

ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ _____

ПРЕПОДАВАТЕЛЬ

старший преподаватель				С. Ю. Гуков
должность, уч. степень, звание		подпись, дата		инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2

Нативный пользовательский интерфейс

Вариант 5

по курсу: Разработка мобильных приложений. Разработка мобильных
приложений на Kotlin

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	4128			В. А. Воробьев
			подпись, дата	инициалы, фамилия

Санкт-Петербург 2024

СОДЕРЖАНИЕ

1	Постановка задачи	3
1.1	Цель работы	3
1.2	Задание	3
2	Выполнение работы	4
3	Вывод	11
	ПРИЛОЖЕНИЕ	12

1 Постановка задачи

1.1 Цель работы

Выполнить проектирование и разработку мобильного приложения под ОС Android на языке программирования высокого уровня Kotlin.

1.2 Задание

1. Создайте приложение для управления списком задач в соответствии с вариантом (предметной областью) – для определенного класса. Пользователь должен иметь возможность добавлять, удалять и отмечать задачи как выполненные. Используйте RecyclerView для отображения списка задач.
2. Разработайте галерею изображений (в соответствии с вариантом (предметной областью) – для определенного класса), где пользователь может просматривать изображения, а также добавлять новые изображения из галереи устройства. Используйте RecyclerView для отображения изображений в виде сетки.

Чтобы подобрать вариант необходимо найти число $N = (\text{номер группы} + \text{номер студента в журнале}) \bmod \text{количество_вариантов} + 1$. где mod – остаток от деления, N - номер варианта.

Вариант: геометрические фигуры.

2 Выполнение работы

Для выполнения работы был выбран фреймворк Jetpack Compose в виду простоты его использования.

Итоговая структура проекта изображена на рисунке 2.1

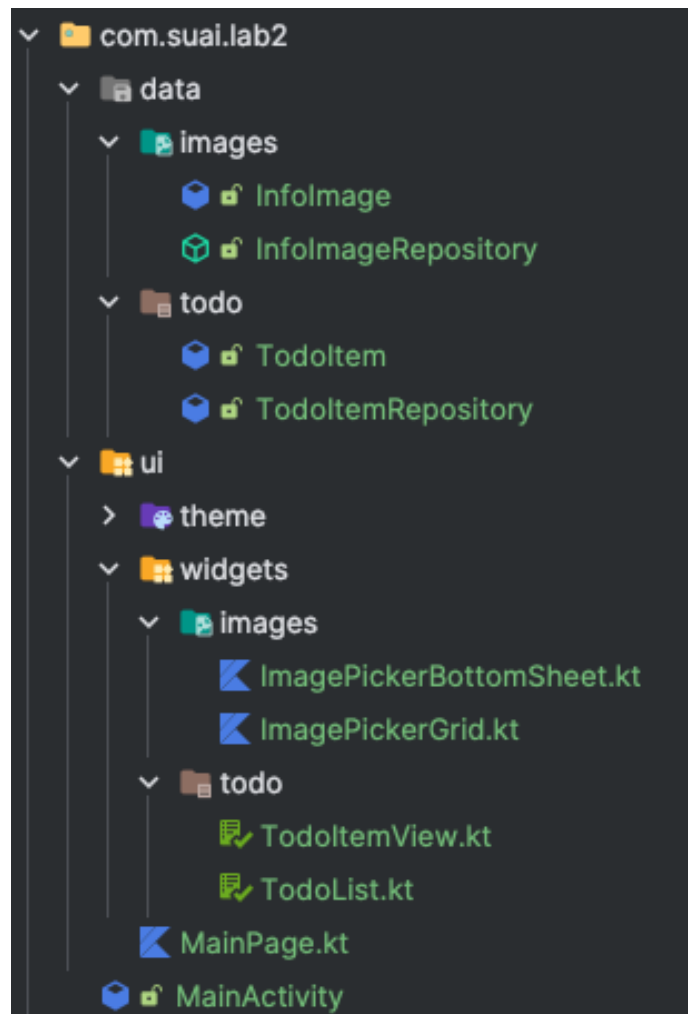


Рисунок 2.1 - Структура проекта

Код итоговый проект доступен в Приложении и на GitHub (URI - https://github.com/vladcto/suai-labs/tree/main/6_semester/ИТМ/lab2).

Демонстрация работы представлена на рисунках 2.2 - 2.7.

