Міністерство освіти і науки України  
Національний технічний університет України

«Київський політехнічний інститут  ім. І. Сікорського»

Кафедра інженерії програмного забезпечення в енергетиці

Лабораторна робота № 1

з курсу: «Розробка програмного забезпечення мобільних пристроїв»

**Виконав:**  
студент 4-го курсу,  
групи ТВ-11

Гущін Володимир Сергійович

Посилання на GitHub репозиторій:

https://github.com/HushchinVolodymyr/MobileDevLr\_1

**Перевірив:**

Недашківський О.Л.

Київ 2024/2025

Завдання 1:

Написати мобільний калькулятор для розрахунку складу сухої та горючої маси палива та нижчої теплоти згоряння для робочої, сухої та горючої маси за заданим складом компонентів палива, що задаються у вигляді значень окремих компонентів типу: H %, C %, S %, N %, O %, W %, A %.

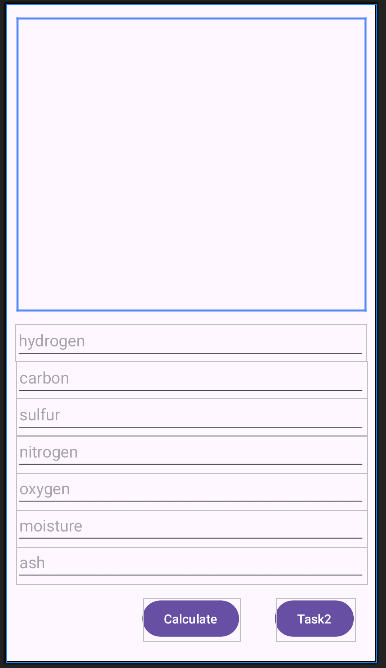
Завдання 2:

Написати мобільний калькулятор для перерахунку елементарного складу та нижчої теплоти згоряння мазуту на робочу масу для складу горючої маси мазуту, що задається наступними параметрами: вуглець, %; водень, %; кисень, %; сірка, %; нижча теплота згоряння горючої маси мазуту, МДж/кг; вологість робочої маси палива, %; зольність сухої маси, %; вміст ванадію (V), мг/кг.

Хід виконання:

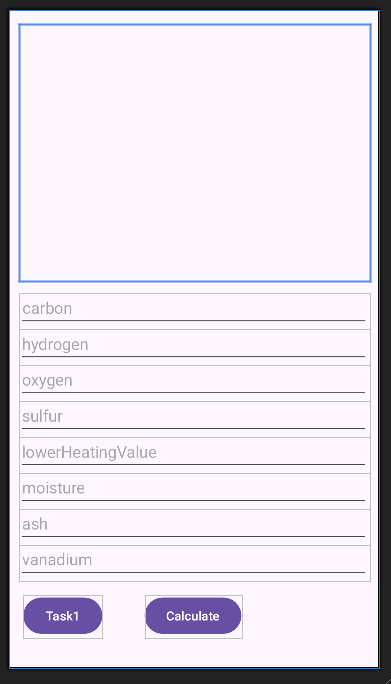
Для кожного завдання були спроектовані відповідні сторінки:

Завдання 1:



Сторінка моє поле виводу обчислень та поля для вводу заначень, також присутня кнопка для початку розрахунків та кнопка до переходу до калькулятора для іншого завдання.

Завдання 2:



Має ту саму структуру як і для першого завдання.

Для обчислень буть створений клас з 3 методами два з обчисленнями для 1-го та 2-го завдання й один допоміжний для 2 завдання:

class Calculator() {  
 fun calculateFuelParametrs(*fuel*: FuelComponents?): *Map*<String, Any> {  
 if(*fuel* != null) {  
 val kRS = 100 / (100 - *fuel*.moisture)  
 val kRG = 100 / (100 - *fuel*.moisture - *fuel*.ash)  
  
 val hydrogenDry = *fuel*.hydrogen \* kRS  
 val carbonDry = *fuel*.carbon \* kRS  
 val sulfurDry = *fuel*.sulfur \* kRS  
 val nitrogenDry = *fuel*.nitrogen \* kRS  
 val oxygenDry = *fuel*.oxygen \* kRS  
 val ashDry = *fuel*.ash \* kRS  
  
