МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное бюджетное образовательное учреждение высшего образования  
 «Кемеровский государственный университет»**

**Институт Цифры**

**ЛАБОРАТОРНАЯ РАБОТА №4**

**ПО ДИСЦИПЛИНЕ**

**“Разработка мобильных приложений”**

студент 3 курса

**Кислицын Тимофей Алексеевич**

Направление 02.03.02 – Фундаментальная информатика и информационные технологии

Преподаватель:

Киселев К.Е.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Работа защищена:

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025г.

с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_

Кемерово 2025

Цель работы

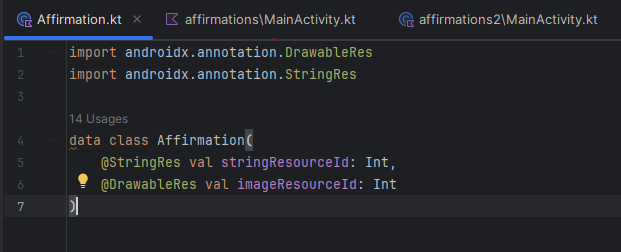
Изучить возможности создания прокручиваемых списков.

Задачи

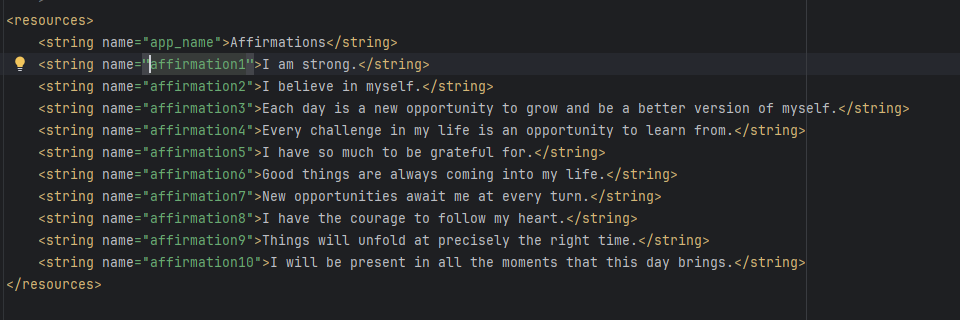
Создание прокручиваемых списков.

**Ход работы:**

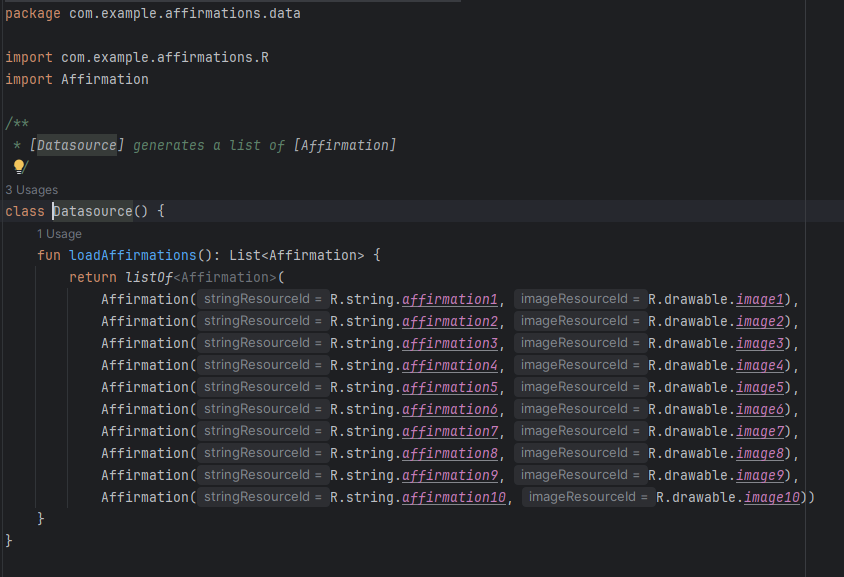
Data class



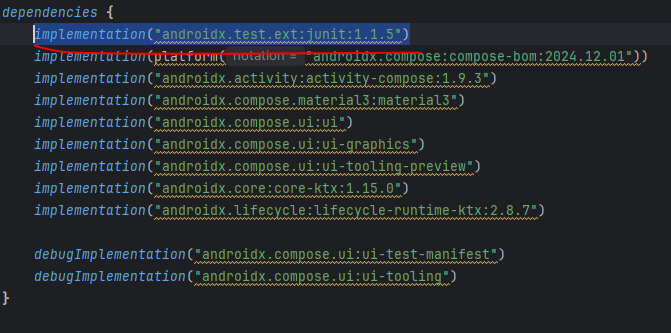
Strings.xml



Datasource



Добавил зависимость, без нее не запускалось



Код из MainActivity  
class MainActivity : ComponentActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContent {

