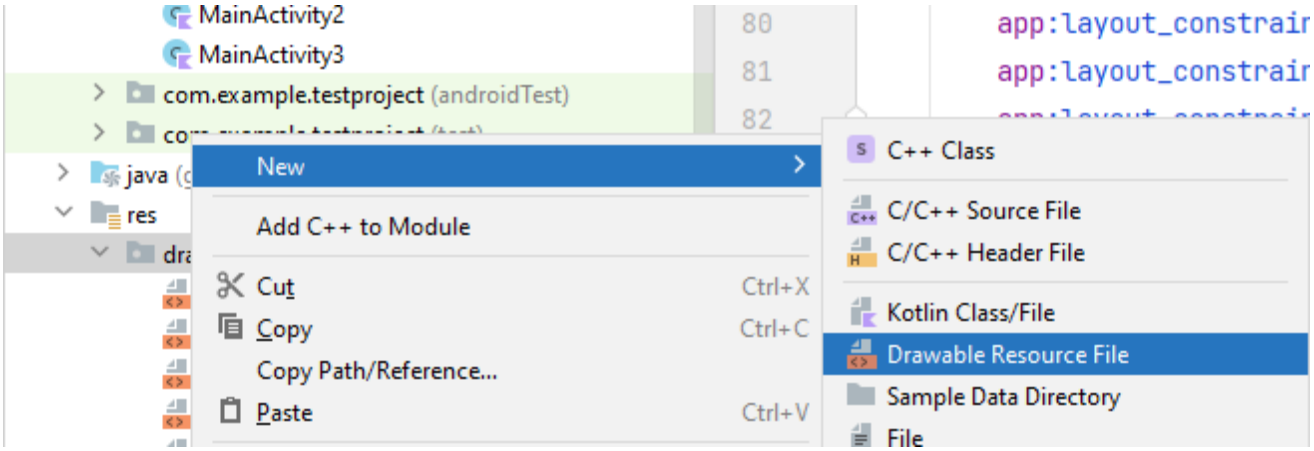
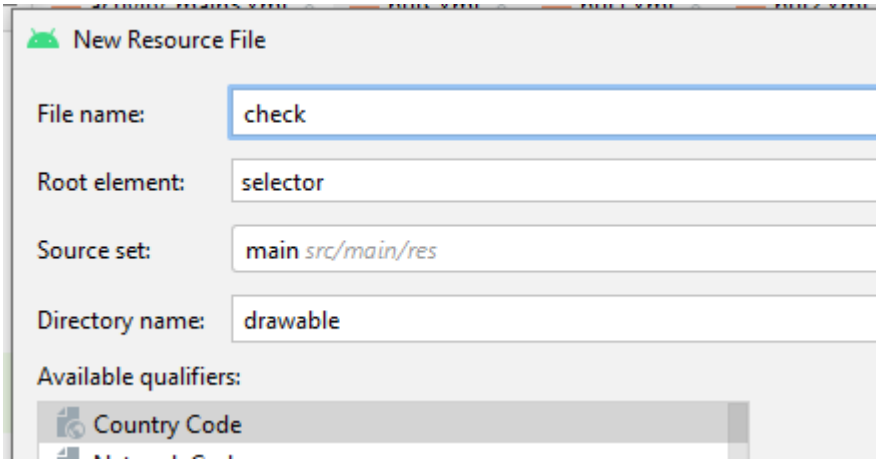
Практическая работа 7

