

Python OOP projektas

Užduotis. Surasti ir vizualizuoti nurodytus dydžius, taikant OOP. DATA/OOP_DATA aplanke yra pateikti DAT failai, reiklaingi šiai užduočiai.

Skaiciavimų aprašas (šiek tiek teorijos) . Pateiktuose DAT failuose yra matavimai (voltamperinė charakteristika, jV), charakterizuojantys saulės elementą - vienas DAT failas vienam saulės elementui (SE). Kiekvienas SE yra apibūdinamas 4-riais dydžiais:

- pce - naudingumo koeficientas, procentais. Parodo, kiek efektyvus yra SE.
- $j_{sc}, \left[\frac{mA}{cm^2} \right]$ - parodo, kiek maksimaliai gali sugeneruoti elektros srovės toks SE.
- U_{oc} , voltais - parodo, kokią maksimalią įtampą gali sugeneruoti SE.
- FF, procentais - parodo, kiek dar trūksta iki idealaus SE. Idealiai saulės elementui FF yra lygus 100 proc.

Kaip rasti šiuos dydžius:

Pateiktuose DAT failuose yra 4-ri stulpeliai, atskirti vienas nuo kito ;:

- U[V] įtampa, voltais;
- I[A] srovė;
- j[mA/cm²] srovės tankis;
- P[mW/cm²] galios tankis

Norėdami rasti pce - naudingumą, turite rasti minimalią 4-to stulpelio vertę (galios tankio minimumą), ir tada pce skaičiuojamas taip: $pce = \frac{P_{min}}{100} \cdot 100$. Gausite neigiamą dydį - paverskite teigiamu - pce visada turi būti teigiamas.

Norint rasti j_{sc} reikia daryti tokius veiksmus: surasti pirmajame stulpelyje (U[V] stulpelyje) vertę, artimą arba lygią 0, ir tada trečiajame stulpelyje (j[mA/cm²] stulpelyje), toje pat eilutėje bus ieškomas dydis j_{sc} .

Norint rasti U_{oc} , reikia atlikti tokius veiksmus: surasti trečiajame stulpelyje (j[mA/cm²] stulpelyje) vertę, lygią arba artimą 0, ir tada ieškomas dydis U_{oc} bus toje pat eilutėje, pirmajame stulpelyje.

Norint surasti FF, reikia pasinaudoti ankstesniais dydžiais:

$$FF = \frac{P_{min}}{j_{sc} \cdot U_{oc}} \cdot 100$$

FF irgi turibūti teigiamas dydis!

Nurodymai klasei, funkcionalumui, rezultatams

Klasės pavadinimas - `SolarAnalyzer`. Klasė turi būti aprašyta faile `manoFunkcijos.py` (šalia `pout()` funkcijos). Į Jupyter Notebook klasė importuojama taip:

```
from manoFunkcijos import SolarAnalyzer
```

Klasei inicializacijos metu (`__init__()` funkcijai) turi būti nurodoma tokia informacija - failo pavadinimas, stulpelių skirtukas. `__init__()` funkcija turi iškart nuskaityti pateiktą failą, išskaidyti duomenis į 4-ris sąrašus (kiekvieno stulpelio duomenys atskiram sąrašui).

Klasė turi turėti 5-kis atributus arba metodus, skirtus išgauti: Medžiagos pavadinimą, kuris yra kiekvino failo pradžioje iki `SE` simbolio, bei `SE` apibūdinantiems dydžiams - `pce`, `FF`, `jsc`, `Uoc`. Patys nuspręskite, ar tai bus metodai ar atributai, ir pagal tai susikonstruokite likusį klasės funkcionalumą.

Jūsų Jupyter Notebooke turi būti kodas, atliekantis tokius veiksmus: Suranda visus `DAT` failus, susikelia juos į sąrašą

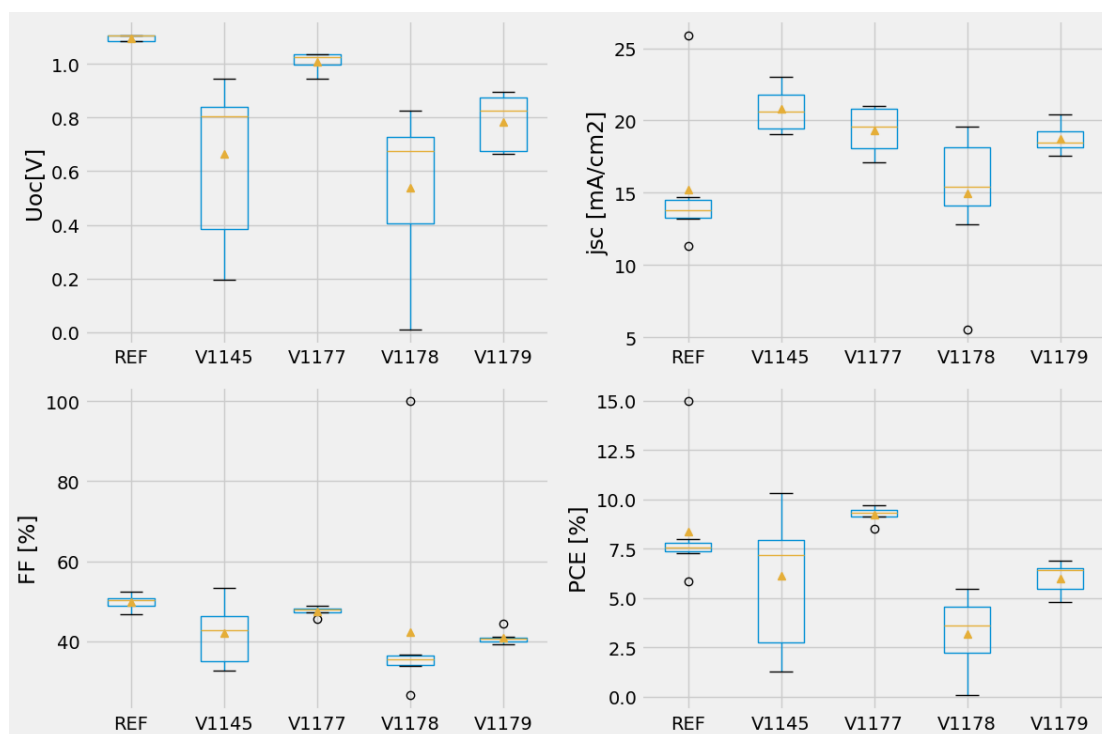
Tie failai paeiliui nuskaityti su `SolarAnalyzer()` klase.

Surandami prašomi 4-ri parametrai (`pde`, `FF`, `Uoc`, `jsc`) bei medžiagos pavadinimas.

Turite sugalvoti, kaip susirūšiuosite surastus parametrus pagal medžiagos pavadinimą.

Kai turėsite duomenis, susirūšiuotus pagal medžiagos pavadinimą, turite pasinaudoti `.boxplot()` komanda ir pateikti 4-ris grafikus, kurių pavyzdys pateiktas žemiau.

Pageidaujamas vizualizacijos rezultatas



1 pav.: Pageidaujamas rezultatas