AffirmationsTheme {

// A surface container using the 'background' color from the theme

Surface(

modifier = Modifier.fillMaxSize(),

color = MaterialTheme.colorScheme.background

) {

AffirmationsApp()

}

}

}

}

}

@Composable

fun AffirmationCard(affirmation: Affirmation, modifier: Modifier = Modifier) {

Card(modifier = modifier) {

Column {

Image(

painter = painterResource(affirmation.imageResourceId),

contentDescription = stringResource(affirmation.stringResourceId),

modifier = Modifier

.fillMaxWidth()

.height(194.dp),

contentScale = ContentScale.Crop

)

Text(

text = LocalContext.current.getString(affirmation.stringResourceId),

modifier = Modifier.padding(16.dp),

style = MaterialTheme.typography.headlineSmall

)

}

}

}

@Preview

@Composable

private fun AffirmationCardPreview() {

AffirmationCard(Affirmation(R.string.affirmation1, R.drawable.image1))

}

@Composable

fun AffirmationsList(affirmationList: List<Affirmation>, modifier: Modifier = Modifier) {

LazyColumn(modifier = modifier) {

items(affirmationList) { affirmation ->

AffirmationCard(

affirmation = affirmation,

modifier = Modifier.padding(8.dp)

)

}

}

}

@Composable

fun AffirmationsApp() {

val layoutDirection = LocalLayoutDirection.current

Surface(

modifier = Modifier

.fillMaxSize()

.statusBarsPadding()

.padding(

start = WindowInsets.safeDrawing.asPaddingValues()

.calculateStartPadding(layoutDirection),

end = WindowInsets.safeDrawing.asPaddingValues()

.calculateEndPadding(layoutDirection),

),

)

{

AffirmationsList(

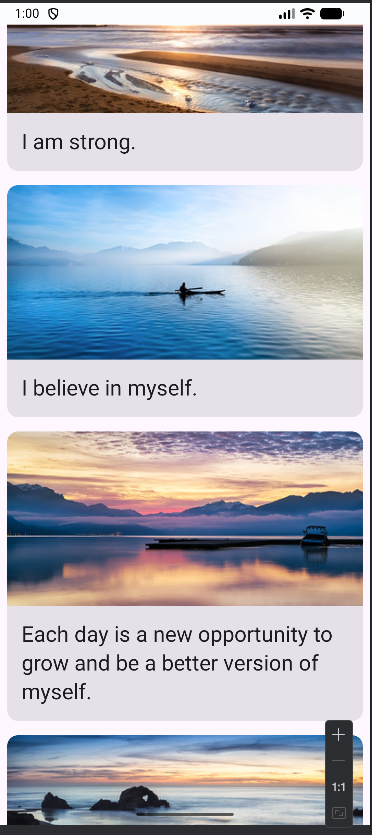
affirmationList = Datasource().loadAffirmations(),

)