 val hydrogenCombustible = *fuel*.hydrogen \* kRG  
 val carbonCombustible = *fuel*.carbon \* kRG  
 val sulfurCombustible = *fuel*.sulfur \* kRG  
 val nitrogenCombustible = *fuel*.nitrogen \* kRG  
 val oxygenCombustible = *fuel*.oxygen \* kRG  
  
 val lowerHeatingValueWorking = (339 \* *fuel*.carbon + 1030 \* *fuel*.hydrogen - 108.8 \* (*fuel*.oxygen - *fuel*.sulfur) - 25 \* *fuel*.moisture) / 1000  
  
 val lowerHeatingValueDry = (lowerHeatingValueWorking + 0.025 \* *fuel*.moisture) \* (100 / (100 - *fuel*.moisture))  
  
 val lowerHeatingValueCombustible = (lowerHeatingValueWorking + 0.025 \* *fuel*.moisture) \* (100 / (100 - *fuel*.moisture - *fuel*.ash))  
  
 return *mapOf* (  
 "kRS" *to* kRS,  
 "kRG" *to* kRG,  
 "hydrogenDry" *to* "$hydrogenDry %",  
 "carbonDry" *to* "$carbonDry %",  
 "sulfurDry" *to* "$sulfurDry %",  
 "nitrogenDry" *to* nitrogenDry,  
 "oxygenDry" *to* "$oxygenDry %",  
 "ashDry" *to* "$ashDry %",  
 "hydrogenCombustible" *to* "$hydrogenCombustible %",  
 "carbonCombustible" *to* "$carbonCombustible %",  
 "sulfurCombustible" *to* "$sulfurCombustible %",  
 "nitrogenCombustible" *to* nitrogenCombustible,  
 "oxygenCombustible" *to* "$oxygenCombustible %",  
 "lowerHeatingValueWorking" *to* "$lowerHeatingValueWorking МДж/кг",  
 "lowerHeatingValueDry" *to* "$lowerHeatingValueDry МДж/кг",  
 "lowerHeatingValueCombustible" *to* "$lowerHeatingValueCombustible МДж/кг"  
 )  
 } else return *mapOf*("error" *to* "Fuel components are missing")  
  
 }  
  
 fun wokingCalc(*element*: Float, *lowerHeatingsComponents*: LowerHeatingComponents): Float {  
 return *element* \* (100 - *lowerHeatingsComponents*.moisture - *lowerHeatingsComponents*.ash) / 100  
 }  
  
 fun calculateLowerHeatingValue(*lowerHeatingsComponents*: LowerHeatingComponents?): *Map*<String, Any> {  
 if (*lowerHeatingsComponents* != null) {  
 val moistureFraction = *lowerHeatingsComponents*.moisture / 100  
 val ashFraction = *lowerHeatingsComponents*.ash / 100  
  
 val carbonWorking = this.wokingCalc(*lowerHeatingsComponents*.carbon, *lowerHeatingsComponents*)  
 val hydrogenWorking = this.wokingCalc(*lowerHeatingsComponents*.hydrogen, *lowerHeatingsComponents*)  
 val oxygenWorking = this.wokingCalc(*lowerHeatingsComponents*.oxygen, *lowerHeatingsComponents*)  
 val sulfurWorking = this.wokingCalc(*lowerHeatingsComponents*.sulfur, *lowerHeatingsComponents*)  
 val ashWorking = this.wokingCalc(*lowerHeatingsComponents*.ash, *lowerHeatingsComponents*)  
 val vanadiumWorking = this.wokingCalc(*lowerHeatingsComponents*.vanadium, *lowerHeatingsComponents*)  
  
 val heatingValueWorkingMass = *lowerHeatingsComponents*.lowerHeatingValue \* (1 - moistureFraction) \* (1 - ashFraction)  
  
 return *mapOf*(  
 "carbonWorkingMass" *to* "${carbonWorking} %",  
 "hydrogenWorkingMass" *to* "${hydrogenWorking} %",  
 "oxygenWorkingMass" *to* "${oxygenWorking} %",  
 "sulfurWorkingMass" *to* "${sulfurWorking} %",  
 "ashWorkingMass" *to* "${ashWorking} %",  
 "vanadiumWorkingMass" *to* "${vanadiumWorking} %",  
 "lowerHeatingValueWorkingMass" *to* "%.2f МДж/кг".*format*(heatingValueWorkingMass)  
 )  
 } else return return *mapOf*("error" *to* "Fuel components are missing")  
 }  
}