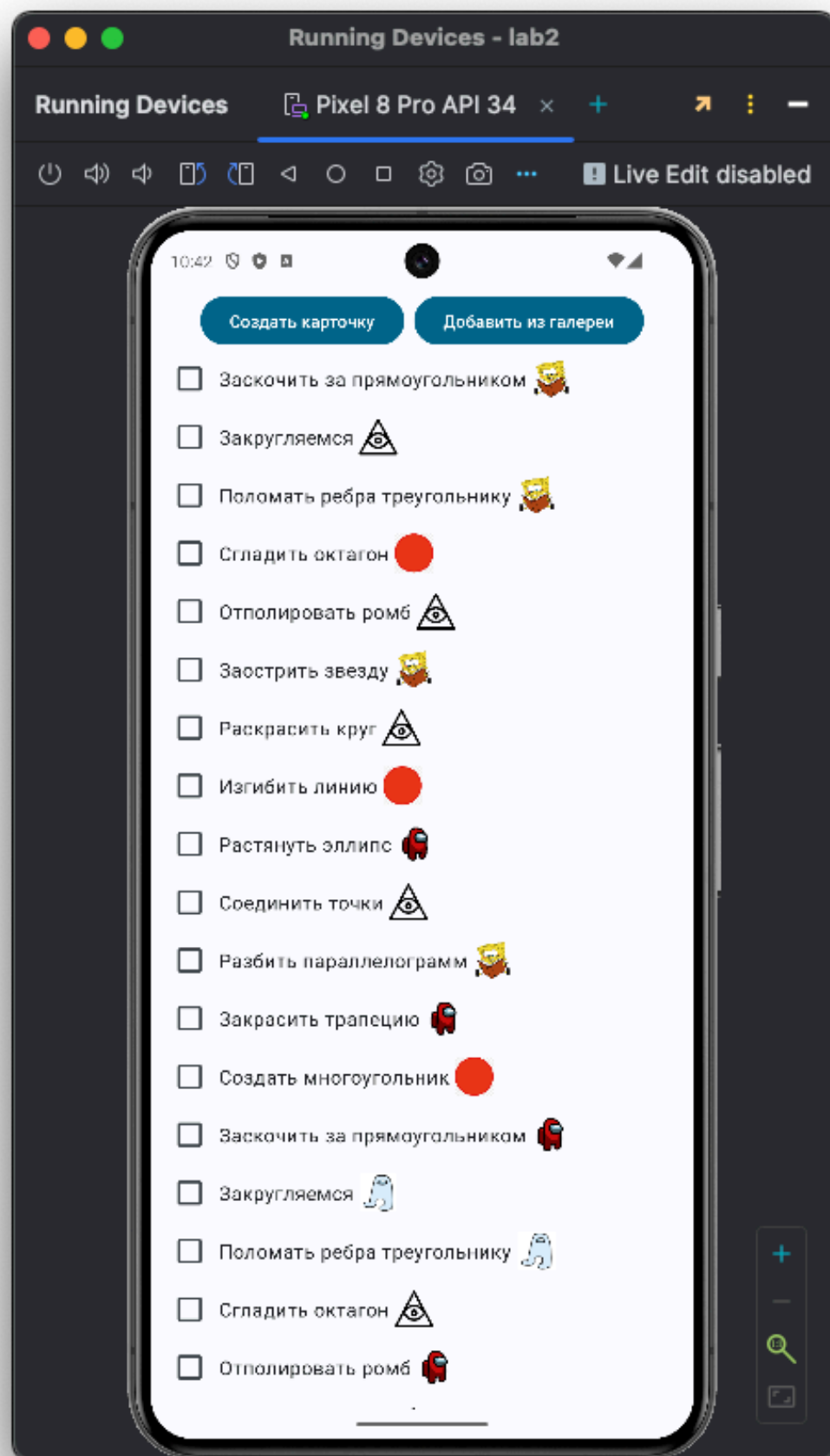


Рисунок 2.2 - Список ToDo

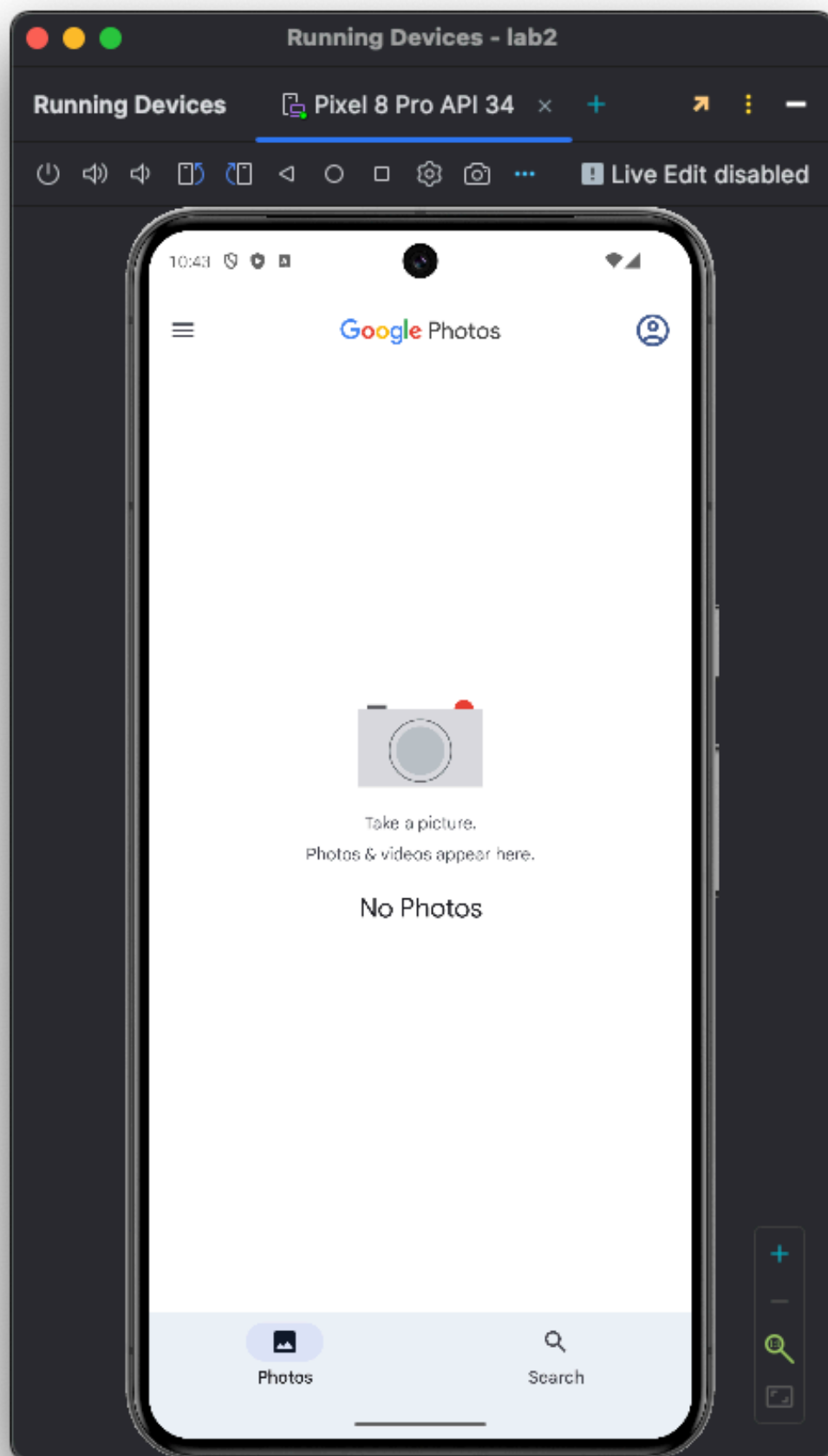


Рисунок 2.3 - Заход в галерею

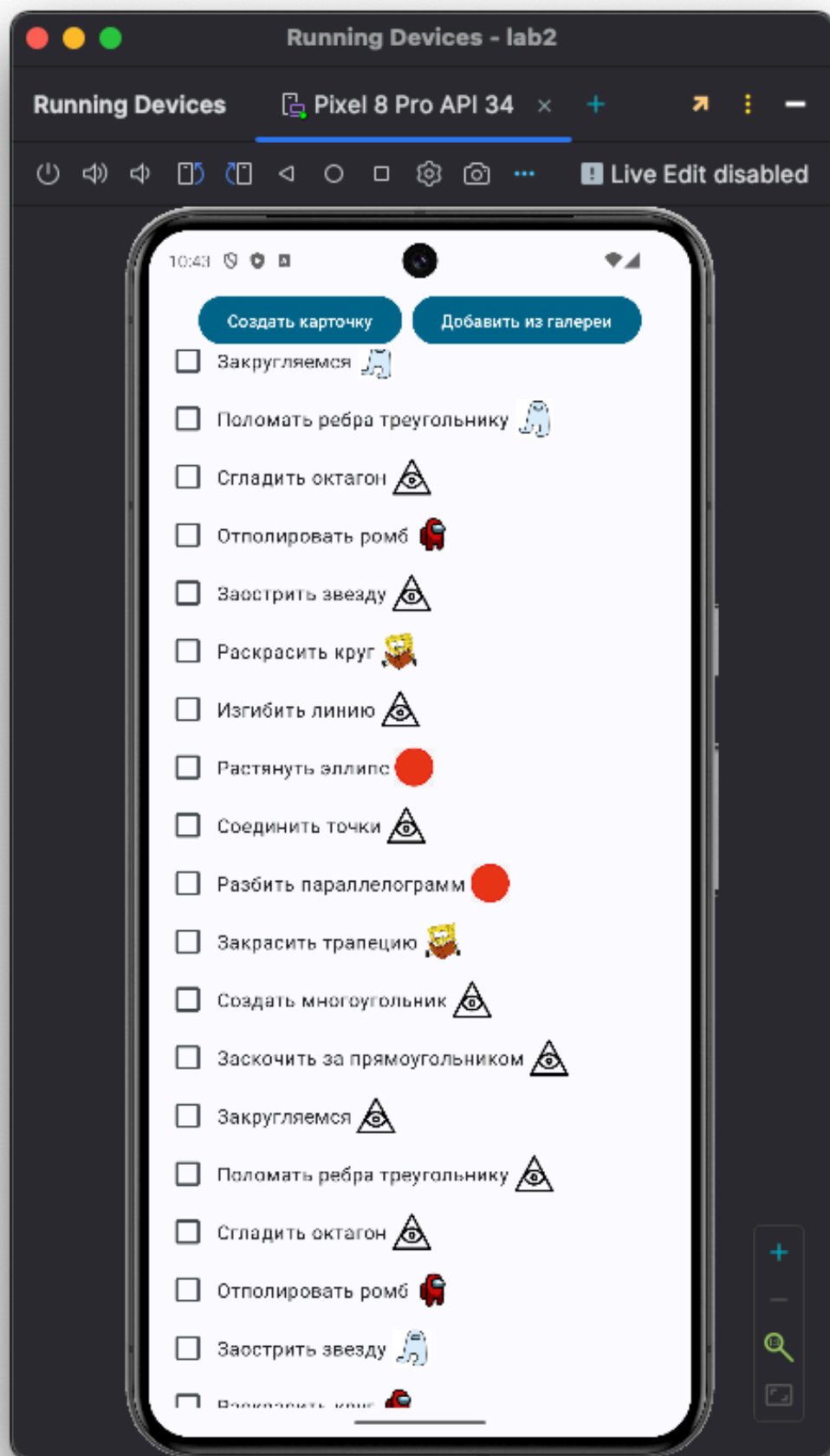


Рисунок 2.4 - Прокручивание списка

