Министерство науки и образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №3

по курсу «Графические интерфейсы Android и Qt»

на тему «Жизненный цикл Activity в Android-приложениях. Intent»

Выполнили студент группы 19ВВ3:

Ланцов Андрей

Приняли:

д.т.н. Митрохин М.А.

**2022**

**Задание**

1. Переопределить методы жизненного цикла первой activity.
2. Добавить в переопределенные методы жизненного цикла код для вывода сообщений на экран или в системный журнал.
3. Создать intent для вызова второй activity.
4. Проанализировать порядок вызова методов жизненного цикла на основе выводимых сообщений при различных действиях пользователя.
5. Передать данные из одной activity в другую при помощи intent и продемонстрировать результат.

**Ход работы**

Листинг

## **AppCompatActivityWithLog.kt**

open class AppCompatActivityWithLog : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle ? ) {

super.onCreate(savedInstanceState)

Log.i("App Logger", "[${javaClass.name}]: Created")

}

override fun onPause() {

super.onPause()

Log.i("App Logger", "[${javaClass.name}]: Paused")

}

override fun onResume() {

super.onResume()

Log.i("App Logger", "[${javaClass.name}]: Resumed")

}

override fun onStart() {

super.onStart()

Log.i("App Logger", "[${javaClass.name}]: Started")

}

override fun onRestart() {

super.onRestart()

Log.i("App Logger", "[${javaClass.name}]: Restarted")

}

override fun onDestroy() {

super.onDestroy()

Log.i("App Logger", "[${javaClass.name}]: Destroyed")

}

override fun onStop() {

super.onStop()

Log.i("App Logger", "[${javaClass.name}]: Stopped")

}

}

Классы *MainActivity* и *ContentListActivity* теперь наследуются от ***AppCompatActivityWithLog*.**

**Добавили Intent**

class MainActivity : AppCompatActivityWithLog() {

override fun onCreate(savedInstanceState: Bundle ? ) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

Globals.contentListData.add("Sample")

}

fun btnLoginOnClick(view: View) {

val login = findViewById<EditText>(R.id.editTextTextEmailAddress)

val password = findViewById<EditText>(R.id.editTextTextPassword)

if (login.text.isEmpty() or password.text.isEmpty()) {

Toast.makeText(

baseContext,

"Enter login and password",

Toast.LENGTH\_SHORT

).show()

return

}

val intent = Intent(this, ContentListActivity::class.java).apply{

putExtra(Globals.EXTRA\_MESSAGE, login.text)

}

startActivity(intent)

}

}

class ContentListActivity : AppCompatActivityWithLog(), MultiChoiceModeListener{

//...

override fun onCreate(savedInstanceState: Bundle ? ) {

super.onCreate(savedInstanceState)

val arguments = intent.extras

if (arguments != null)

Log.i("App Logger", "Login: " + arguments.get(Globals.EXTRA\_MESSAGE).toString())

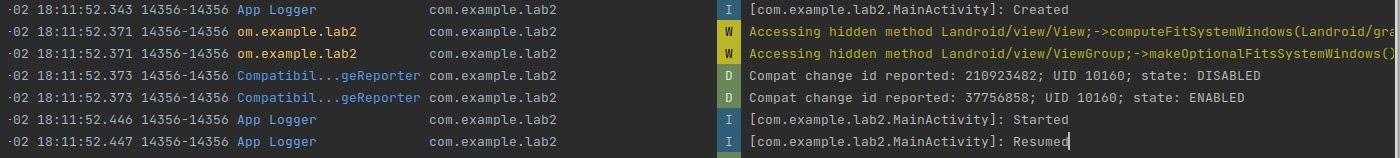
}

//...

}

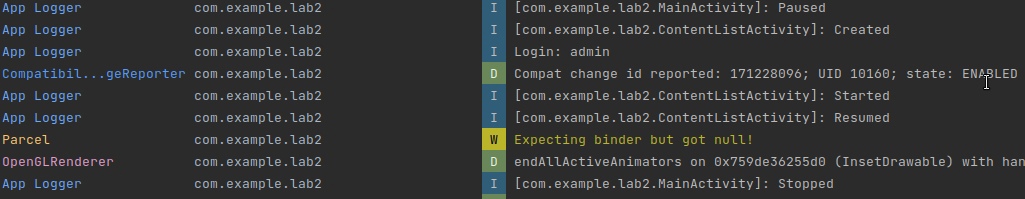
**Результат работы**

Запустили приложение.

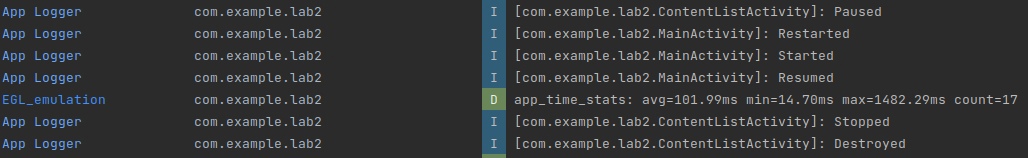


Авторизовались

Переданные данные: “Login: admin”



Перешли в предыдущее активити



Закрыли приложение



**Вывод**

Проанализировали порядок вызова методов жизненного цикла на основе выводимых сообщений при различных действиях пользователя.