Як параметри для функцій створені об’єкти для складу палив:

1 завдання:

data class FuelComponents (  
 val hydrogen: Float,  
 val carbon: Float,  
 val sulfur: Float,  
 val nitrogen: Float,  
 val oxygen: Float,  
 val moisture: Float,  
 val ash: Float  
)

2 завдання:

data class LowerHeatingComponents (  
 val carbon: Float,  
 val hydrogen: Float,  
 val oxygen: Float,  
 val sulfur: Float,  
 val lowerHeatingValue: Float,  
 val moisture: Float,  
 val ash: Float,  
 val vanadium: Float,  
)

Кожна сторінка має певну логіку с отриманням полів вводу, полів виводу, та кнопок

Завдання 1:

val resultText: TextView = findViewById(R.id.*resultText*)  
  
val hydrogenInput: EditText = findViewById(R.id.*hydrogenInput*)  
val carbonInput: EditText = findViewById(R.id.*carbonInput*)  
val sulfurInput: EditText = findViewById(R.id.*sulfurInput*)  
val nitrogenInput: EditText = findViewById(R.id.*nitrogenInput*)  
val oxygenInput: EditText = findViewById(R.id.*oxygenInput*)  
val moistureInput: EditText = findViewById(R.id.*moistureInput*)  
val ashInput: EditText = findViewById(R.id.*ashInput*)  
  
val calcButton: Button = findViewById(R.id.*calcButton*)  
val toTask2Button: Button = findViewById(R.id.*toTask2*)

Завдання 2:

val resultText: TextView = findViewById(R.id.*resultText2*)  
  
val carbonInput2: EditText = findViewById(R.id.*carbonInput2*)  
val hydrogenInput2: EditText = findViewById(R.id.*hydrogenInput2*)  
val oxygenInput2: EditText = findViewById(R.id.*oxygenInput2*)  
val sulfurInput2: EditText = findViewById(R.id.*sulfurInput2*)  
val lowerHeatingValueInput: EditText = findViewById(R.id.*lowerHeatingValueInput*)  
val moistureInput2: EditText = findViewById(R.id.*moistureInput2*)  
val ashInput2: EditText = findViewById(R.id.*ashInput2*)  
val vanadiumInput2: EditText = findViewById(R.id.*vanadiumInput2*)  
  
val calcButton2: Button = findViewById(R.id.*calcButton2*)  
val toTask1Button: Button = findViewById(R.id.*toTask1*)

Для кнопок переходу між сторінками логіка однакова різниця у посиланнях на сторінки:

toTask2Button.setOnClickListener **{** val intent = Intent(this, task2::class.*java*)  
 startActivity(intent)  
**}**

У обох сторінках при натисканні кнопок «Calculate» виконуються однакові дії

1. Отримується інформація з полів вводу
2. Інформація формується у об’єкт складу палива
3. Створюється об’єкт калькулятору та викликається функція для обчислення яка приймає об’єкт палива
4. Формується та виводиться результат обчислень

Завдання 1:

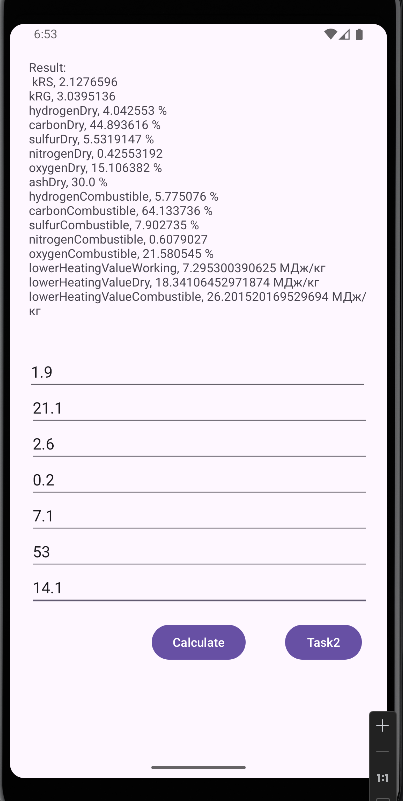
calcButton.setOnClickListener **{** val hydrogen = hydrogenInput.*text*.toString().*toFloatOrNull*() ?: 0f  
 val carbon = carbonInput.*text*.toString().*toFloatOrNull*() ?: 0f  
 val sulfur = sulfurInput.*text*.toString().*toFloatOrNull*() ?: 0f  
 val nitrogen = nitrogenInput.*text*.toString().*toFloatOrNull*() ?: 0f  
 val oxygen = oxygenInput.*text*.toString().*toFloatOrNull*() ?: 0f  
 val moisture = moistureInput.*text*.toString().*toFloatOrNull*() ?: 0f  
 val ash = ashInput.*text*.toString().*toFloatOrNull*() ?: 0f  
  
 val fuel = FuelComponents(hydrogen, carbon, sulfur, nitrogen, oxygen, moisture, ash)  
  
 val calculator = Calculator()  
  
 val calcResults = calculator.calculateFuelParametrs(fuel)  
  
 var result = ""  
  
 calcResults.*forEach* **{**(key, value) **->** result += "$key, $value\n"  
 **}** resultText.*text* = "Result: \n $result"  
**}**

Завдання 2:

calcButton2.setOnClickListener **{** val carbon = carbonInput2.*text*.toString().*toFloatOrNull*() ?: 0f  
 val hydrogen = hydrogenInput2.*text*.toString().*toFloatOrNull*() ?: 0f  
 val oxygen = oxygenInput2.*text*.toString().*toFloatOrNull*() ?: 0f  
 val sulfur = sulfurInput2.*text*.toString().*toFloatOrNull*() ?: 0f  
 val lowerHeatingValue = lowerHeatingValueInput.*text*.toString().*toFloatOrNull*() ?: 0f  
 val moisture = moistureInput2.*text*.toString().*toFloatOrNull*() ?: 0f  
 val ash = ashInput2.*text*.toString().*toFloatOrNull*() ?: 0f  
 val vanadium = vanadiumInput2.*text*.toString().*toFloatOrNull*() ?: 0f  
  
 val lowerHeatingComponents = LowerHeatingComponents(carbon, hydrogen, oxygen, sulfur, lowerHeatingValue, moisture, ash, vanadium)  
  
 val calculator = Calculator()  
  
 val calcResult = calculator.calculateLowerHeatingValue(lowerHeatingComponents)  
  
 var result = ""  
  
 calcResult?.*forEach* **{** (key, value) **->** result += "$key, $value\n"  
 **}** resultText.*text* = "Result: \n$result"  
**}**

Контрольний приклад для перевірки:

Завдання 1:



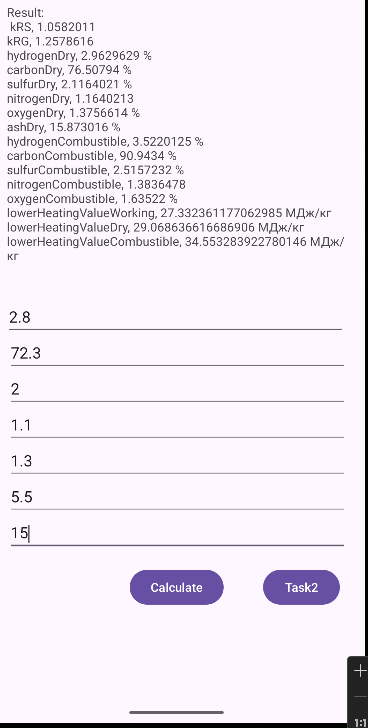
Завдання 2:



Результат обчислень варіанту:

(В моєму випадку варіант 5 номер студ. кв. 14261905)

Завдання 1:



Висновок: Результатами роботу є опанування базових навичок використання мови Kotlin та AndroidStudio, створення графічного інтерфейсі застосунку вбудованим редактором, було розроблено мобільний застосунок калькулятор для обчислення складу сухої та горючої маси палива та нижчої теплоти згоряння для робочої, сухої та горючої маси за заданим складом компонентів палива.