



Python Basic

Заняття 14

```
pip install dataclasses-json
```

```
from dataclasses import dataclass
from dataclasses_json import dataclass_json
```

```
@dataclass_json
```

```
@dataclass
```

```
class Person:
```

```
    name: str
```

```
person = Person(name='lidatong')
```

```
person.to_json()  # '{"name": "lidatong"}' <- this is a string
```

```
person.to_dict()  # {'name': 'lidatong'} <- this is a dict
```

```
Person.from_json('{"name": "lidatong"}')  # Person(1)
```

```
Person.from_dict({'name': 'lidatong'})  # Person(1)
```

```
# You can also apply _schema validation_ using an alternative API
```

```
# This can be useful for "typed" Python code
```

```
Person.from_json('{"name": 42}')  # This is ok. 42 is not a `str`, but
```

```
    # dataclass creation does not validate types
```

```
Person.schema().loads('{"name": 42}')  # Error! Raises `ValidationError`
```

SOLID — це аббревіатура, складена з перших літер п'яти базових принципів об'єктно-орієнтованого програмування та дизайну і запропонована Робертом Мартіном у статті 2000 року англ. Design Principles and Design Patterns.

S - Принцип єдиної відповідальності
(Single Responsibility Principle),

O – Принцип відкритості/закритості
(Open -Closed Principle),

L - Принцип підстановки Барбари Лисків
(Liskov Substitution Principle),

I – Принцип поділу інтерфейсів
(Interface Segregation Principle),

D – Принцип інверсії залежностей
(Dependency Inversion Principle).

- Принцип єдиної відповідальності

У класу має бути лише одна причина для зміни.

- Принцип відкритості/закритості

Сутність програми (класи, модулі, функції тощо) повинні бути відкриті для розширення, але закриті для змін.

- Принцип підстановки Барбари Лисків

Об'єкти у програмі мають бути замінені екземплярами їх підтипів без шкоди коректності роботи програми. Функції, які використовують посилання на базові класи, повинні мати можливість використовувати об'єкти похідних класів, не знаючи про це.

- Принцип поділу інтерфейсів

Жоден клієнт не залежати від методів, які він використовує. Іншими словами, якщо клас не використовує певних методів або атрибутів, тоді ці методи та атрибути слід розділити на більш конкретні класи

- Принцип інверсії залежностей

Модуль високого рівня повинен залежати від модулів низького рівня. І те, й інше має залежати від абстракцій.

Абстракції повинні залежати від деталей реалізації. Деталі реалізації мають залежати від абстракцій. Надання об'єкту речей, які йому потрібні, замість того, щоб він створював їх сам