Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическое задание №19\_2**»

Выполнил: Зиятдинова Алина Ленаровна

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2023

**Задание:**

Работа с ListView и файлами в формате json. Реализовать программу в соответствии с вариантом из практики 18 - приложение «Учет расходов».

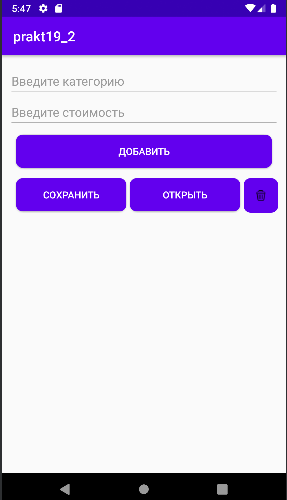
**Листинг программы**

|  |
| --- |
| class Consumption (var catigories: String, var price: Int ) {  override fun toString(): String {  return "Категория: $catigories - $price руб."  } } |
| object ConsumptionJSONHelper {  private val FILE\_NAME = "data.json"  fun exportToJSON(context: Context, dataList:List<Consumption?>?):Boolean  {  val gson= Gson()  val dataItems=DataItems()  dataItems.consumptions=dataList  val jsonString:String = gson.toJson(dataItems)  try {  context.openFileOutput(FILE\_NAME, Context.*MODE\_PRIVATE*).*use***{** fileOutPutStream**->**fileOutPutStream.write(jsonString.*toByteArray*()) **}** return true  }  catch (e:Exception)  {  e.printStackTrace()  }  return false  }   fun importFromJSON(context: Context):List<Consumption?>?{  try {  context.openFileInput(FILE\_NAME).*use* **{**fileInputStream**->** InputStreamReader(fileInputStream).*use***{** streamReader**->** val gson= Gson()  val dataItems:DataItems=gson.fromJson(streamReader,DataItems::class.*java*)  return dataItems.consumptions  **}  }** }  catch (ex: IOException)  {  ex.printStackTrace()  }  return null  }  private class DataItems{  var consumptions: List<Consumption?>?=null  } } |
| class MainActivity : AppCompatActivity() {  lateinit var catigories\_edittext: EditText  lateinit var price\_edittext: EditText  lateinit var adapter: ArrayAdapter<Consumption>  lateinit var listview: ListView  private lateinit var consumptions: ArrayList<Consumption>  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.*activity\_main*)  catigories\_edittext = findViewById(R.id.*category\_et*)  price\_edittext = findViewById(R.id.*price\_et*)  listview = findViewById(R.id.*list\_view*)  consumptions = ArrayList()  adapter = ArrayAdapter(this, android.R.layout.*simple\_list\_item\_1*, consumptions)  listview.*adapter* = adapter  }   fun save(view: View) {  val result = ConsumptionJSONHelper.exportToJSON(this, consumptions)  if (result) {  var toast = Toast.makeText(this, "Данные сохранены", Toast.*LENGTH\_LONG*)  toast.setGravity(Gravity.*TOP*, 0,160)  toast.show()  } else {  var toast = Toast.makeText(this, "Не удалось сохранить данные", Toast.*LENGTH\_LONG*)  toast.setGravity(Gravity.*TOP*, 0,160)  toast.show()  }   }   fun add(view: View) {  try {  if (catigories\_edittext.*text*.toString().*isNotEmpty*() &&  price\_edittext.*text*.toString().*isNotEmpty*()) {  val category\_add = catigories\_edittext.*text*.toString()  val price\_add = price\_edittext.*text*.toString().*toInt*()  val consumption\_add = Consumption(category\_add, price\_add)  consumptions.add(consumption\_add)  adapter.notifyDataSetChanged()  var toast = Toast.makeText(this, R.string.*adding*, Toast.*LENGTH\_SHORT*)  toast.setGravity(Gravity.*TOP*, 0,160)  toast.show()  }  else  {  var toast = Toast.makeText(this, R.string.*emptyfields*, Toast.*LENGTH\_SHORT*)  toast.setGravity(Gravity.*TOP*, 0,160)  toast.show()  }  price\_edittext.*text* = null  catigories\_edittext.*text* = null  }  catch(e: Exception)  {  val toast = Toast.makeText(this,R.string.*excaption*, Toast.*LENGTH\_SHORT*)  toast.setGravity(Gravity.*TOP*, 0,160)  toast.show()  }  }   fun open(view: View) {  consumptions = ConsumptionJSONHelper.importFromJSON(this) as ArrayList<Consumption>  if (consumptions != null) {  adapter.clear()  adapter.addAll(consumptions)  var toast = Toast.makeText(this, "Данные восстановлены", Toast.*LENGTH\_LONG*)  toast.setGravity(Gravity.*TOP*, 0,160)  toast.show()  } else {  var toast = Toast.makeText(this, "Не удалось восстановить данные", Toast.*LENGTH\_LONG*)  toast.setGravity(Gravity.*TOP*, 0,160)  toast.show()  }  }   fun remove(view: View) {  consumptions.clear()  adapter.clear()  var toast = Toast.makeText(this, "Все данные удалены", Toast.*LENGTH\_LONG*)  toast.setGravity(Gravity.*TOP*, 0,160)  toast.show()  } } |

**Блок схема**

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

**Экраны**

****

**Тестовые ситуации**

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