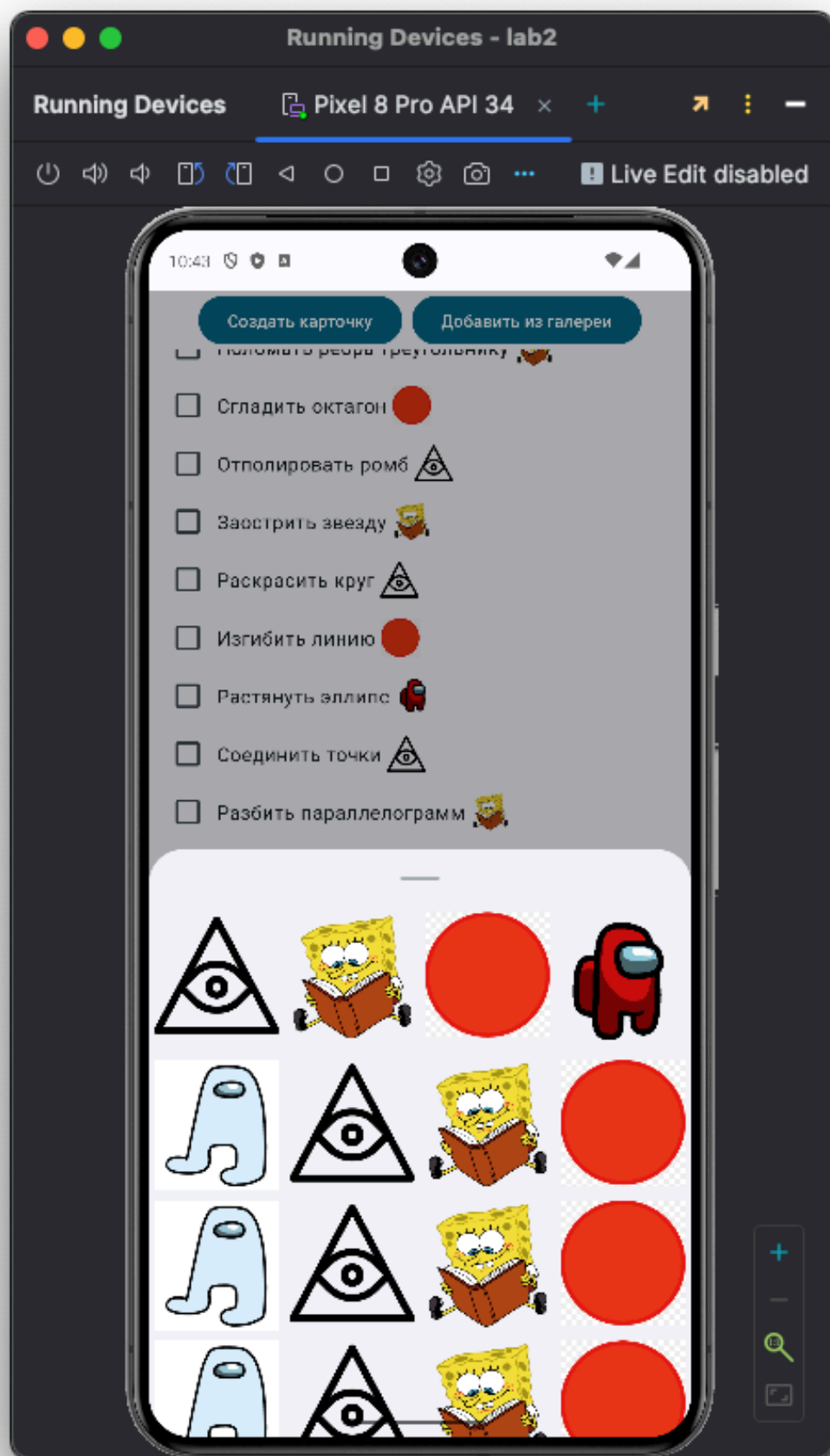


Рисунок 2.5 - BottomSheet изображений



Рисунок 2.6 - Прокрутка изображений

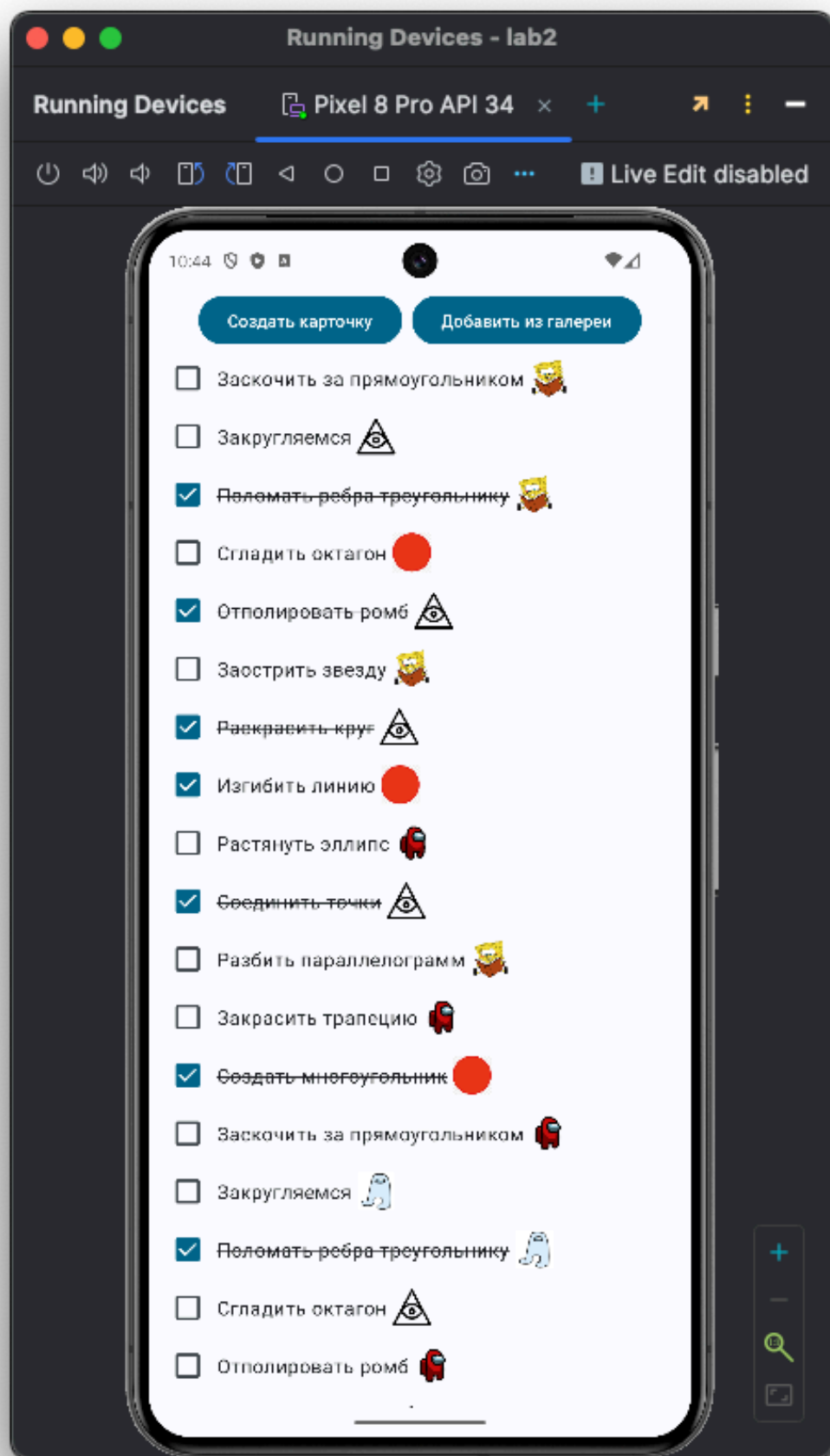


Рисунок 2.7 - Выполнение ToDo

3 Вывод

В результате выполнения лабораторной работы было успешно разработано мобильное приложение под ОС Android на языке программирования Kotlin с использованием фреймворка Jetpack Compose. Приложение включает в себя функционал управления списком задач и галерею изображений для геометрических фигур.

