Министерство науки и высшего образования Республики Казахстана

Некоммерческое акционерное общество

«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ имени Гумарбека Даукеева»

Институт автоматизации и информационных технологий

Кафедра IT-инженерии и искусственного интеллекта

**ОТЧЕТ**

**по лабораторной работе №5**

По дисциплине: Разработка программных приложений на базе Android

На тему: Intent

Образовательная программа: 6B06103 – Вычислительная техника и программное обеспечение

Выполнил: Кенес А. Ж. Группа ВТ(ПИ)у-23-4

Проверила: Старший преподаватель Ахсутова А.А.

\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_\_» \_\_\_\_\_\_\_\_2024 г.

Алматы 2024

**Содержание**

[Задание 1 3](#_Toc180939145)

[Задание 2 5](#_Toc180939146)

[Задание 3 8](#_Toc180939147)

[Задание 4 9](#_Toc180939148)

[Заключение 13](#_Toc180939149)

[Список литературы 14](#_Toc180939150)

# **Задание 1**

Надо сделать запуск activity через action из Intent-filter.

Листинг 1

@Composable  
fun FifthLabFirstTask(  
 context: Context  
) {  
 Row(  
 horizontalArrangement = Arrangement.Center,  
 verticalAlignment = Alignment.CenterVertically,  
 modifier = Modifier.fillMaxSize()  
 ) {  
 Button(onClick = {  
 val intent =  
 Intent("com.github.shortmikelon.univerandroidlabs.screens.lab5.showtime")  
 context.startActivity(intent)  
 }) {  
 Text("Показать время")  
 }  
  
 Button(  
 modifier = Modifier.padding(start = 10.dp),  
 onClick = {  
 val intent =  
 Intent("com.github.shortmikelon.univerandroidlabs.screens.lab5.showdate")  
 context.startActivity(intent)  
 }  
 ) {  
 Text("Показать дату")  
 }  
 }}

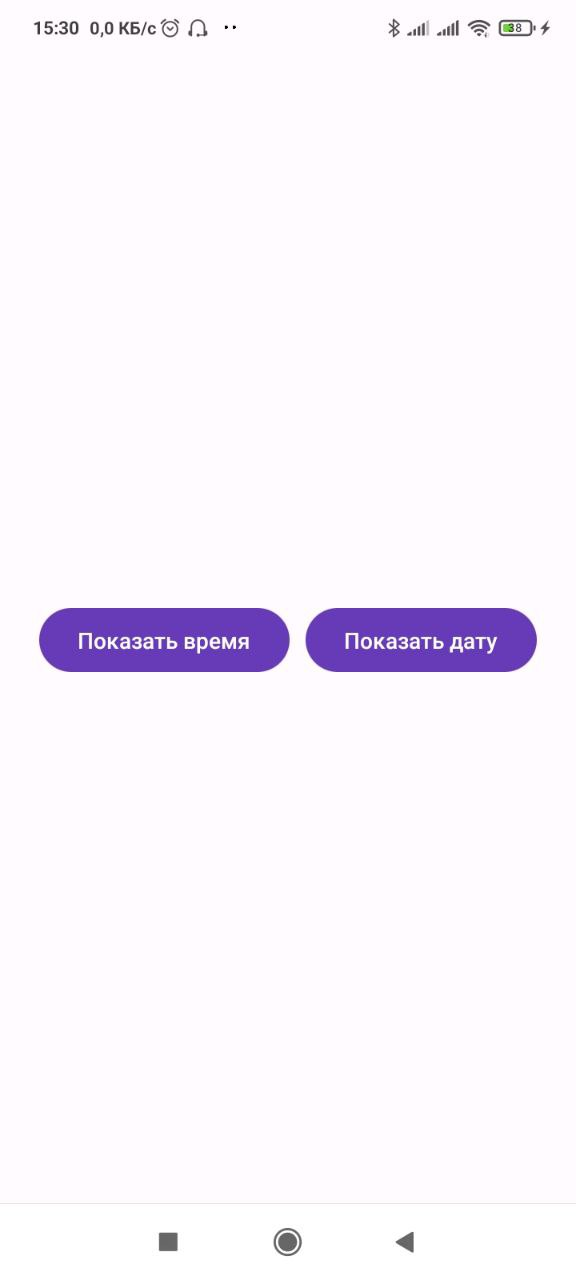


Рисунок 1

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 2

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 3

# **Задание 2**

Необходимо передать данные из первого activity ко второму через Intent.data

Листинг 2

@Composable  
fun NameInput(  
 context: Context  
) {  
 var firstName by remember { *mutableStateOf*("") }  
 var lastName by remember { *mutableStateOf*("") }  
  
 val onSubmitClicked = {  
 val intent = Intent(context, ShowNameActivity::class.*java*)  
  
 intent.putExtra("first\_name", firstName)  
 intent.putExtra("last\_name", lastName)  
  
 startActivity(  
 context,  
 intent,  
 null  
 )  
 }  
  
 Column(  
 horizontalAlignment = Alignment.CenterHorizontally,  
 verticalArrangement = Arrangement.Center,  
 modifier = Modifier.*fillMaxSize*()  
 ) {  
 Text(  
 text = "Введите имя и фамилия",  
 fontSize = 25.*sp* )  
  
