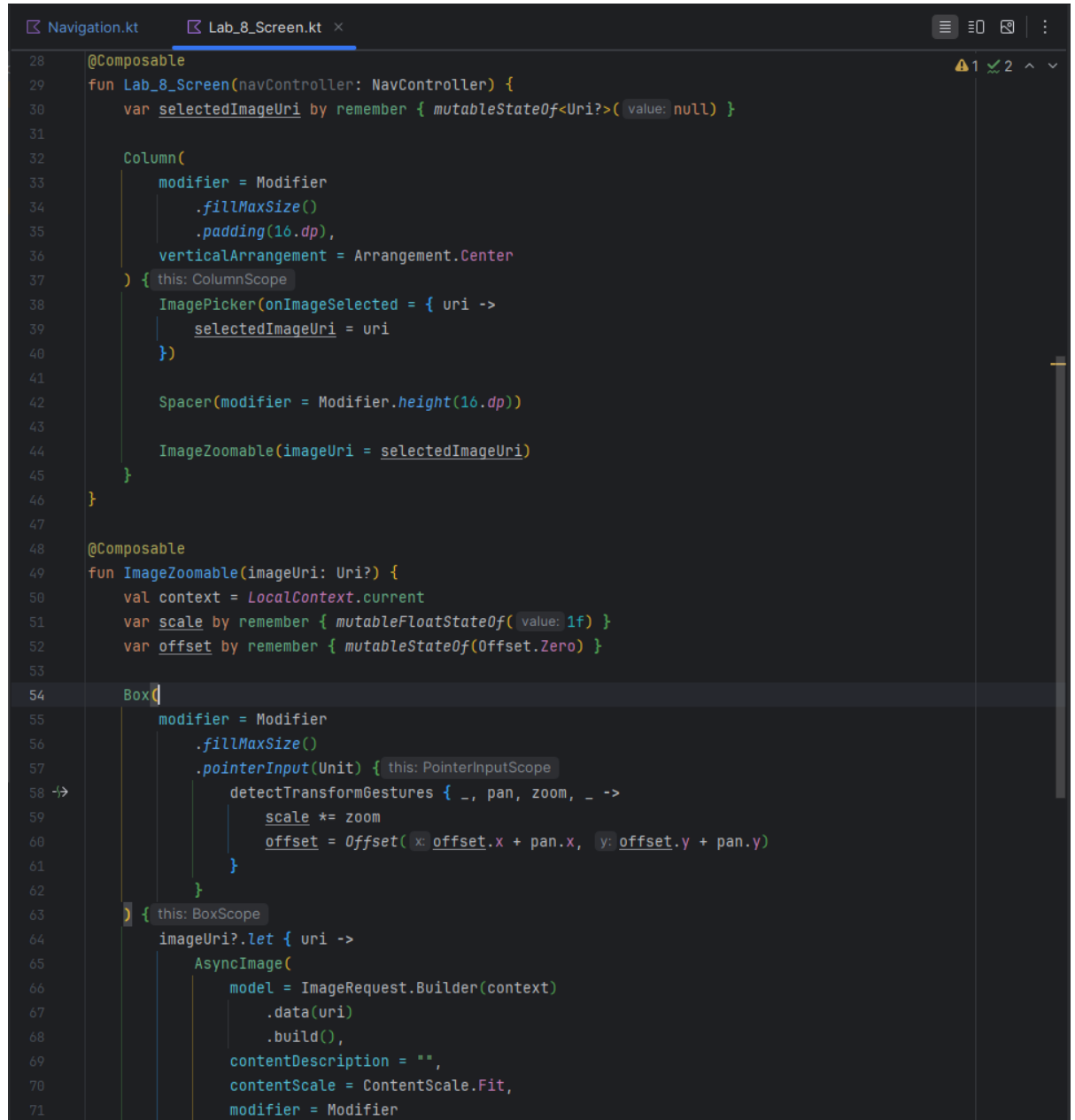


## Лабораторная работа №8

### РАБОТА С ИЗОБРАЖЕНИЯМИ

**Цель работы:** создать приложение с возможностью выбора картинки из библиотеки и её масштабирования.



```
Navigation.kt
Lab_8_Screen.kt x

28 @Composable
29 fun Lab_8_Screen(navController: NavController) {
30     var selectedImageUri by remember { mutableStateOf<Uri?>(value: null) }
31
32     Column(
33         modifier = Modifier
34             .fillMaxSize()
35             .padding(16.dp),
36         verticalArrangement = Arrangement.Center
37     ) { this: ColumnScope
38         ImagePicker(onImageSelected = { uri ->
39             selectedImageUri = uri
40         })
41
42         Spacer(modifier = Modifier.height(16.dp))
43
44         ImageZoomable(imageUri = selectedImageUri)
45     }
46 }
47
48 @Composable
49 fun ImageZoomable(imageUri: Uri?) {
50     val context = LocalContext.current
51     var scale by remember { mutableFloatStateOf(value: 1f) }
52     var offset by remember { mutableStateOf(Offset.Zero) }
53
54     Box(
55         modifier = Modifier
56             .fillMaxSize()
57             .pointerInput(Unit) { this: PointerInputScope
58                 detectTransformGestures { _, pan, zoom, _ ->
59                     scale *= zoom
60                     offset = Offset(x: offset.x + pan.x, y: offset.y + pan.y)
61                 }
62             }
63     ) { this: BoxScope
64         imageUri?.let { uri ->
65             AsyncImage(
66                 model = ImageRequest.Builder(context)
67                     .data(uri)
68                     .build(),
69                 contentDescription = "",
70                 contentScale = ContentScale.Fit,
71                 modifier = Modifier
```