}

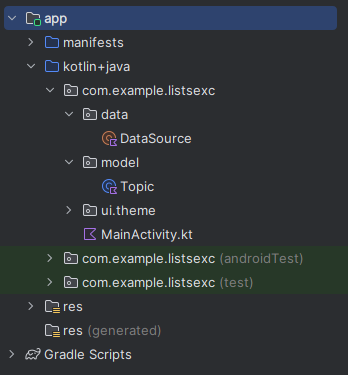
}

Всё получилось как в примере

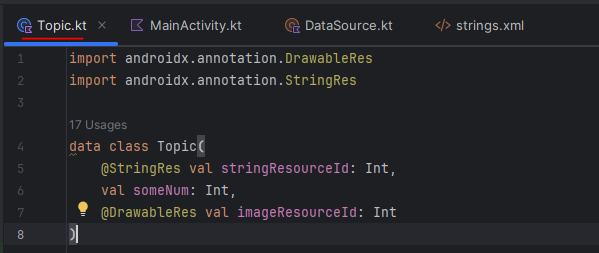


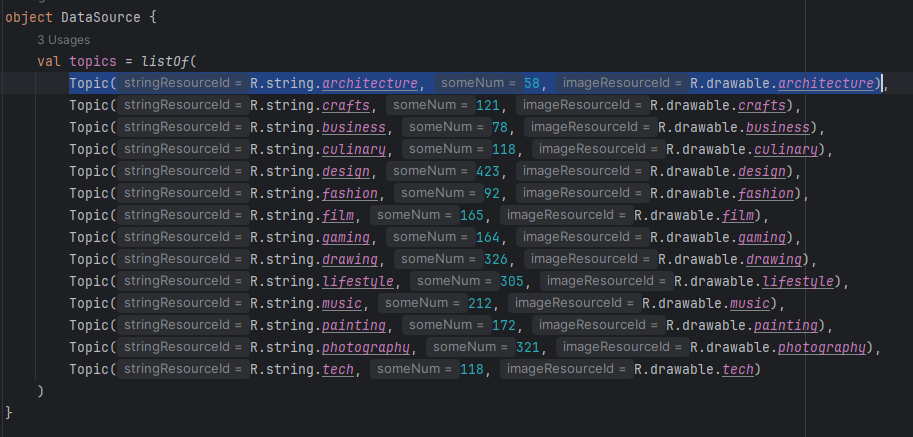
**Контрольное задание для самопроверки**

Схожая с предыдущим примером структура



Мой датакласс

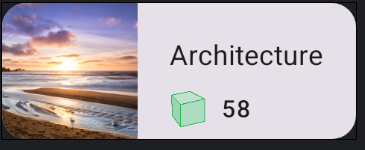


Datasource  


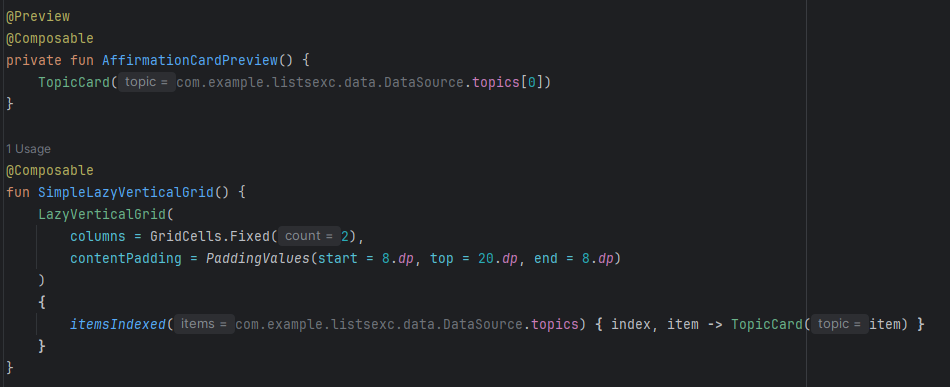
String.xml  

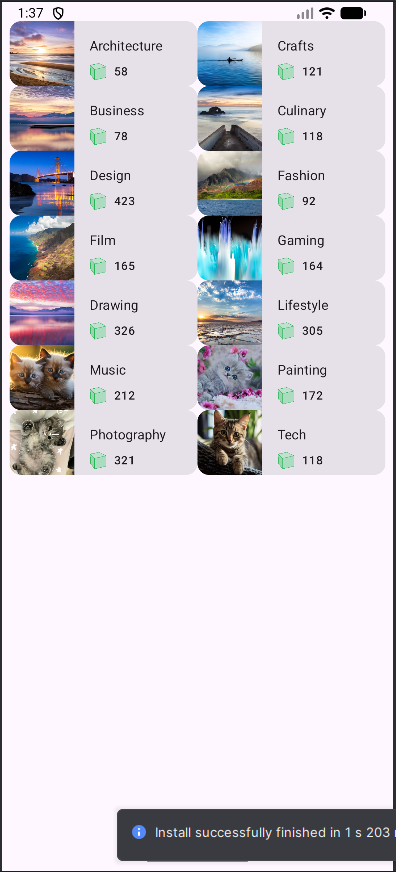

Основная функция, отвечающая за внешний вид карточки



Что получилось. Картинки взял из предыдущего примера, а симпатичный кубик нашел на каком-то стоке.  


Функции превью и размещение карточек в 2 колонки



**Итог:**

ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы изучил возможности создания прокручиваемых списков. Создал таковые самостоятельно.