 Row(  
 horizontalArrangement = Arrangement.SpaceEvenly,  
 modifier = Modifier.*fillMaxWidth*()  
 ) {  
  
 TextField(  
 value = firstName,  
 onValueChange = { firstName = it },  
 modifier = Modifier.*width*(230.*dp*)  
 )  
 }  
  
 Row(  
 horizontalArrangement = Arrangement.SpaceEvenly,  
 modifier = Modifier.*fillMaxWidth*()  
 ) {  
  
 TextField(  
 value = lastName,  
 onValueChange = { lastName = it },  
 modifier = Modifier.*width*(230.*dp*)  
 )  
 }  
  
 Button(onClick = onSubmitClicked) {  
 Text(text = "Отправить")  
 }  
 }  
}

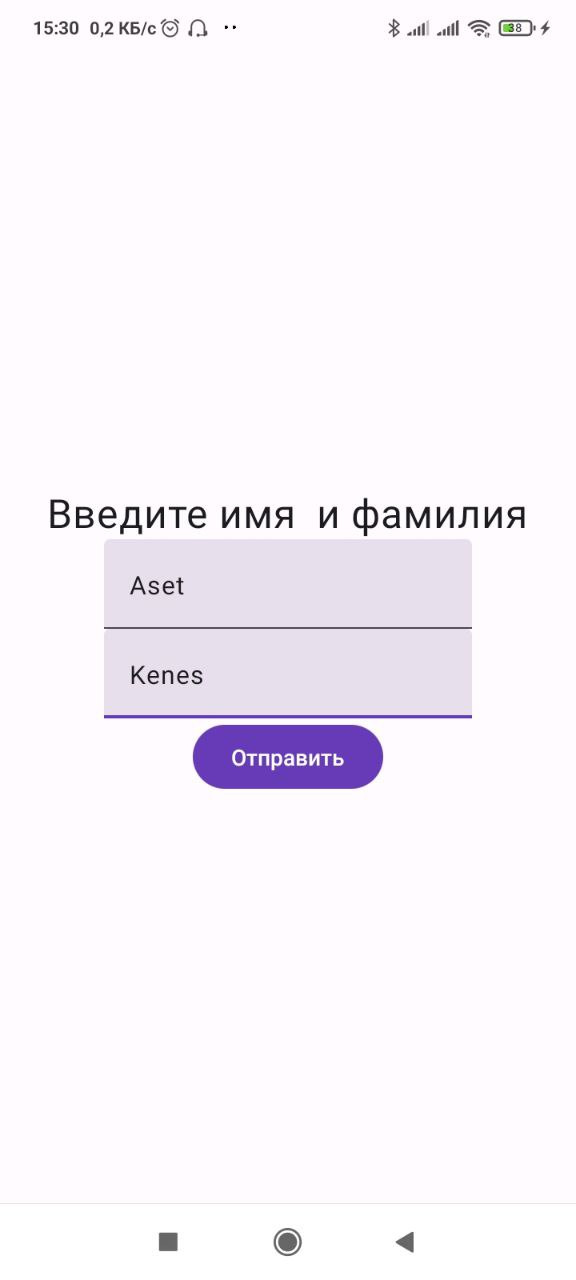


Рисунок 4

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 5

# **Задание 3**

Надо сделать возвращение результата из второго activity в первую

Листинг 3

@Composable  
fun InputActivityResult() {  
 var text by remember { *mutableStateOf*("") }  
  
 val inputNameActivityLauncher =  
 rememberLauncherForActivityResult(contract = InputNameActivityContract()) { result ->  
 text = result  
 }  
  
 Column(  
 horizontalAlignment = Alignment.CenterHorizontally,  
 verticalArrangement = Arrangement.Center,  
 modifier = Modifier.*fillMaxSize*()  
 ) {  
 Text(  
 text = text,  
 fontSize = 25.*sp*,  
 )  
  
 Button(onClick = { inputNameActivityLauncher.*launch*() }) {  
 Text("Запустить экран ввода")  
 }  
 }  
}

