

ЛАБОРАТОРНАЯ РАБОТА №6

ПЕРЕДАЧА УПРАВЛЕНИЯ МЕЖДУ СТРАНИЦАМИ

Цель работы: создать приложение из двух окон с возможностью передачи данных от одной страницы к другой в одну сторону.

```
import java.io.Serializable
import android.annotation.SuppressLint
import androidx.annotation.NonNull

abstract class Human(
    var firstName: String? = null,
    var lastName: String? = null,
    var age: String? = null
) : Serializable {

    constructor(name: String) : this(name, lastName: null, age: null)

    constructor(name: String, lastName: String) : this(name, lastName, age: null)

    override fun toString(): String {
        return "Human = {firstName: $firstName, lastName: $lastName, age: $age}"
    }
}

class Student(
    firstName: String? = null,
    lastName: String? = null,
    age: String? = null,
    var group: String? = null
) : Human(firstName, lastName, age), Serializable {

    constructor(name: String, group: String) : this(name, lastName: null, age: null, group)

    constructor(name: String, lastName: String, group: String) : this(name, lastName, age: null, group)

    @SuppressLint("NewApi")
    @NonNull
    override fun toString(): String {
        return "Human = {firstName: $firstName, lastName: $lastName, age: $age, group: $group}"
    }
}
```

Рисунок 1.1. – data.kt

```

composable(route = Screen.Lab_6_Screen_1.route) { this: AnimatedContentScope it: NavBackStackEntry
    Lab_6_Screen_1(navController = navController)
}

composable(
    route = Screen.Lab_6_Screen_2.route + "{name}/{lastName}/{group}/{age}",
    arguments = listOf(
        navArgument(name: "name") { this: NavArgumentBuilder
            type = NavType.StringType
        },
        navArgument(name: "lastName") { this: NavArgumentBuilder
            type = NavType.StringType
        },
        navArgument(name: "age") { this: NavArgumentBuilder
            type = NavType.StringType
        },
        navArgument(name: "group") { this: NavArgumentBuilder
            type = NavType.StringType
        }
    )
) { this: AnimatedContentScope it: NavBackStackEntry
    val name = it.arguments?.getString(key: "name")
    val lastName = it.arguments?.getString(key: "lastName")
    val age = it.arguments?.getString(key: "age")
    val group = it.arguments?.getString(key: "group")
    Lab_6_Screen_2(
        navController = navController,
        name = name ?: "Не указано",
        lastName = lastName ?: "Не указана",
        age = age ?: "Не указан",
        group = group ?: "Не указана",
    )
}

```

Рисунок 1.3. – Navigation.kt

```

fun Lab_6_Screen_1(navController: NavHostController) {
    ) { this: ColumnScope
        Text(
            text = "Имя"
        )
        TextField(
            value = nameState.value,
            onValueChange = { nameState.value = it},
        )
        Text(
            text = "Фамилия"
        )
        TextField(
            value = lastNameState.value,
            onValueChange = { lastNameState.value = it},
        )
        Text(
            text = "Группа"
        )
        TextField(
            value = groupState.value,
            onValueChange = { groupState.value = it},
        )
        Text(
            text = "Возраст"
        )
        TextField(
            value = ageState.value,
            onValueChange = { ageState.value = it},
        )
        Button(
            onClick = {
                val student = Student(nameState.value, lastNameState.value, ageState.value, groupState.value)

                navController.navigate( route: Screen.Lab_6_Screen_2.route + "/" + "${student.firstName}/${student.lastName}/${student.group}/${student.age}")
            },
            modifier = Modifier
                .padding(50.dp)
        )
    } this: RowScope