Выполненная лабораторная работа способствовала углублению знаний и навыков в области разработки мобильных приложений, а также позволила оценить преимущества и недостатки использования фреймворка Jetpack Compose для создания пользовательского интерфейса.

ПРИЛОЖЕНИЕ

```
1  MainPage.kt
2  package com.suai.lab2.ui
3
4  import android.content.Intent
5  import androidx.compose.foundation.layout.Box
6  import androidx.compose.foundation.layout.Column
7  import androidx.compose.foundation.layout.Row
8  import androidx.compose.foundation.layout.fillMaxWidth
9  import androidx.compose.foundation.layout.width
10 import androidx.compose.material3.Button
11 import androidx.compose.material3.ExperimentalMaterial3Api
12 import androidx.compose.material3.Text
13 import androidx.compose.material3.
    rememberModalBottomSheetState
14 import androidx.compose.runtime.Composable
15 import androidx.compose.runtime.getValue
16 import androidx.compose.runtime.mutableStateOf
17 import androidx.compose.runtime.remember
18 import androidx.compose.runtime.setValue
19 import androidx.compose.ui.Alignment
20 import androidx.compose.ui.Modifier
21 import androidx.compose.ui.platform.LocalContext
22 import androidx.compose.ui.unit.dp
23 import com.suai.lab2.ui.widgets.images.ImagePickerBottomSheet
24 import com.suai.lab2.ui.widgets.todo.TODOList
25
26 @OptIn(ExperimentalMaterial3Api::class)
27 @Composable
28 fun MainPage(modifier: Modifier = Modifier) {
29     val ctx = LocalContext.current
30     val sheetState = rememberModalBottomSheetState()
31     var showBottomSheet by remember { mutableStateOf(false) }
32
33     Column(
34         horizontalAlignment = Alignment.CenterHorizontally,
35         modifier = modifier.fillMaxWidth()
36     ) {
37         if (showBottomSheet) {
38             ImagePickerBottomSheet(
39                 onDismiss = { showBottomSheet = false },
```

```

40         sheetState = sheetState
41     )
42 }
43 // Just mock :3
44 Row {
45     Button(onClick = { showBottomSheet = !
46         showBottomSheet }) {
47         Text(text = "Создать карточку")
48     }
49     Box(
50         modifier = Modifier.width(8.dp)
51     )
52     Button(onClick = {
53         val intent = Intent(Intent.ACTION_VIEW)
54         intent.type = "image/*"
55         intent.flags = Intent.FLAG_ACTIVITY_NEW_TASK
56         ctx.startActivity(intent)
57     }) {
58         Text(text = "Добавить из галереи")
59     }
60     TodoList()
61 }
62 }
63 Color.kt
64 package com.suai.lab2.ui.theme
65
66 import androidx.compose.ui.graphics.Color
67
68 val Purple80 = Color(0xFFD0BCFF)
69 val PurpleGrey80 = Color(0xFFCCC2DC)
70 val Pink80 = Color(0xFFE6B8C8)
71
72 val Purple40 = Color(0xFF6650a4)
73 val PurpleGrey40 = Color(0xFF625b71)
74 val Pink40 = Color(0xFF7D5260)
75 Theme.kt
76 package com.suai.lab2.ui.theme
77
78 import android.app.Activity
79 import android.os.Build

```

```

80 import androidx.compose.foundation.isSystemInDarkTheme
81 import androidx.compose.material3.MaterialTheme
82 import androidx.compose.material3.darkColorScheme
83 import androidx.compose.material3.dynamicDarkColorScheme
84 import androidx.compose.material3.dynamicLightColorScheme
85 import androidx.compose.material3.lightColorScheme
86 import androidx.compose.runtime.Composable
87 import androidx.compose.ui.platform.LocalContext
88
89 private val DarkColorScheme = darkColorScheme(
90     primary = Purple80,
91     secondary = PurpleGrey80,
92     tertiary = Pink80
93 )
94
95 private val LightColorScheme = lightColorScheme(
96     primary = Purple40,
97     secondary = PurpleGrey40,
98     tertiary = Pink40
99
100     /* Other default colors to override
101     background = Color(0xFFFFFBFE),
102     surface = Color(0xFFFFFBFE),
103     onPrimary = Color.White,
104     onSecondary = Color.White,
105     onTertiary = Color.White,
106     onBackground = Color(0xFF1C1B1F),
107     onSurface = Color(0xFF1C1B1F),
108     */
109 )
110
111 @Composable
112 fun Lab2Theme(
113     darkTheme: Boolean = isSystemInDarkTheme(),
114     // Dynamic color is available on Android 12+
115     dynamicColor: Boolean = true,
116     content: @Composable () -> Unit
117 ) {
118     val colorScheme = when {
119         dynamicColor && Build.VERSION.SDK_INT >= Build.
            VERSION_CODES.S -> {

```

```

120         val context = LocalContext.current
121         if (darkTheme) dynamicDarkColorScheme(context)
122             else dynamicLightColorScheme(context)
123     }
124     darkTheme -> DarkColorScheme
125     else -> LightColorScheme
126 }
127
128 MaterialTheme(
129     colorScheme = colorScheme,
130     typography = Typography,
131     content = content
132 )
133 }
134 Type.kt
135 package com.suai.lab2.ui.theme
136
137 import androidx.compose.material3.Typography
138 import androidx.compose.ui.text.TextStyle
139 import androidx.compose.ui.text.font.FontFamily
140 import androidx.compose.ui.text.font.FontWeight
141 import androidx.compose.ui.unit.sp
142
143 // Set of Material typography styles to start with
144 val Typography = Typography(
145     bodyLarge = TextStyle(
146         fontFamily = FontFamily.Default,
147         fontWeight = FontWeight.Normal,
148         fontSize = 16.sp,
149         lineHeight = 24.sp,
150         letterSpacing = 0.5.sp
151     )
152     /* Other default text styles to override
153     titleLarge = TextStyle(
154         fontFamily = FontFamily.Default,
155         fontWeight = FontWeight.Normal,
156         fontSize = 22.sp,
157         lineHeight = 28.sp,
158         letterSpacing = 0.sp
159     ),