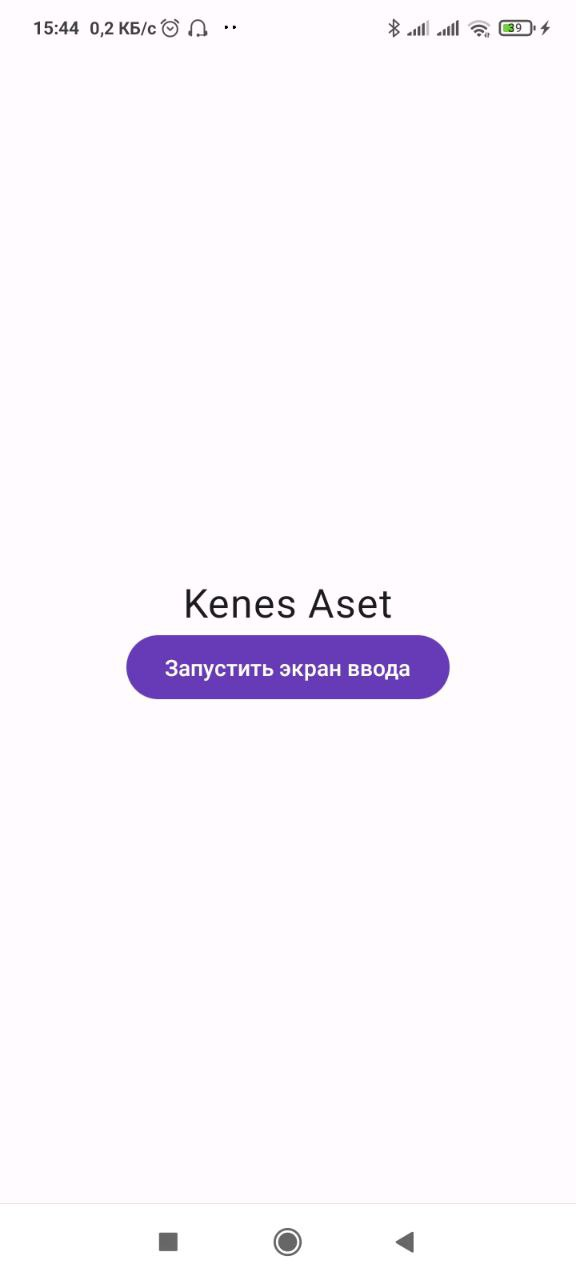


Рисунок 6

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 7

# **Задание 4**

Надо экран с определенный надписью, у которого можно менять расположение и цвет через соответствующие экраны.

Листинг 4

@Composable  
fun ChangeColorAndAlignment(  
 context: Context  
) {  
 var color by remember { *mutableStateOf*(Color.Black) }  
 var arrangement: Arrangement.Horizontal by remember { *mutableStateOf*(Arrangement.Center) }  
  
 val changeColorActivityLauncher =  
 rememberLauncherForActivityResult(contract = ActivityResultContracts.StartActivityForResult()) {  
 if (it.resultCode == Activity.*RESULT\_OK*) {  
 val i = it.data?.getIntExtra("Color", 1)  
 color = when (i) {  
 1 -> Color.Red  
 2 -> Color.Yellow  
 3 -> Color.Black  
 else -> Color.Red  
 }  
 }  
 }  
  
 val changeArrangementActivityLauncher =  
 rememberLauncherForActivityResult(contract = ActivityResultContracts.StartActivityForResult()) {  
 if (it.resultCode == Activity.*RESULT\_OK*) {  
 val i = it.data?.getIntExtra("Alignment", 2)  
 arrangement = when (i) {  
 1 -> Arrangement.Start  
 2 -> Arrangement.Center  
 3 -> Arrangement.End  
 else -> Arrangement.Center  
 }  
 }  
 }  
  
 Column(  
 modifier = Modifier.*fillMaxSize*(),  
 horizontalAlignment = Alignment.CenterHorizontally,  
 verticalArrangement = Arrangement.Center  
 ) {  
 Row(horizontalArrangement = arrangement, modifier = Modifier.*fillMaxWidth*()) {  
 Text(text = "Кенес Асет", color = color)  
 }  
  
 Row(modifier = Modifier.*fillMaxWidth*()) {  
 Button(onClick = {  
 changeArrangementActivityLauncher.launch(  
 Intent(  
 context,  
 ChangeAlignmentActivity::class.*java* )  
 )  
 }) {  
 Text("Сменить расположение")  
 }  
  
 Button(onClick = {  
 changeColorActivityLauncher.launch(  
 Intent(  
 context,  
 ChangeColorActivity::class.*java* )  
 )  
 }) {  
 Text("Сменить цвет")  
 }  
 }  
 }  
}



Рисунок 8

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 10

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 11

# **Заключение**

В этом проекте на Jetpack Compose были исследованы возможности создания анимаций и управления различными аспектами пользовательского интерфейса, такими как цвет, размер, видимость и трансформация элементов. Пример демонстрирует, как эффективно применять функции animate\*AsState для создания плавных анимаций и динамичных взаимодействий с пользователем. Использование различных анимационных спецификаций, таких как tween и spring, показало, как можно контролировать скорость и физику анимаций.

Основной вывод заключается в том, что Jetpack Compose предоставляет гибкий и мощный инструментарий для создания анимаций, которые легко интегрируются с состоянием приложения, делая интерфейсы более отзывчивыми и интерактивными.

# **Список литературы**

1. Динамически меняем параметры элементов экрана в андроид-приложении | Android Studio. URL: <https://www.fandroid.info/urok-18-dinamicheski-menyaem-parametry-elementov-ekrana-v-android-prilozhenii-android-studio/>

2. Делаем приложение: калькулятор для андроид | Android Studio. URL: <https://www.fandroid.info/urok-19-delaem-prilozhenie-kalkulyator-dlya-android-android-studio/>