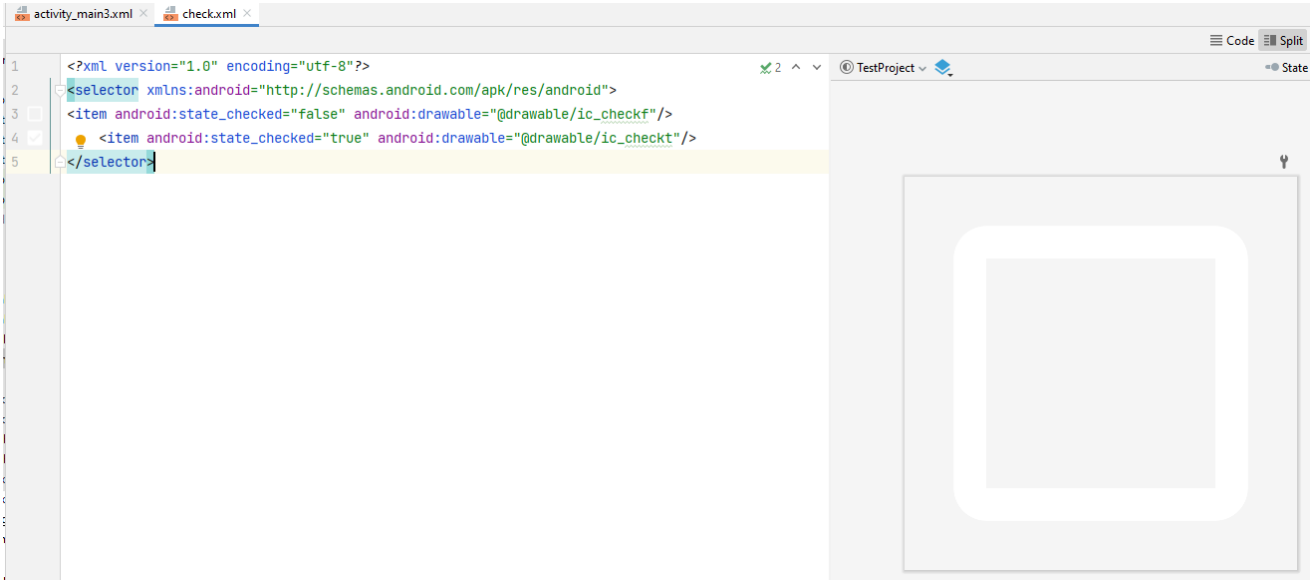
Сохранение почты и пароля пользователя

Продолжите работу над проектом Cinema, если он отсутствует - создайте новый. Сделайте активной активити входа и добавьте на экран элемент CheckBox:

```
82          app:layout_constraintTop_toBottomOf="@+id/name" />
83
84      <CheckBox
85          android:id="@+id/checkBox"
86          android:layout_width="wrap_content"
87          android:layout_height="wrap_content"
88          android:layout_marginStart="16dp"
89          android:layout_marginTop="48dp"
90          android:text="Запомнить данные"
91          android:textColor="@color/white"
92          android:button="@null"
93          android:drawableStart="@drawable/check"
94          android:drawablePadding="20dp"
95          app:layout_constraintStart_toStartOf="parent"
96          app:layout_constraintTop_toBottomOf="@+id/pass" />
97
98      <ImageView
99          android:id="@+id/imageView"
100         android:layout_width="wrap_content"
101         android:layout_height="wrap_content"
102         android:layout_marginTop="116dp"
103         app:layout_constraintEnd_toEndOf="parent"
104         app:layout_constraintStart_toStartOf="parent"
105         app:layout_constraintTop_toTopOf="parent"
106         android:src="@drawable/logo" />
107 </androidx.constraintlayout.widget.ConstraintLayout>
```





