Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт «**Практическая работа №1\_2**

**Вариант 1**»

Выполнила: Зиятдинова Алина Ленаровна

Группа: ПР-31

2023

|  |
| --- |
| **Манифест** |
|  |

|  |
| --- |
| **DBContract** |
| package com.example.prakt1\_2\_v1  import android.provider.BaseColumns  object DBContract {   class ConsumptionEntry: BaseColumns  {  companion object{  val TABLE\_NAME = "consumption\_database"  val NAME = "name"  val CATEGORY = "category"  val DATE = "date"  val AMOUNT = "amount"  }  } } |

|  |
| --- |
| **DBHelper** |
| class DBHelper (val context: Context):SQLiteOpenHelper(context,"consumption\_database.db", null, 1) {  override fun onCreate(db: SQLiteDatabase?) {  val quary = "CREATE TABLE consumption\_database (id INT PRIMARY KEY, name TEXT, category TEXT, date TEXT, amount TEXT)"  db!!.execSQL(quary)  }   override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {  db!!.execSQL("DROP TABLE IF EXISTS consumption\_database")  onCreate(db)  }   fun addConsumption(consumption: ExpenseItem)  {  val values = ContentValues()  values.put("name", consumption.name)  values.put("category", consumption.category)  values.put("date", consumption.date)  values.put("amount", consumption.amount)  val dp = this.*writableDatabase* dp.insert("consumption\_database", null, values)  dp.close()  }   @SuppressLint("Range")  fun readallConsumption(): ArrayList<ExpenseItem> {  val movie = ArrayList<ExpenseItem>()  val db = *writableDatabase* var cursor: Cursor? = null  try {  cursor = db.rawQuery("select \* from consumption\_database", null)  } catch (e: SQLiteException) {  return ArrayList()  }  if (cursor!!.moveToFirst()) {  while (cursor.*isAfterLast* == false) {  var name = cursor.getString(cursor.getColumnIndex(DBContract.ConsumptionEntry.NAME)).toString()  var category = cursor.getString(cursor.getColumnIndex(DBContract.ConsumptionEntry.CATEGORY)).toString()  var date = cursor.getString(cursor.getColumnIndex(DBContract.ConsumptionEntry.DATE)).toString()  var sum = cursor.getString(cursor.getColumnIndex(DBContract.ConsumptionEntry.AMOUNT)).toString()  movie.add(ExpenseItem(name,category,date,sum))  cursor.moveToNext()  }  }  return movie  }    @Throws(SQLiteConstraintException::class)  fun deleteConsumption(title: String): Boolean {  val db = *writableDatabase* val selection = DBContract.ConsumptionEntry.NAME + " LIKE ?"  val selectionArgs = *arrayOf*(title)  db.delete(DBContract.ConsumptionEntry.TABLE\_NAME, selection, selectionArgs)  return true  }   @Throws(SQLiteConstraintException::class)  fun getConsumption(title: String): Boolean {  val db = this.*readableDatabase* val result = db.rawQuery("SELECT \* FROM consumption\_database WHERE name = '$title'",null)  return result.moveToFirst()  }   fun deleteAllConsumption(): Boolean {  val db = *writableDatabase* db.delete("consumption\_database", null, null)  return true  }   fun editConsumptionInfo(name: String, category: String, date: String, amount:String): Boolean  {  val db = *writableDatabase* val result = db.rawQuery("UPDATE consumption\_database SET name = '$name', date='$date', category='$category', amount='$amount' WHERE name = '$name'",null)  return result.moveToFirst()  } } |

|  |
| --- |
| **ExpenseAdapter** |
| class ExpenseAdapter(private val expenseList: List<ExpenseItem>,  private val clickListener: Fragment1 ) :  RecyclerView.Adapter<ExpenseAdapter.ExpenseViewHolder>() {   override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ExpenseViewHolder {  val view =  LayoutInflater.from(parent.*context*).inflate(R.layout.*recycler\_disain*, parent, false)  return ExpenseViewHolder(view)  }   override fun onBindViewHolder(holder: ExpenseViewHolder, position: Int) {   val expense = expenseList[position]  holder.bind(expense)   // Обработка нажатия на элемент списка  holder.itemView.setOnClickListener **{** clickListener.onExpenseItemClick(expense)  **}** }   override fun getItemCount(): Int {  return expenseList.size  }   class ExpenseViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {  private val nameTextView: TextView = itemView.findViewById(R.id.*nameTextView*)  private val amountTextView: TextView = itemView.findViewById(R.id.*amountTextView*)   fun bind(expense: ExpenseItem) {  nameTextView.*text* = expense.name  amountTextView.*text* = "${expense.amount} рублей"  }  } } |