Рис 8.1. Код лабораторной работы

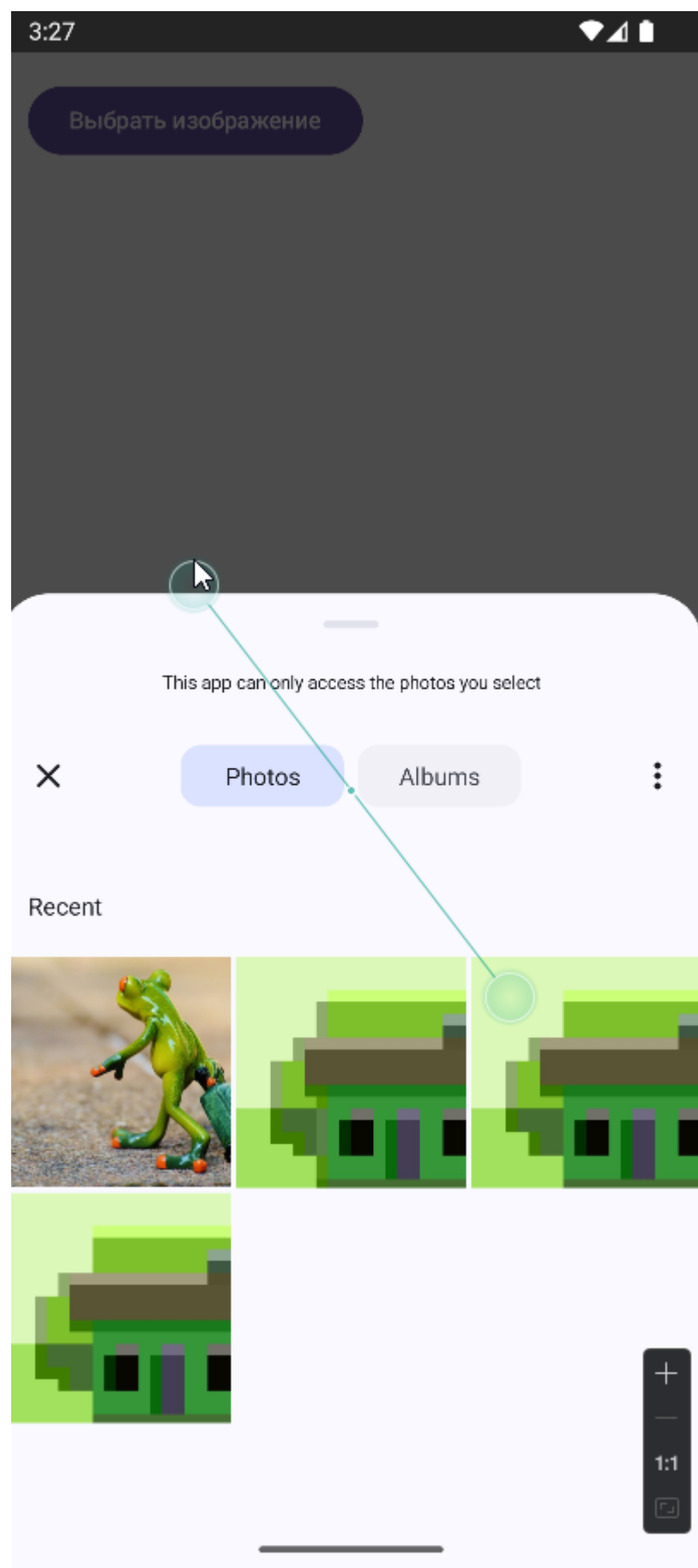
```

Navigation.kt  Lab_8_Screen.kt x
49 fun ImageZoomable(imageUri: Uri?) {
57     .pointerInput(Unit) { this: PointerInputScope
62     }
63 } { this: BoxScope
64     imageUri?.let { uri ->
65         AsyncImage(
66             model = ImageRequest.Builder(context)
67                 .data(uri)
68                 .build(),
69             contentDescription = "",
70             contentScale = ContentScale.Fit,
71             modifier = Modifier
72                 .graphicsLayer(
73                     scaleX = scale,
74                     scaleY = scale,
75                     translationX = offset.x,
