Міністерство освіти і науки України  
Національний технічний університет України

«Київський політехнічний інститут  ім. І. Сікорського»

Кафедра інженерії програмного забезпечення в енергетиці

Лабораторна робота № 6

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\_6

**Перевірив:**

Недашківський О.Л.

Київ 2024/2025

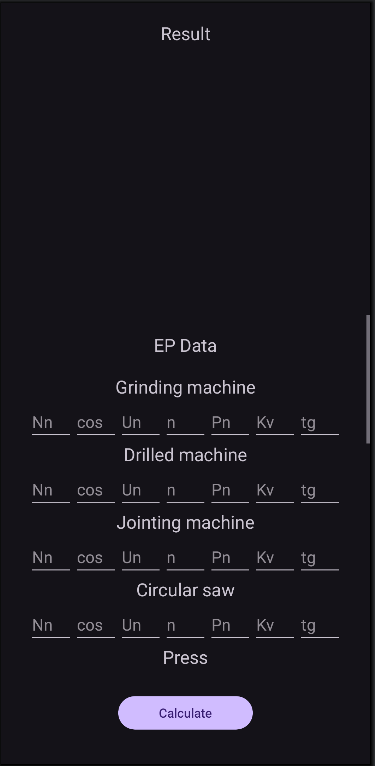
Завдання:

Створіть мобільний калькулятор для розрахунку електричних навантажень об’єктів з використанням методу впорядкованих діаграм.

Цехова мережа складається з трьох типових цехів які під’єднується до трьох різних розподільчих шин (ШР1-ШР3) та кількох крупних електроприймачів (ЕП). Для спрощення приймемо що склад, номенклатура і характеристики ЕП всіх трьох цехів однакові.

На основі складу ЕП та їх характеристик необхідно розрахувати силове навантаження цехової мережі.

Інтерфейс системи:



Об’єкти для зберігання даних:

*data class* ShrData (  
 *val* countEp: Int,  
 *val* nPGroup: Int,  
 *val* useKefGroup: Double,  
 *val* npkGroup: Double,  
 *val* npktgGroup: Double,  
 *val* npPow: Double,  
 *val* effectionCountity: Double,  
 *val* activPowerKefGroup: Double,  
 *val* activLoadKefGroup: Double,  
 *val* reactLoadKefGroup: Double,  
 *val* fullPower: Double,  
 *val* groupCurrent: Double,  
)

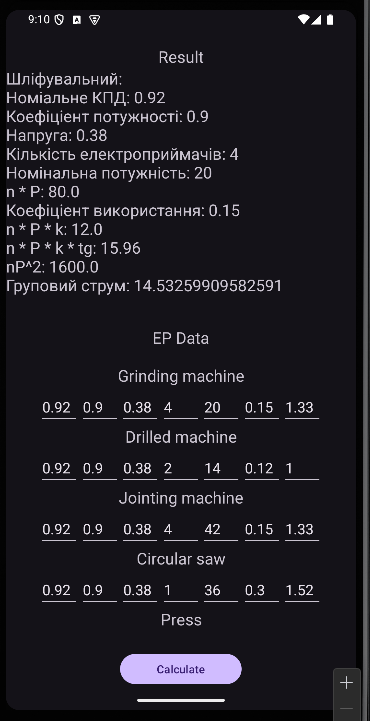
*data class* EpData (  
 *val* kpd: Double,  
 *val* cos: Double,  
 *val* power: Double,  
 *val* countEp: Int,  
 *val* numPower: Int,  
 *var* nP: Double = *0.0*,  
 *val* useKef: Double,  
 *val* reactKef: Double,  
 *var* npk: Double = *0.0*,  
 *var* npktg: Double = *0.0*,  
 *var* npPow: Double = *0.0*,  
 *var* groupCurrent: Double =*0.0*,   
)

*data class* OverloadData (  
 *val* countEp: Int,  
 *val* nP: Int,  
 *val* useKef: Double,  
 *val* npk: Double,  
 *val* npktg: Double,  
 *val* npPow: Double,  
 *val* effectionCountity: Double,  
 *val* activPowerKefGroup: Double,  
 *val* activLoadKefGroup: Double,  
 *val* reactLoadKefGroup: Double,  
 *val* fullPower: Double,  
 *val* groupCurrent: Double,  
)

Клас калькулятору для обрахунків:

*class* EpCalculator {  
 *fun* findEmptyFields(epList: List<EpData>): List<EpData> {  
  
 epList.*forEach* **{  
 it**.nP = (**it**.countEp \* **it**.numPower).*toDouble*()  
  
 **it**.groupCurrent = (**it**.countEp \* **it**.numPower)/ *sqrt*(*3.0*) \* **it**.power \* **it**.cos \* **it**.kpd  
  
 **it**.npk = **it**.countEp \* **it**.numPower \* **it**.useKef  
  
 **it**.npktg = **it**.countEp \* **it**.numPower \* **it**.useKef \* **it**.reactKef  
  
 **it**.npPow = **it**.numPower.*toDouble*().*pow*(*2.0*) \* **it**.countEp  
 **}** *return* epList  
 }  
  
 *fun* calcShrData(epList: List<EpData>): ShrData {  
 *val* sumOfNp = epList.*sumOf* **{ it**.nP **}** *val* shrData: ShrData = *ShrData*(  
 countEp = epList.*sumOf* **{ it**.countEp **}**,  
 nPGroup = epList.*sumOf* **{ it**.nP.*toInt*() **}**,  
 useKefGroup = epList.*sumOf* **{ it**.npk **}** / epList.*sumOf* **{ it**.nP **}**,  
 npkGroup = epList.*sumOf* **{ it**.npk **}**,  
 npktgGroup = epList.*sumOf* **{ it**.npktg **}**,  
 npPow = epList.*sumOf* **{ it**.npPow **}**,  
 effectionCountity = sumOfNp.*pow*(*2*) / epList.*sumOf* **{ it**.npPow **}**,  
 activPowerKefGroup = *1.25*,  
 activLoadKefGroup = *1.25* \* epList.*sumOf* **{ it**.npk **}** ,  
 reactLoadKefGroup = *1.0* \* *107.302*,  
 fullPower = *sqrt*((*1.25* \* epList.*sumOf* **{ it**.npk **}**).*pow*(*2*) + (*1.0* \* *107.302*).*pow*(*2*)),  
 groupCurrent = (*1.25* \* epList.*sumOf* **{ it**.npk **}**) / *0.38* )  
  
  
 *return* shrData  
  
 }  
  
 *fun* fullOverload(epList: List<EpData>, shrData: ShrData, epHeavyList: List<EpData>): OverloadData {  
  
 *val* overload: OverloadData = *OverloadData*(  
 countEp = *81*,  
 nP = *2330*,  
 useKef = *752* / *2330.0*,  
 npk = *752.0*,  
 npktg = *657.0*,  
 npPow = *96399.0*,  
 effectionCountity = *2330*.*toDouble*().*pow*(*2*) / *96399.0*,  
 activPowerKefGroup = *0.7*,  
 activLoadKefGroup = *0.7* \* *752.0*,  
 reactLoadKefGroup = *0.7* \* *657.0*,  
 fullPower = *sqrt*((*0.7* \* *752.0*).*pow*(*2*) + (*0.7* \* *657.0*).*pow*(*2*)),  
 groupCurrent = (*0.7* \* *752.0*) / *0.38* )  
  
 *return* overload  
 }  
  
}

Результат роботи:



Результати сходяться з таблицею

Висновок: розроблено мобільний застосунок для розрахунку електричних навантажень об’єктів з використанням методу впорядкованих діаграм.