Министерство образования и молодежной политики Свердловской области

ГАПОУ СО «ЕКАТЕРИНБУРГСКИЙ КОЛЛЕДЖ

ТРАНСПОРТНОГО СТРОИТЕЛЬСТВА»

СПЕЦИАЛЬНОСТЬ

Информационные системы и программирование

ПРОЕКТ

По дисциплине «УП 01.02»

Выполнил: студент группы ПР-32

Шмелев М.К

Проверил: Преподаватель

Мирошниченко Г.В

Екатеринбург, 2025

# Входные и выходные данные

**Входные данные:**

* **Пользовательский ввод**:
  + Текст, вводимый в EditText для заголовка преступления.
  + Выбор контакта из адресной книги для подозреваемого.
  + Взаимодействие с CheckBox для активации кнопки даты.
* **Разрешения**:
  + Разрешение на чтение контактов (READ\_CONTACTS).

# Выходные данные:

* **Отчет о преступлении**:
  + Формируется отчет с информацией о преступлении, включая заголовок, дату, статус решения и информацию о подозреваемом.
  + Отчет отправляется через неявный интент, позволяя пользователю выбрать приложение для отправки (например, электронная почта, мессенджер).

# Классы и код

#### Класс Crime

package com.bignerdranch.android.application\_19

import java.time.LocalDate

import java.util.UUID

data class Crime(

val id: UUID = UUID.randomUUID(),

var title: String,

var isSolved: Boolean,

val date: LocalDate,

var suspect: String = ""

)

**Класс CrimeFragment:**

class CrimeFragment : Fragment() {  
  
 private lateinit var crime: Crime  
 private lateinit var titleField: EditText  
 private lateinit var dateButton: Button  
 private lateinit var reportButton: Button  
 private lateinit var suspectButton: Button  
 private lateinit var solvedCheckBox: CheckBox  
 private lateinit var titleName: TextView  
 private var selectedContactName: String = ""  
 private var selectedContactNumber: String = ""  
  
 companion object {  
 private const val DATE\_FORMAT = "EEE, MMM dd"  
 private const val REQUEST\_CONTACT = 1  
 }  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 crime = Crime(  
 title = "Убийство",  
 isSolved = true,  
 date = LocalDate.now()  
 )  
 }  
  
 override fun onCreateView(  
 inflater: LayoutInflater,  
 container: ViewGroup?,  
 savedInstanceState: Bundle?  
 ): View? {  
 val view = inflater.inflate(R.layout.*fragment\_crime*, container, false)  
 titleField = view.findViewById(R.id.*crime\_title*)  
 dateButton = view.findViewById(R.id.*crime\_date*)  
 reportButton = view.findViewById(R.id.*send\_crime\_report*)  
 suspectButton = view.findViewById(R.id.*choose\_suspect*)  
 solvedCheckBox = view.findViewById(R.id.*crime\_solved*)  
 titleName = view.findViewById(R.id.*textView*)  
  
 // Отключение кнопки dateButton по умолчанию  
 dateButton.*isEnabled* = false  
  
 dateButton.*text* = crime.date.toString()  
  
 // Установка слушателя на CheckBox  
 solvedCheckBox.setOnCheckedChangeListener **{** \_, isChecked **->** dateButton.*isEnabled* = isChecked  
 **}** // Установка TextWatcher для обновления TextView  
 titleField.addTextChangedListener(object : TextWatcher {  
 override fun afterTextChanged(s: Editable?) {}  
  
 override fun beforeTextChanged(s: CharSequence?, start: Int, count: Int, after: Int) {}  
  
 override fun onTextChanged(s: CharSequence?, start: Int, before: Int, count: Int) {  
 titleName.*text* = s.*toString*()  
 }  
 })  
  
 reportButton.setOnClickListener **{** sendContactInfo()  
 **}** suspectButton.setOnClickListener **{** val pickContactIntent = Intent(Intent.*ACTION\_PICK*, ContactsContract.Contacts.*CONTENT\_URI*)  
 startActivityForResult(pickContactIntent, REQUEST\_CONTACT)  
 **}** dateButton.setOnClickListener **{** crime.title = titleField.*text*.toString()  
 // Обновите интерфейс или выполните другие действия  
 **}** return view  
 }  
  
 override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {  
 if (requestCode == REQUEST\_CONTACT && resultCode == Activity.*RESULT\_OK*) {  
 data?.*data*?.*let* **{** contactUri **->** try {  
 val queryFields = *arrayOf*(ContactsContract.Contacts.*DISPLAY\_NAME*)  
 requireActivity().*contentResolver*.query(contactUri, queryFields, null, null, null)?.*use* **{** cursor **->** if (cursor.moveToFirst()) {  
 selectedContactName = cursor.getString(0)  
  
 val idIndex = cursor.getColumnIndex(ContactsContract.Contacts.*\_ID*)  
 if (idIndex != -1) {  
 val id = cursor.getString(idIndex)  
 val phoneCursor = requireActivity().*contentResolver*.query(  
 ContactsContract.CommonDataKinds.Phone.*CONTENT\_URI*,  
 null,  
 ContactsContract.CommonDataKinds.Phone.*CONTACT\_ID* + " = ?",  
 *arrayOf*(id),  
 null  
 )  
 phoneCursor?.*use* **{** if (**it**.moveToFirst()) {  
 val numberIndex = **it**.getColumnIndex(ContactsContract.CommonDataKinds.Phone.*NUMBER*)  
 if (numberIndex != -1) {  
 selectedContactNumber = **it**.getString(numberIndex)  
 }  
 }  
 **}** }  
 // Обновите интерфейс  
 updateUI()  
 }  
 **}** } catch (e: Exception) {  
 Log.e("CrimeFragment", "Error querying contact", e)  
 }  
 **}** ?: *run* **{** Log.e("CrimeFragment", "Contact URI is null")  
 **}** }  
 }  
  
 private fun sendContactInfo() {  
 val message = "Контакт: $selectedContactName, $selectedContactNumber. Это ужас!"  
 Intent(Intent.*ACTION\_SEND*).*apply* **{** *type* = "text/plain"  
 putExtra(Intent.*EXTRA\_TEXT*, message)  
 putExtra(Intent.*EXTRA\_SUBJECT*, getString(R.string.*crime\_report\_subject*))  
 **}**.*also* **{** intent **->** val chooserIntent = Intent.createChooser(intent, getString(R.string.*send\_report*))  
 startActivity(chooserIntent)  
 **}** }  
  
 private fun updateUI() {  
 // Обновите интерфейс  
 suspectButton.*text* = if (selectedContactName.*isBlank*()) {  
 getString(R.string.*choose\_suspect*)  
 } else {  
 selectedContactName  
 }  
 }  
}