76                     translationY = offset.y
77                 )
78             .fillMaxSize()
79         )
80     }
81 }
82 }
83
84 @Composable
85 fun ImagePicker(onImageSelected: (Uri?) -> Unit) {
86     val imagePickerLauncher = rememberLauncherForActivityResult(
87         contract = ActivityResultContracts.GetContent()
88     ) { uri: Uri? ->
89         onImageSelected(uri)
90     }
91
92     Button (
93         onClick = {
94             imagePickerLauncher.launch("image/*")
95         } { this: RowScope
96             Text( text: "Выбрать изображение")
97         }
98     )

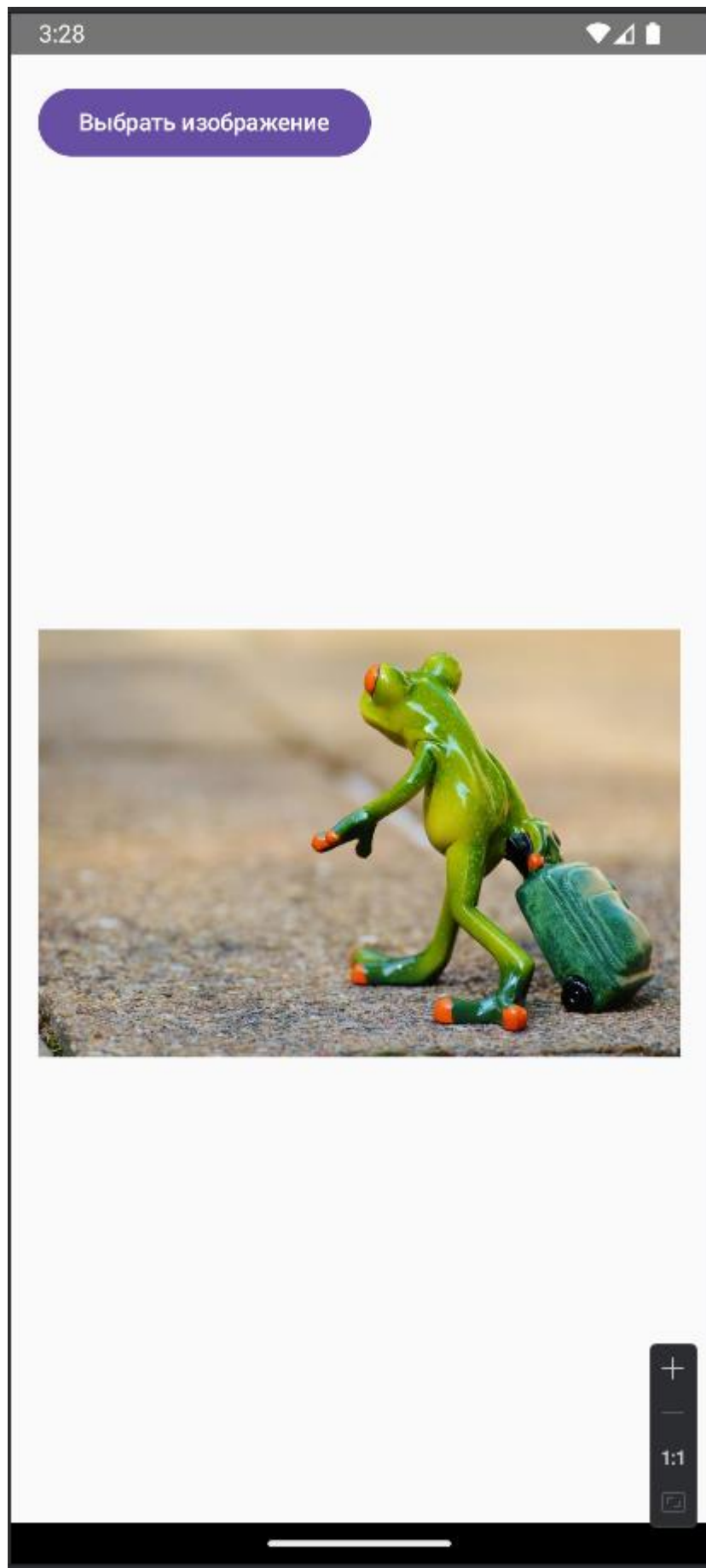
```