|  |
| --- |
| **ExpenseAdapter2** |
| class ExpenseAdapter2(private val expenseList: List<ExpenseItem>,  private val clickListener: Fragment2 ) :  RecyclerView.Adapter<ExpenseAdapter2.ExpenseViewHolder>() {   override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ExpenseViewHolder {  val view =  LayoutInflater.from(parent.*context*).inflate(R.layout.*recycler\_disain*, parent, false)  return ExpenseViewHolder(view)  }   override fun onBindViewHolder(holder: ExpenseViewHolder, position: Int) {  val expense = expenseList[position]  holder.bind(expense)  holder.itemView.setOnClickListener **{** clickListener.onExpenseItemClick(expense)  **}** }    override fun getItemCount(): Int {  return expenseList.size  }   class ExpenseViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {  private val nameTextView: TextView = itemView.findViewById(R.id.*nameTextView*)  private val amountTextView: TextView = itemView.findViewById(R.id.*amountTextView*)   fun bind(expense: ExpenseItem) {  nameTextView.*text* = expense.name  amountTextView.*text* = "${expense.amount} рублей"  }  } } |

|  |
| --- |
| **ExpenseItem** |
| data class ExpenseItem(  var name: String, var category: String, var date: String, var amount: String ) |

**Фрагмент 1**

|  |
| --- |
| class Fragment1 : Fragment() {  // *TODO: Rename and change types of parameters* private var param1: String? = null  private var param2: String? = null   private lateinit var nameEditText: EditText  private lateinit var dateEditText: EditText  private lateinit var amountEditText: EditText  private lateinit var spinner: Spinner  private lateinit var addButton: Button  private lateinit var datePicker:DatePicker  private val expenseList = ArrayList<ExpenseItem>()    override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  *arguments*?.*let* **{** param1 = **it**.getString(*ARG\_PARAM1*)  param2 = **it**.getString(*ARG\_PARAM2*)  **}** }   override fun onCreateView(  inflater: LayoutInflater, container: ViewGroup?,  savedInstanceState: Bundle?  ): View? {   val fragmentLayout=inflater.inflate(R.layout.*fragment\_1*, container, false)  val navController=NavHostFragment.findNavController(this)   datePicker = fragmentLayout.findViewById<DatePicker>(R.id.*date\_Consumption*)   nameEditText = fragmentLayout.findViewById(R.id.*name\_edittext*)  dateEditText = fragmentLayout.findViewById(R.id.*name\_edittext*)  amountEditText = fragmentLayout.findViewById(R.id.*sum\_edittext*)  spinner = fragmentLayout.findViewById(R.id.*spinner*)  addButton = fragmentLayout.findViewById(R.id.*button\_addConsumtoin*)   val db = DBHelper (requireContext())   addButton.setOnClickListener **{** val check = db.getConsumption(nameEditText.*text*.toString())  if (check == false)  {  addExpense()  }  else  {  Toast.makeText(*activity*, "Такие данные уже есть", Toast.*LENGTH\_SHORT*).show()  }  **}** return fragmentLayout   }  fun onExpenseItemClick(expenseItem: ExpenseItem) {  // Обработка нажатия на элемент списка  val bundle = Bundle()  bundle.putString("category",expenseItem.category)  bundle.putString("name", expenseItem.name)  bundle.putString("date", expenseItem.date)  bundle.putString("amount", expenseItem.amount)   val navController = NavHostFragment.findNavController(this)  navController.navigate(R.id.*action\_fragment1\_to\_fragment2*, bundle)  }    private fun addExpense() {  var day = datePicker.*dayOfMonth* var month = datePicker.*month* + 1  var year = datePicker.*year* var string\_category = "${spinner.*selectedItem*.toString()}"  var string\_date = "$day.$month.$year"  var string\_name = nameEditText.*text*.*trim*().toString()  var string\_sum = amountEditText.*text*.*trim*().toString()   if (string\_category.*isNotEmpty*() && string\_date.*isNotEmpty*() && string\_name.*isNotEmpty*() && string\_sum.*isNotEmpty*()) {  val expenseItem = ExpenseItem(string\_name,string\_category,string\_date,string\_sum)  expenseList.add(expenseItem)  val db = DBHelper (requireContext())  db.addConsumption(expenseItem)  // Очищаем поля ввода  nameEditText.*text*.clear()  dateEditText.*text*.clear()  amountEditText.*text*.clear()  }  else {  Toast.makeText(*activity*, "Введите корректные данные", Toast.*LENGTH\_SHORT*).show()  }   }  } |
| <TextView  android:layout\_width="match\_parent"  android:layout\_height="25dp"  android:textSize="18dp"  android:textColor="@color/black"  android:layout\_marginRight="10dp"  android:layout\_marginTop="20dp"  android:layout\_marginLeft="10dp"  android:text="@string/textview\_NameConsumption"> </TextView> <EditText  android:id="@+id/name\_edittext"  android:layout\_width="match\_parent"  android:layout\_height="40dp"  android:background="@drawable/background"  android:paddingLeft="5dp"  android:layout\_marginRight="10dp"  android:layout\_marginLeft="10dp"  ></EditText> <TextView  android:layout\_width="match\_parent"  android:layout\_height="25dp"  android:textSize="18dp"  android:textColor="@color/black"  android:layout\_marginRight="10dp"  android:layout\_marginTop="10dp"  android:layout\_marginLeft="10dp"  android:text="@string/textview\_SumComsumption"> </TextView> <EditText  android:id="@+id/sum\_edittext"  android:layout\_width="match\_parent"  android:layout\_height="40dp"  android:background="@drawable/background"  android:paddingLeft="5dp"  android:layout\_marginRight="10dp"  android:layout\_marginLeft="10dp"  ></EditText> <TextView  android:layout\_width="match\_parent"  android:layout\_height="25dp"  android:textSize="18dp"  android:textColor="@color/black"  android:layout\_marginRight="10dp"  android:layout\_marginTop="10dp"  android:layout\_marginLeft="10dp"  android:text="@string/textview\_CategoryConsumption"> </TextView> <Spinner  android:id="@+id/spinner"  android:layout\_width="match\_parent"  android:layout\_height="40dp"  android:entries="@array/category"  android:layout\_marginLeft="5dp"  android:layout\_marginRight="10dp"  android:hint="@string/edittext\_CategoryConsumption"></Spinner> <TextView  android:layout\_width="match\_parent"  android:layout\_height="25dp"  android:textSize="18dp"  android:textColor="@color/black"  android:layout\_marginRight="10dp"  android:layout\_marginTop="10dp"  android:layout\_marginLeft="10dp"  android:text="@string/textview\_DateConsumption"> </TextView>  <DatePicker  android:id="@+id/date\_Consumption"  android:layout\_width="392dp"  android:layout\_height="200dp"  android:layout\_marginLeft="10dp"  android:layout\_marginRight="10dp"  android:calendarViewShown="false"  android:datePickerMode="spinner"></DatePicker> <Button  android:id="@+id/button\_addConsumtoin"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginRight="10dp"  android:layout\_marginTop="10dp"  android:layout\_marginLeft="10dp"  android:text="@string/addСonsumption"> </Button> |
|  |

