163
    164
    165
    166
    167
    168
    169
    170
    171
    172
    173
    174
    175
    176
    177
    178
    179
    180
    181
    182
    183
}

else -> {
    val operators = listOf("+", "-", "*", "/", "%")

    if (input.isNotEmpty() && input.last() in operators && buttonItem.single() in operators) {
        return@Button
    }

    if (isBigResult) {
        isBigResult = false
        input = result + buttonItem
    } else if (input.isEmpty() && (buttonItem == "+" || buttonItem == "-" || buttonItem == "*" || buttonItem == "/" || buttonItem == "%")) {
        return@Button
    } else input += buttonItem
}

},
modifier = Modifier
    .padding(8.dp)
    .aspectRatio(1f)
    .background(MaterialTheme.colorScheme.onPrimary)
) {
    Text(
        text = buttonItem,
        fontSize = 24.sp,
        color = MaterialTheme.colorScheme.onBackground
    )
}
}
```

```
fun Calculator(paddingValues: PaddingValues) {
    63
    118
    122
    128
    173
    174
    175
    176
    177
    178
    179
    180
    181
    182
    183
    184
    185
    186
    187
    188
    189
    190
    191
    192
    193
    194
    195
    196
    197
    198
    199
    200
    201
    202
}

Text(
    text = buttonItem,
    fontSize = 24.sp,
    color = MaterialTheme.colorScheme.onBackground
)
}
}

fun calculation(input: String): String {
    if (input.contains(Regex("[0-9]")) {
        return "Некорректно"
    }

    return try {
        val result = net.objecthunter.exp4j.ExpressionBuilder(input).build().evaluate()
        if (result.toDouble() == 0.0) {
            result.toString()
        } else result.toString()
    } catch (e: Exception) {
        ""
    }
}
```