**Рис 8.2. Код лабораторной работы**

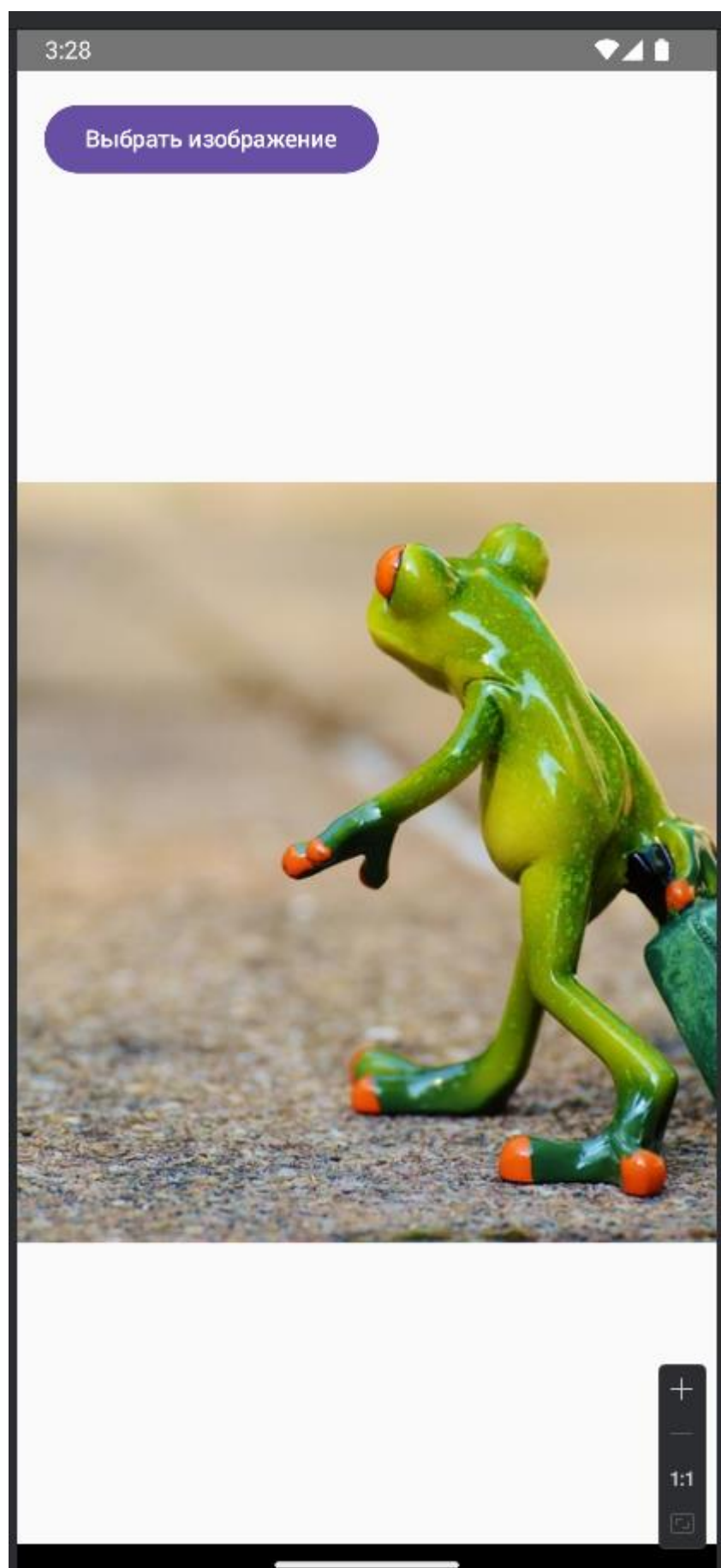
**Результаты работы программы:**



**Рис 8.3. Запуск**



**Рис 8.4. Отображение изображения**



**Рис 8.4. Увеличение изображения**

Выполнил	Баранько Д.А. 090303-ПИа-о22
Проверил	Елкин Н.С.