```

```

160         labelSmall = TextStyle(
161             fontFamily = FontFamily.Default,
162             fontWeight = FontWeight.Medium,
163             fontSize = 11.sp,
164             lineHeight = 16.sp,
165             letterSpacing = 0.5.sp
166         )
167     */
168 )
169 ImagePickerGrid.kt
170 package com.suai.lab2.ui.widgets.images
171
172 import android.icu.text.IDNA.Info
173 import androidx.compose.foundation.clickable
174 import androidx.compose.foundation.layout.Box
175 import androidx.compose.foundation.layout.fillMaxSize
176 import androidx.compose.foundation.layout.padding
177 import androidx.compose.foundation.lazy.grid.GridCells
178 import androidx.compose.foundation.lazy.grid.LazyVerticalGrid
179 import androidx.compose.foundation.lazy.grid.items
180 import androidx.compose.runtime.Composable
181 import androidx.compose.ui.Alignment
182 import androidx.compose.ui.Modifier
183 import androidx.compose.ui.layout.ContentScale
184 import androidx.compose.ui.tooling.preview.Preview
185 import androidx.compose.ui.unit.dp
186 import coil.compose.AsyncImage
187 import com.suai.lab2.data.images.InfoImage
188 import com.suai.lab2.data.images.InfoImageRepository
189
190 @Composable
191 fun ImagePickerGrid(onChoose: (InfoImage) -> Unit, modifier:
    Modifier = Modifier) {
192     LazyVerticalGrid(
193         columns = GridCells.Fixed(4),
194         modifier = modifier,
195     ) {
196         items(InfoImageRepository.images) { infoImage ->
197             ImageCard(infoImage = infoImage, onChoose =
                onChoose)
198         }

```



```

199     }
200 }
201
202 @Composable
203 fun ImageCard(infoImage: InfoImage, onChoose: (InfoImage) ->
    Unit, modifier: Modifier = Modifier) {
204     Box(
205         contentAlignment = Alignment.Center,
206         modifier = modifier
207             .padding(4.dp)
208             .fillMaxSize()
209             .clickable { onChoose(infoImage) }
210     ) {
211         AsyncImage(
212             model = infoImage.src,
213             contentDescription = "Image from network",
214             contentScale = ContentScale.Crop,
215             modifier = Modifier.fillMaxSize()
216         )
217     }
218 }
219
220 @Preview
221 @Composable
222 fun PreviewImagePickerGrid() {
223     ImagePickerGrid({
224     })
225 }
226 ImagePickerBottomSheet.kt
227 package com.suai.lab2.ui.widgets.images
228
229 import androidx.compose.material3.ExperimentalMaterial3Api
230 import androidx.compose.material3.ModalBottomSheet
231 import androidx.compose.material3.SheetState
232 import androidx.compose.runtime.Composable
233 import androidx.compose.ui.Modifier
234
235 @OptIn(ExperimentalMaterial3Api::class)
236 @Composable
237 fun ImagePickerBottomSheet(
238     onDismiss: () -> Unit,

```

```

239     modifier: Modifier = Modifier ,
240     sheetState: SheetState ,
241 ) {
242     ModalBottomSheet(
243         onDismissRequest = onDismiss ,
244         sheetState = sheetState ,
245         modifier = modifier ,
246     ) {
247         ImagePickerGrid(
248             onChoose = {}
249         )
250     }
251 }
252 TodoItemView.kt
253 package com.suai.lab2.ui.widgets.todo
254
255 import androidx.compose.foundation.clickable
256 import androidx.compose.foundation.layout.Box
257 import androidx.compose.foundation.layout.Row
258 import androidx.compose.foundation.layout.fillMaxWidth
259 import androidx.compose.foundation.layout.height
260 import androidx.compose.foundation.layout.width
261 import androidx.compose.material.icons.Icons
262 import androidx.compose.material.icons.rounded.ShoppingCart
263 import androidx.compose.material3.Checkbox
264 import androidx.compose.material3.Icon
265 import androidx.compose.material3.Text
266 import androidx.compose.runtime.Composable
267 import androidx.compose.ui.Alignment
268 import androidx.compose.ui.Modifier
269 import androidx.compose.ui.input.pointer.pointerInput
270 import androidx.compose.ui.text.style.TextDecoration
271 import androidx.compose.ui.tooling.preview.Preview
272 import androidx.compose.ui.unit.dp
273 import coil.compose.AsyncImage
274 import com.suai.lab2.data.images.InfoImage
275 import com.suai.lab2.data.todo.TODOItem
276
277 @Composable
278 fun TodoItemView(
279     onCheckChange: () -> Unit, item: TODOItem, modifier:

```

```

        Modifier = Modifier
280    ) {
281        Row(
282            modifier = modifier
283                .fillMaxWidth()
284                .clickable(
285                    onClick = onCheckChange,
286                ),
287            verticalAlignment = Alignment.CenterVertically,
288        ) {
289            Checkbox(
290                checked = item.closed,
291                onCheckedChange = {},
292                modifier = Modifier.pointerInput(Unit) {}
293            Text(
294                text = item.title,
295                textDecoration = if (item.closed) TextDecoration.
                    LineThrough else null
296            )
297            Box(modifier = Modifier.width(4.dp))
298            AsyncImage(
299                model = item.image.src,
300                contentDescription = "Item image",
301                modifier = Modifier
302                    .height(32.dp)
303                    .width(32.dp)
304            )
305        }
306    }
307
308    @Preview
309    @Composable
310    private fun TodoItemPreviewOpened() {
311        TodoItemView(
312            item = TodoItem(title = "Длинный title", closed =
                false, image = InfoImage("")),
313            onCheckChange = {},
314        )
315    }
316
317    @Preview

