

Documentation TP_SOLARX

Partie I : Test

1) Le code a été revu sur plusieurs plans

- On a réindented le code pour une meilleure lisibilité
On a utilisé l'outil Black.
- On a renommé les fonctions pour que cela soit le plus clair et compréhensible :

Exemple : Le paramètre `n` a été renommé `name`

`solar_panel_level_of_the_battery` a été renommé `battery_level`

2) On a utilisé Pylint pour se rassurer de bien avoir tout réorganisé

- Le premier test n'a pas été concluant

```
(.venv) PS C:\Users\scarf\Desktop\3CQT\TP_3CQT\projet_solarx> pylint .\solarx.py
***** Module solarx
solarx.py:1:0: C0114: Missing module docstring (missing-module-docstring)
solarx.py:5:0: C0115: Missing class docstring (missing-class-docstring)
solarx.py:14:4: C0116: Missing function or method docstring (missing-function-docstring)
solarx.py:22:4: C0116: Missing function or method docstring (missing-function-docstring)
solarx.py:25:4: C0116: Missing function or method docstring (missing-function-docstring)
solarx.py:28:4: C0116: Missing function or method docstring (missing-function-docstring)
solarx.py:29:8: R1705: Unnecessary "else" after "return", remove the "else" and de-indent the code inside it (no-else-return)
solarx.py:1:0: W0611: Unused import random (unused-import)
solarx.py:2:0: W0611: Unused import pytest (unused-import)
```

- On a revu les détails et le test Pylint suivant montre que tout est ok.

```
Your code has been rated at 10.00/10 (previous run: 6.25/10, +3.75)
```

```
(.venv) PS C:\Users\scarf\Desktop\3CQT\TP_3CQT\projet_solarx> pylint .\solarx.py
```

```
-----
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
```

3) On va ensuite réaliser des tests unitaires pour limiter les bugs et assurer la maintenabilité.

- Pour ce test on va utiliser le module `unittest` intégré à python.
`Import unittest`
- On effectue des tests sur chaque fonction et on essaie de tester toutes les éventualités.

- Sur l'image qui suit on remarque que le code a passé les tests mais pas à 100% sur toutes les fonctions.

```

Test Results 0 ms
└─ test_solarx 0 ms
    └─ TestSolarPanel 0 ms
        └─ test_check_maintenance 0 ms
        └─ test_perform_maintenance 0 ms
        └─ test_use_energy 0 ms

Tests passed: 3 of 3 tests - 0 ms

===== test session starts =====
collecting ... collected 3 items

test_solarx.py::TestSolarPanel::test_check_maintenance PASSED [ 33%]
test_solarx.py::TestSolarPanel::test_perform_maintenance PASSED [ 66%]
test_solarx.py::TestSolarPanel::test_use_energy PASSED [100%]

===== 3 passed in 0.01s =====

```

4) On vérifie que le code est bien sécurisé et n'a aucune faille grâce à bandit.

```

(.venv) PS C:\Users\scarf\Desktop\3CQT\TP_3CQT\projet_solarx> bandit -r .\solarx.py
[main] INFO profile include tests: None
[main] INFO profile exclude tests: None
[main] INFO cli include tests: None
[main] INFO cli exclude tests: None
[main] INFO running on Python 3.12.0
Run started:2024-02-28 01:43:01.454619

Test results:
    No issues identified.

Code scanned:
    Total lines of code: 43
    Total lines skipped (#nosec): 0

Run metrics:
    Total issues (by severity):
        Undefined: 0
        Low: 0
        Medium: 0
        High: 0
    Total issues (by confidence):
        Undefined: 0
        Low: 0
        Medium: 0
        High: 0

```

5) On va faire un Test Coverage

- On installe le package Coverage
pip install coverage
coverage run -m unittest .\solarx.py

```

(.venv) PS C:\Users\scarf\Desktop\3CQT\TP_3CQT\projet_solarx> coverage report -m
Name      Stmt  Miss  Cover  Missing
-----
solarx.py   15    10    33%   18-21, 28, 37, 49-52
-----
TOTAL       15    10    33%
(.venv) PS C:\Users\scarf\Desktop\3CQT\TP_3CQT\projet_solarx>

```

coverage html

Coverage report: 33%

coverage.py v7.4.3, created at 2024-02-28 02:52 +0100

<i>Module</i>	<i>statements</i>	<i>missing</i>	<i>excluded</i>	<i>coverage</i>
solarx.py	15	10	0	33%
Total	15	10	0	33%

coverage.py v7.4.3, created at 2024-02-28 02:52 +0100

Pour le moment le coverage n'est pas bon, on doit corriger cela.

Après vérification des Tests, le coverage est passé à 100%. Ci-dessous la commande utilisée.

`pytest --cov=solarx tests/ --cov-report=html`

Coverage report: 100%

coverage.py v7.4.3, created at 2024-03-03 00:18 +0100

<i>Module</i>	<i>statements</i>	<i>missing</i>	<i>excluded</i>	<i>coverage</i>
solarx.py	24	0	0	100%
Total	24	0	0	100%

coverage.py v7.4.3, created at 2024-03-03 00:18 +0100

Pour la suite de l'exercice,
ceci sera notre Menu.