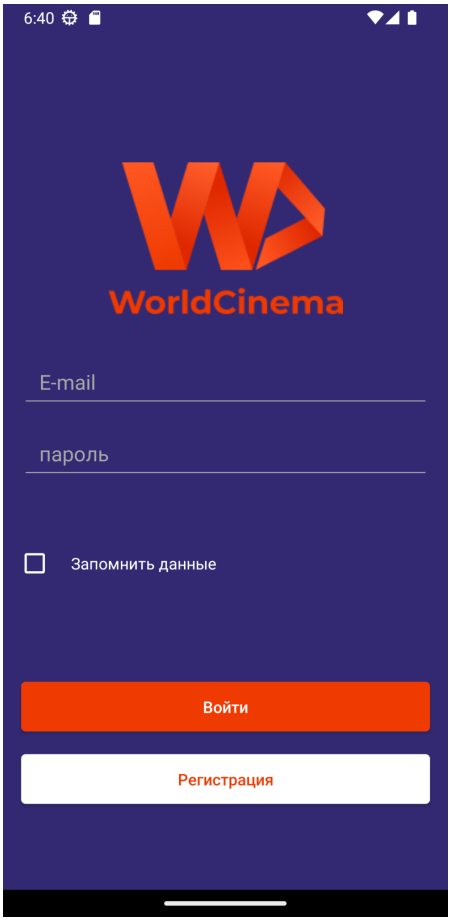


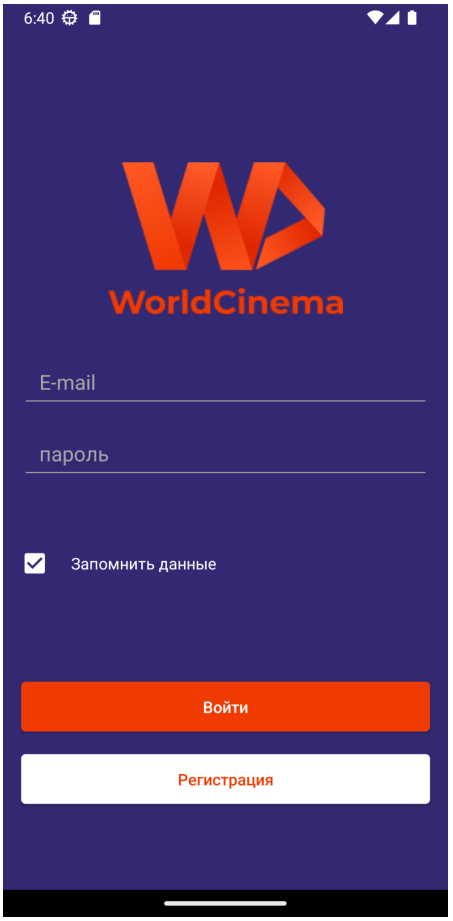
92 строка - отключение рамки чекбокса по умолчанию

93 строка - назначение пользовательских изображений для состояний чекбокс нажат и не нажат, подключение файла drawable selector

При создании **файла-селектора** (см.изображение выше) назначаются пользовательские иконки для двух состояний чекбокса: checked - true нажат и checked - false не нажат. Предварительно эти иконки должны быть размещены в папке drawable, это должны быть изображения в svg формате. Для добавления векторных изображений используется команда New Vector Asset (см.контрольную работу 2). Для добавления иконки-состояния чекбокса используйте встроенную библиотеку Android Studio

Запустите приложение:





Настройте запоминание введенных пользователем почты, пароля и состояния чекбокса (нажат или нет) при нажатии на кнопку **Войти**, но только в том случае, если будет установлена галочка “Запомнить данные”.

Откройте файл настройки логики экрана и дополните код следующими строками (внимательно прочитайте пояснения к строкам кода):

TABLEE.xml | MainActivity3.kt | check.xml

```
1  package com.example.testproject
2
3  import android.content.Intent
4  import android.content.SharedPreferences
5  import androidx.appcompat.app.AppCompatActivity
6  import android.os.Bundle
7  import android.view.View
8  import android.widget.CheckBox
9  import android.widget.EditText
10 import android.widget.TextView
11
12 class MainActivity3 : AppCompatActivity() {
13
14     var preff: SharedPreferences?=null
15     lateinit var email: EditText
16     lateinit var password:EditText
17     lateinit var check:CheckBox
18     override fun onCreate(savedInstanceState: Bundle?) {
19         super.onCreate(savedInstanceState)
20         setContentView(R.layout.activity_main3)
21         email=findViewById(R.id.name)
22         password=findViewById(R.id.pass)
23         check=findViewById(R.id.checkBox)
24         preff=getSharedPreferences( name: "TABLEE", MODE_PRIVATE)
25        check.isChecked=preff?.getBoolean( key: "key3", defValue: false)?:false
26         email.setText(preff?.getString( key: "key1", defValue: ""))
27         password.setText(preff?.getString( key: "key2", defValue: ""))
28     }
29     fun savestate(check:Boolean)
30     {
31         val editor=preff?.edit()
32         editor?.putBoolean("key3", check)
33         editor?.apply()
34     }
```

25 строка - для чекбокса присваивается значение состояния: нажат или нет. Значение состояния чекбокса считывается из памяти (истина - нажат, ложь - не нажат). Значение записано в таблицу (см.24 строку) под ключом “key3”. Если ничего не было записано - значение по умолчанию установлено false.

?: *– оператор "элвис"*

Оператор, обозначаемый вопросительным знаком с двоеточием (Elvis operator), подобен проверке на null в варианте if-else. Он возвращает значение слева от себя, если оно не null.