```

```

318 @Composable
319 private fun TodoItemPreviewClosed() {
320     TodoItemView(
321         item = TodoItem(title = "Длинный title", closed =
322             true, image = InfoImage("")),
323         onChange = {},
324     )
325 }
326
327 package com.suai.lab2.ui.widgets.todo
328
329 import androidx.compose.foundation.layout.Column
330 import androidx.compose.foundation.layout.padding
331 import androidx.compose.foundation.rememberScrollState
332 import androidx.compose.foundation.verticalScroll
333 import androidx.compose.runtime.Composable
334 import androidx.compose.runtime.mutableStateListOf
335 import androidx.compose.runtime.remember
336 import androidx.compose.ui.Modifier
337 import androidx.compose.ui.unit.dp
338 import com.suai.lab2.data.todo.TodoItem
339 import com.suai.lab2.data.todo.TodoItemRepository
340
341 @Composable
342 fun TodoList(modifier: Modifier = Modifier) {
343     val items = remember {
344         mutableStateListOf<TodoItem>().apply {
345             addAll(TodoItemRepository.items)
346         }
347     }
348
349     val verticalScroll = rememberScrollState()
350
351     Column(
352         modifier = modifier
353             .verticalScroll(verticalScroll)
354             .padding(horizontal = 8.dp)
355     ) {
356         items.onEachIndexed { i, item ->
357             TodoItemView(

```

```

358         item = item ,
359         onChangeChange = {
360             items[i] = items[i].copy(closed = !items[
                i ].closed)
361         },
362     )
363 }
364 }
365 }
366 MainActivity.kt
367 package com.suai.lab2
368
369 import android.os.Bundle
370 import androidx.activity.ComponentActivity
371 import androidx.activity.compose.setContent
372 import androidx.activity.enableEdgeToEdge
373 import androidx.compose.foundation.layout.fillMaxSize
374 import androidx.compose.foundation.layout.padding
375 import androidx.compose.material3.Scaffold
376 import androidx.compose.material3.Text
377 import androidx.compose.runtime.Composable
378 import androidx.compose.ui.Modifier
379 import androidx.compose.ui.tooling.preview.Preview
380 import com.suai.lab2.ui.MainPage
381 import com.suai.lab2.ui.theme.Lab2Theme
382
383 class MainActivity : ComponentActivity() {
384     override fun onCreate(savedInstanceState: Bundle?) {
385         super.onCreate(savedInstanceState)
386         enableEdgeToEdge()
387         setContent {
388             Lab2Theme {
389                 Scaffold(modifier = Modifier.fillMaxSize()) {
390                     innerPadding ->
391                     MainPage(
392                         modifier = Modifier.padding(
393                             innerPadding)
394                     )
395                 }
396             }
397         }
398     }
399 }

```

```

396     }
397 }
398 InfoImage.kt
399 package com.suai.lab2.data.images
400
401 data class InfoImage(val src: String)
402 InfoImageRepository.kt
403 package com.suai.lab2.data.images
404
405 abstract class InfoImageRepository {
406     fun fetch(): List<InfoImage> = images
407
408     companion object {
409         val images = listOf(
410             InfoImage(src = "https://cdn-icons-png.flaticon.
                com/512/1913/1913916.png"),
411             InfoImage(src = "https://avatanplus.com/files/
                resources/original/59c422d10dd3015ea62800a1.
                png"),
412             InfoImage(src = "https://e7.pngegg.com/pngimages
                /65/230/png-clipart-circle-red-circle-orange-
                wikimedia-commons-thumbnail.png"),
413             InfoImage(src = "https://preview.redd.it/
                an871k4o1sn51.png?width=440&format=png&auto=
                webp&s=85
                dcd6cb73b8760802e254ee14dfa3c7ab444591"),
414             InfoImage(src = "https://memozg.ru/img/posts/3833
                _639b61f0347b9.jpg"),
415             InfoImage(src = "https://cdn-icons-png.flaticon.
                com/512/1913/1913916.png"),
416             InfoImage(src = "https://avatanplus.com/files/
                resources/original/59c422d10dd3015ea62800a1.
                png"),
417             InfoImage(src = "https://e7.pngegg.com/pngimages
                /65/230/png-clipart-circle-red-circle-orange-
                wikimedia-commons-thumbnail.png"),
418             InfoImage(src = "https://memozg.ru/img/posts/3833
                _639b61f0347b9.jpg"),
419             InfoImage(src = "https://cdn-icons-png.flaticon.
                com/512/1913/1913916.png"),
420             InfoImage(src = "https://avatanplus.com/files/

```

```

resources/original/59c422d10dd3015ea62800a1.
png"),
421 InfoImage(src = "https://e7.pngegg.com/pngimages
/65/230/png-clipart-circle-red-circle-orange-
wikimedia-commons-thumbnail.png"),
422 InfoImage(src = "https://memozg.ru/img/posts/3833
_639b61f0347b9.jpg"),
423 InfoImage(src = "https://cdn-icons-png.flaticon.
com/512/1913/1913916.png"),
424 InfoImage(src = "https://avatanplus.com/files/
resources/original/59c422d10dd3015ea62800a1.
png"),
425 InfoImage(src = "https://e7.pngegg.com/pngimages
/65/230/png-clipart-circle-red-circle-orange-
wikimedia-commons-thumbnail.png"),
426 InfoImage(src = "https://memozg.ru/img/posts/3833
_639b61f0347b9.jpg"),
427 InfoImage(src = "https://cdn-icons-png.flaticon.
com/512/1913/1913916.png"),
428 InfoImage(src = "https://avatanplus.com/files/
resources/original/59c422d10dd3015ea62800a1.
png"),
429 InfoImage(src = "https://e7.pngegg.com/pngimages
/65/230/png-clipart-circle-red-circle-orange-
wikimedia-commons-thumbnail.png"),
430 InfoImage(src = "https://memozg.ru/img/posts/3833
_639b61f0347b9.jpg"),
431 InfoImage(src = "https://cdn-icons-png.flaticon.
com/512/1913/1913916.png"),
432 InfoImage(src = "https://avatanplus.com/files/
resources/original/59c422d10dd3015ea62800a1.
png"),
433 InfoImage(src = "https://e7.pngegg.com/pngimages
/65/230/png-clipart-circle-red-circle-orange-
wikimedia-commons-thumbnail.png"),
434 InfoImage(src = "https://memozg.ru/img/posts/3833
_639b61f0347b9.jpg"),
435 InfoImage(src = "https://cdn-icons-png.flaticon.
com/512/1913/1913916.png"),
436 InfoImage(src = "https://avatanplus.com/files/
resources/original/59c422d10dd3015ea62800a1.