**Фрагмент 2**

|  |
| --- |
| class Fragment2 : Fragment() {  // *TODO: Rename and change types of parameters* private var param1: String? = null  private var param2: String? = null    lateinit var recyclerView: RecyclerView  lateinit var DBHelper : DBHelper  lateinit var expenseList: ArrayList<ExpenseItem>  lateinit var adapter: ExpenseAdapter2  lateinit var delButton: Button  lateinit var edittext: EditText  lateinit var editButton: Button  lateinit var expense: ExpenseItem  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  *arguments*?.*let* **{** param1 = **it**.getString(*ARG\_PARAM1*)  param2 = **it**.getString(*ARG\_PARAM2*)  **}** }  fun onExpenseItemClick(expenseItem: ExpenseItem) {  val itemValue = expenseItem.name  var amount: String = expenseItem.amount  val builder = AlertDialog.Builder(requireContext())  builder.setTitle("Вы выбрали: $itemValue")  .setCancelable(true)  .setPositiveButton("Удалить")  **{** \_, \_ **->** val result = DBHelper.deleteConsumption(itemValue)  if (result)  {  expenseList.remove(expenseItem)  update()  }  **}** .setNegativeButton("Редактировать")  **{** \_, \_ **->** expense = expenseItem  **}** val dialog = builder.create()  dialog.show()  }  @SuppressLint("MissingInflatedId")  override fun onCreateView(  inflater: LayoutInflater, container: ViewGroup?,  savedInstanceState: Bundle?  ): View? {  val fragmentLayout =inflater.inflate(R.layout.*fragment\_2*, container, false)  recyclerView = fragmentLayout.findViewById(R.id.*recuclerview*)  delButton = fragmentLayout.findViewById(R.id.*del\_Consumtoins*)  edittext = fragmentLayout.findViewById(R.id.*sum\_edittext*)  editButton = fragmentLayout.findViewById(R.id.*edit\_Consumtoins*)  update()  delButton.setOnClickListener **{** val result = DBHelper.deleteAllConsumption()  if (result)  {  expenseList.clear()  update()  Toast.makeText(*activity*, "Все данные удалены", Toast.*LENGTH\_SHORT*).show()  }  **}** editButton.setOnClickListener **{** val db = DBHelper (requireContext())  var check = db.editConsumptionInfo(expense.name, expense.category, expense.date, edittext.*text*.toString())  if (check)  {  update()  Toast.makeText(*activity*, "Данные успешно обновлены", Toast.*LENGTH\_SHORT*).show()  }  else  {  Toast.makeText(*activity*, "Данные успешно обновлены", Toast.*LENGTH\_SHORT*).show()  }  **}** return fragmentLayout  }  fun update()  {  DBHelper = DBHelper(requireContext())  expenseList = DBHelper.readallConsumption()  adapter = ExpenseAdapter2(expenseList, this)  recyclerView.*adapter* = adapter  recyclerView.*layoutManager* = LinearLayoutManager(*activity*)  } } |
| <TextView  android:id="@+id/textMain"  android:layout\_gravity="center"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="@string/Redactor"  android:textColor="@color/black"  android:textSize="25dp"  ></TextView>  <android.support.v7.widget.RecyclerView  android:id="@+id/recuclerview"  android:layout\_width="match\_parent"  android:layout\_height="311dp"  android:layout\_marginLeft="10dp"  android:layout\_marginTop="1dp"  android:layout\_marginRight="10dp"  android:divider="@android:color/darker\_gray"  android:dividerHeight="1dp"  android:paddingTop="10dp"  android:paddingBottom="10dp" />  <Button  android:id="@+id/del\_Consumtoins"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginRight="10dp"  android:layout\_marginTop="10dp"  android:layout\_marginLeft="10dp"  android:text="@string/delСonsumptions"> </Button> <TextView  android:layout\_width="match\_parent"  android:layout\_height="25dp"  android:textSize="18dp"  android:textColor="@color/black"  android:layout\_marginRight="10dp"  android:layout\_marginTop="10dp"  android:layout\_marginLeft="10dp"  android:text="@string/textview\_SumComsumption"> </TextView> <EditText  android:id="@+id/sum\_edittext"  android:layout\_width="match\_parent"  android:layout\_height="40dp"  android:background="@drawable/background"  android:paddingLeft="5dp"  android:layout\_marginRight="10dp"  android:layout\_marginLeft="10dp"  ></EditText> <Button  android:id="@+id/edit\_Consumtoins"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginRight="10dp"  android:layout\_marginTop="10dp"  android:layout\_marginLeft="10dp"  android:text="@string/edit"> </Button> |
|  |