И возвращает значение справа от себя, если то, что слева, – null.

Другими словами, после elvis-оператора находится значение по-умолчанию, которое возвращается только в том случае, если выражение до ?: вернуло null.

26 и 27 строки - считываем значения текстовых полей, ранее занесенные в таблицу под ключевыми слова key1 (почта) и key2 (пароль) и устанавливаем их в качестве значения для параметра text (заполняем ими текстовое поле), если в таблицу ничего не было записано - текстовые поля останутся пустыми и в них будут отображаться подсказки - hint

29-34 строки - прописываем функцию записи состояния чекбокса (нажат или нет) и сохраняем значение под ключом “key 3” в таблице

41-46 строки - прописываем функцию удаления всех записанных данных из таблицы

47-61 строки - прописываем функцию, срабатывающую при нажатии на кнопку “Войти”:

48 и 49 строки - объявляем переменные и присваиваем им значения параметра text текстовых полей

50 строка - логической переменной присваиваем состояние чекбокса (isChecked может быть в двух состояниях - ложь (не нажат) и истина (нажат))

51 строка - если чекбокс был нажат, вызвать (52 строка) и выполнить функцию сохранения данных в память (в качестве аргументов указать переменные, значения которых нужно сохранить) и функцию сохранения состояния чекбокса (53 строка)

55 строка - иначе, если значение false, вызываем функцию очистки таблицы (57 строка)

59 строка - переходим на следующий экран (имя активити прописываете в соответствии с вашим проектом), переход будет осуществлен независимо от проверки и выполнения условий операторов if-else (обратите внимание, он прописан ниже всех проверок)

```
35          fun saveData(mail:String,pass:String){
36          val editor=preff?.edit()
37          editor?.putString("key1", mail)
38              editor?.putString("key2", pass)
39          editor?.apply()
40      }
41          fun deleteAll()
42          {
43              val editor=preff?.edit()
44              editor?.clear()
45              editor?.apply()
46          }
47      fun save22(view: View) {
48          val value:String=email.text.toString()
49          val value2:String=password.text.toString()
50          val checkboxstate:Boolean=check.isChecked
51          if (checkboxstate==true){
52          saveData(value,value2)
53              savestate(checkboxstate)
54         }
55          else
56          {
57              deleteAll()
58          }
59          val inten= Intent( packageContext: this,MainActivity2::class.java)
60          startActivity(inten)
61      }
62
63  }
```

Запустите приложение, устанавливайте и снимайте чекбокс - проанализируйте работу приложения, какие данные сохраняются в таблице. Приложение первый раз запустите из Android Studio, оно установится на телефон, повторные тестовые запуски выполняйте только с телефона. Подумайте, как работает код, корректно ли он работает? Данные, вводимые пользователем на экране входа не должны сохраняться в таблице, они должна считываться из нее и автоматически отображаться в полях, если чекбокс был нажат.

Внимательно изучите код и внесите необходимые изменения для обеспечения корректной работы приложения.

Переключитесь на экран Регистрации (3 экран вашего приложения). Вам

необходимо реализовать следующий функционал: пользователь запускает первый раз приложение, он не зарегистрирован. Нажимает на кнопку Регистрация и попадает на соответствующий экран. Заполняет текстовые поля, нажимает на кнопку Зарегистрироваться и попадает на экран Вход. Вводит свою почту и пароль и если пароль не совпадает с тем, что был введен на экране регистрации ему высвечивается сообщение об ошибке. Если пользователь введет другую почту, а не ту, что указал при регистрации, ему также должно быть показано сообщение об ошибке (через всплывающее или диалоговое окно реализовать сообщения системы - решайте самостоятельно). У пользователя должна быть возможность автоматически заполнять поля, указанные на экране входа, чтобы не заполнять поля каждый раз. При нажатии на кнопку Войти пользователь должен попасть на экран-заглушку (последний экран с картинкой, где над изображением должен отображаться следующий текст: Имя_пользователя, с возвращением).

Если вам необходимо присвоить значение из таблицы данных переменной, используйте следующую конструкцию:

```
var conter=0

conter=pref?.getInt( key: "key1", defValue: 0)!!
```

Переменная должна быть такого же типа, как и данные, указанного ключа.