```

```

        png"),
437     InfoImage(src = "https://e7.pngegg.com/pngimages
        /65/230/png-clipart-circle-red-circle-orange-
        wikimedia-commons-thumbnail.png"),
438     InfoImage(src = "https://memozg.ru/img/posts/3833
        _639b61f0347b9.jpg"),
439     InfoImage(src = "https://cdn-icons-png.flaticon.
        com/512/1913/1913916.png"),
440     InfoImage(src = "https://avatanplus.com/files/
        resources/original/59c422d10dd3015ea62800a1.
        png"),
441     InfoImage(src = "https://e7.pngegg.com/pngimages
        /65/230/png-clipart-circle-red-circle-orange-
        wikimedia-commons-thumbnail.png"),
442     InfoImage(src = "https://memozg.ru/img/posts/3833
        _639b61f0347b9.jpg"),
443     InfoImage(src = "https://cdn-icons-png.flaticon.
        com/512/1913/1913916.png"),
444     InfoImage(src = "https://avatanplus.com/files/
        resources/original/59c422d10dd3015ea62800a1.
        png"),
445     InfoImage(src = "https://e7.pngegg.com/pngimages
        /65/230/png-clipart-circle-red-circle-orange-
        wikimedia-commons-thumbnail.png"),
446     InfoImage(src = "https://memozg.ru/img/posts/3833
        _639b61f0347b9.jpg"),
447 )
448 }
449 }
450 TodoItem.kt
451 package com.suai.lab2.data.todo
452
453 import com.suai.lab2.data.images.InfoImage
454
455 data class TodoItem(
456     val title: String, val image: InfoImage,
457     val closed: Boolean
458 )
459 TodoItemRepository.kt
460 package com.suai.lab2.data.todo
461

```



```

462 import com.suai.lab2.data.images.InfoImageRepository
463
464 class TodoItemRepository {
465     companion object {
466         val items = listOf(
467             TodoItem("Заскочить за прямоугольником",
468                 InfoImageRepository.images[1], closed = false)
469             ,
470             TodoItem("Закругляемся", InfoImageRepository.
471                 images[0], closed = false),
472             TodoItem("Поломать ребра треугольнику",
473                 InfoImageRepository.images[1], closed = false)
474             ,
475             TodoItem("Сгладить октагон", InfoImageRepository.
476                 images[2], closed = true),
477             TodoItem("Отполировать ромб", InfoImageRepository
478                 .images[0], closed = false),
479             TodoItem("Заострить звезду", InfoImageRepository.
480                 images[1], closed = false),
481             TodoItem("Раскрасить круг", InfoImageRepository.
482                 images[0], closed = false),
483             TodoItem("Изгибнуть линию", InfoImageRepository.
484                 images[2], closed = true),
485             TodoItem("Растянуть эллипс", InfoImageRepository.
486                 images[3], closed = false),
487             TodoItem("Соединить точки", InfoImageRepository.
488                 images[0], closed = false),
489             TodoItem("Разбить параллелограмм",
490                 InfoImageRepository.images[1], closed = true),
491             TodoItem("Закрасить трапецию",
492                 InfoImageRepository.images[3], closed = false)
493             ,
494             TodoItem("Создать многоугольник",
495                 InfoImageRepository.images[2], closed = false)
496             ,
497             TodoItem("Заскочить за прямоугольником",
498                 InfoImageRepository.images[3], closed = false)
499             ,
500             TodoItem("Закругляемся", InfoImageRepository.
501                 images[4], closed = false),
502             TodoItem("Поломать ребра треугольнику",

```

```

        InfoImageRepository.images[4], closed = false)
    ,
483     TodoItem("Сгладить октагон", InfoImageRepository.
        images[0], closed = true),
484     TodoItem("Отполировать ромб", InfoImageRepository
        .images[3], closed = false),
485     TodoItem("Заострить звезду", InfoImageRepository.
        images[0], closed = false),
486     TodoItem("Раскрасить круг", InfoImageRepository.
        images[1], closed = false),
487     TodoItem("Изгибнуть линию", InfoImageRepository.
        images[0], closed = true),
488     TodoItem("Растянуть эллипс", InfoImageRepository.
        images[2], closed = false),
489     TodoItem("Соединить точки", InfoImageRepository.
        images[0], closed = false),
490     TodoItem("Разбить параллелограмм",
        InfoImageRepository.images[2], closed = true),
491     TodoItem("Закрасить трапецию",
        InfoImageRepository.images[1], closed = false)
    ,
492     TodoItem("Создать многоугольник",
        InfoImageRepository.images[0], closed = false)
    ,
493     TodoItem("Заскочить за прямоугольником",
        InfoImageRepository.images[0], closed = false)
    ,
494     TodoItem("Закругляемся", InfoImageRepository.
        images[0], closed = false),
495     TodoItem("Поломать ребра треугольнику",
        InfoImageRepository.images[0], closed = false)
    ,
496     TodoItem("Сгладить октагон", InfoImageRepository.
        images[0], closed = true),
497     TodoItem("Отполировать ромб", InfoImageRepository
        .images[3], closed = false),
498     TodoItem("Заострить звезду", InfoImageRepository.
        images[4], closed = false),
499     TodoItem("Раскрасить круг", InfoImageRepository.
        images[3], closed = false),
500     TodoItem("Изгибнуть линию", InfoImageRepository.

```

```

501         images[0], closed = true),
        TodoItem("Растянуть эллипс", InfoImageRepository.
            images[1], closed = false),
502         TodoItem("Соединить точки", InfoImageRepository.
            images[0], closed = false),
503         TodoItem("Разбить параллелограмм",
            InfoImageRepository.images[0], closed = true),
504         TodoItem("Закрасить трапецию",
            InfoImageRepository.images[4], closed = false)
        ,
505         TodoItem("Создать многоугольник",
            InfoImageRepository.images[0], closed = false)
        ,
506     )
507 }
508 }

```