```

Рисунок 1.4. – экран заполнения Lab_6_Screen_1

```

@Composable
fun Lab_6_Screen_2(
    navController: NavHostController,
    name: String,
    lastName: String,
    age: String,
    group: String
) {

    val context = LocalContext.current

    Column(
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally,
        modifier = Modifier
            .fillMaxSize()
    ) { this: ColumnScope

        Text(
            text = "Имя: $name $lastName\nВозраст: $age\nГруппа: $group"
        )

        Button(
            onClick = {
                navController.popBackStack()
            },
            modifier = Modifier
                .padding(50.dp)
        ) { this: RowScope

            Text(
                text: "На предыдущий экран"
            )
        }
    }
}

```

Рисунок 1.5. – экран отображения Lab_6_Screen_2

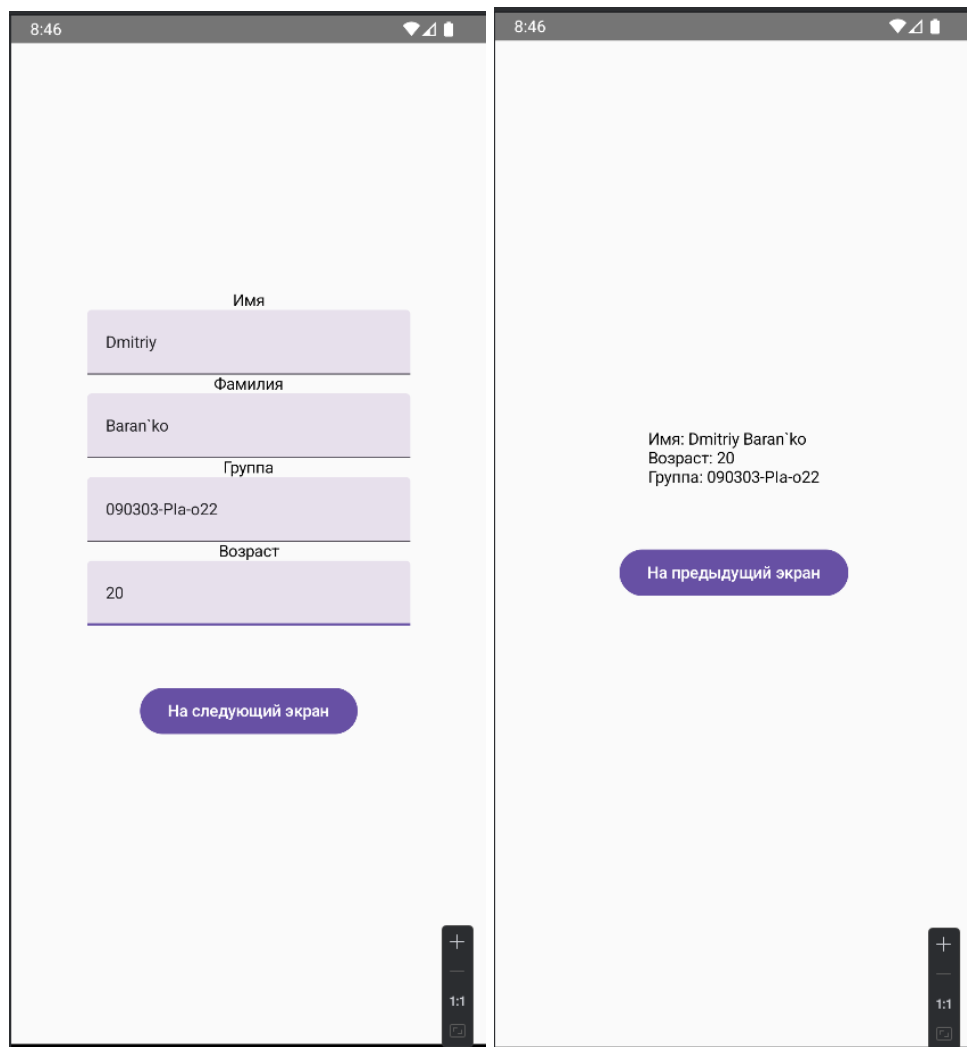


Рисунок 1.6. – Результат работы

Вывод: создал приложение из двух окон с возможностью передачи данных от одной страницы к другой в одну сторону.

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

