



Python Advanced

Типізований Python. Модульне тестування



Python Advanced

Після уроку обов'язково



Повторіть цей урок у форматі відео на
[ITVDN.com](http://itvdn.com)



Перевірте, як Ви засвоїли цей матеріал на
[TestProvider.com](http://testprovider.com)

Типізований Python (модуль typing)

Python Advanced

Статична типізація

- Однозначність.
- Виключення помилок.
- Корисна у великих проєктах.
- Низька швидкість розробки.

Java

```
public class MyMath {  
    public static int maxElement(int a, int b) {  
        return (a > b) ? a : b;  
    }  
}
```

C++

```
int max_element (int a, int b)  
{  
    return (a > b) ? a : b;  
}
```

Python Advanced

Динамічна типізація

- Простота.
- Висока швидкість розробки.
- Універсальність.
- Помилки під час виконання.

JavaScript

```
function maxElement(a, b) {  
    return (a > b) ? a : b;  
}
```

Python

```
def max_element(a, b):  
    if a > b:  
        return a  
    return b
```

Python Advanced

Типізація в Python

- Підтримка типізації лише на рівні підказок (type hints).
- Модуль typing для створення складників типів.
- Реалізація перевірки типів сторонніх пакетів (mypy).
- Типізація у різних версіях Python.
- Підтримка типів в IDE.

```
def max_element(a: int, b: int) -> int:  
    if a > b:  
        return a  
    return b
```



Модульне тестування

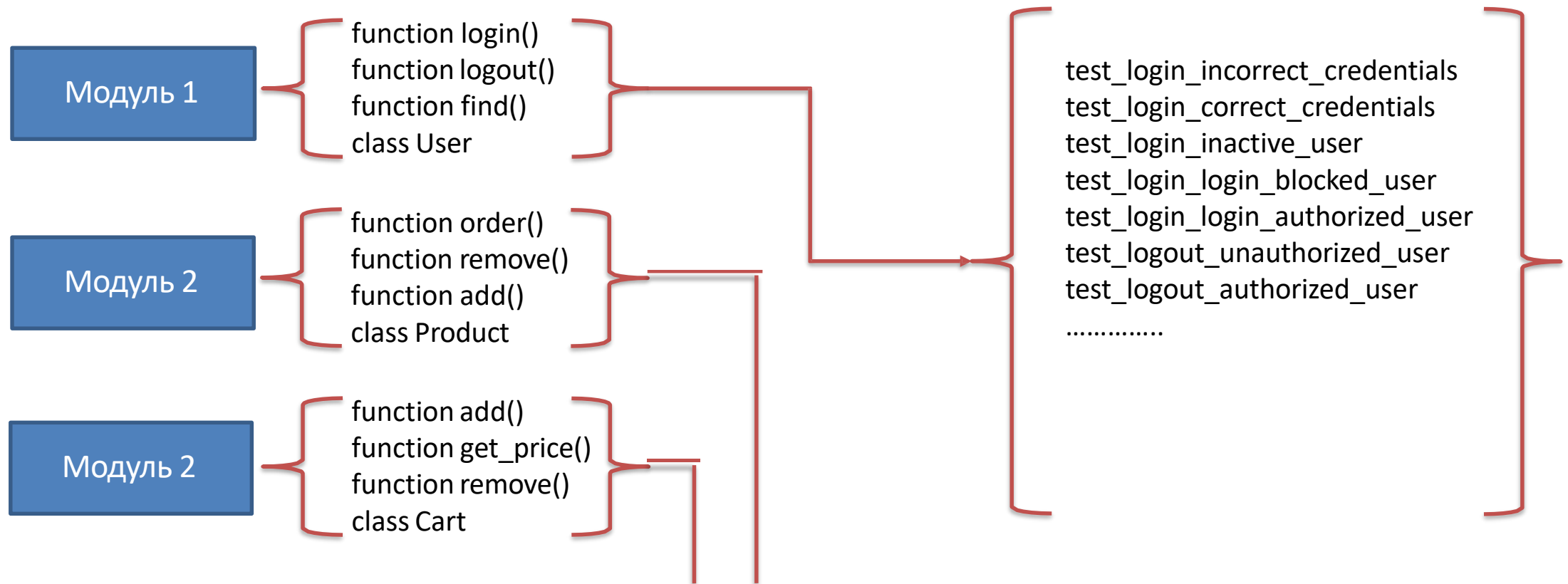
Модульне тестування

Основні поняття

- Різновиди тестування.
- Підтримка актуального стану тестів.
- Додаткові витрати часу.
- Код, який тестує код.
- Підхід «Розробка через тестування», або TDD.

Модульне тестування

Як працюють unit-тести



Модульне тестування

Як працюють unit-тести



Модульне тестування

Модульне тестування у Python

- Стандартний модуль unittest.
- Пакет pytest.
- Тестування в Django.
- Заглушки.



Python Advanced

Q&A

Інформаційний відеосервіс для розробників програмного забезпечення