|  |
| --- |
| class MainActivity : AppCompatActivity() {    private lateinit var navController: NavController  @SuppressLint("MissingInflatedId")  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.*activity\_main*)  val host:NavHostFragment=*supportFragmentManager* .findFragmentById(R.id.*navFragment*)as NavHostFragment? ?:return  val navController=host.*navController* val sideBar=findViewById<NavigationView>(R.id.*nav\_view*)  sideBar?.*setupWithNavController*(navController)  val drawerLayout = findViewById<android.support.v4.widget.DrawerLayout>(R.id.*drawer\_layout*)  var appBarConfiguration=*AppBarConfiguration*(navController.*graph*,drawerLayout )  val toolbar=findViewById<Toolbar>(R.id.*toolbar2*)  setSupportActionBar(toolbar)  toolbar.*setupWithNavController*(navController,appBarConfiguration)    }  // включаем верхнее меню (кнопка вверху справа)  override fun onCreateOptionsMenu(menu: Menu): Boolean {  *menuInflater*.inflate(R.menu.*top\_right\_menu*, menu)  return true  }   // переходим по ссылкам верхнего меню  override fun onOptionsItemSelected(item: MenuItem): Boolean {   return item.*onNavDestinationSelected*(navController)  || super.onOptionsItemSelected(item)  }  } |
| <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical">  <android.support.v7.widget.Toolbar  android:id="@+id/toolbar2"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:background="?attr/colorPrimary"  android:minHeight="?attr/actionBarSize"  android:theme="?attr/actionBarTheme" />  <fragment  android:name="androidx.navigation.fragment.NavHostFragment"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  app:navGraph="@navigation/routes"  app:defaultNavHost="true"  android:id="@+id/navFragment"/> </LinearLayout>   <android.support.design.widget.NavigationView  android:id="@+id/nav\_view"  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:layout\_gravity="start"  android:fitsSystemWindows="true"  app:menu="@menu/drawer\_menu"/> |

|  |
| --- |
| <TextView  android:id="@+id/nameTextView"  android:layout\_width="162dp"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="20dp"  android:background="@color/light\_gray"  android:padding="8dp"  android:textColor="@android:color/black"  android:textSize="15dp" />  <TextView  android:id="@+id/amountTextView"  android:layout\_width="191dp"  android:layout\_height="wrap\_content"  android:background="@color/light\_gray"  android:padding="8dp"  android:textColor="@android:color/black"  android:textSize="15dp" /> |

**Блок-схема**

|  |
| --- |
| **Фрагмент 1** |
|  |

|  |
| --- |
| **Фрагмент 2** |
|  |

**Тестовые ситуации**

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |