

Python Advanced

Типизированный Python. Модульное тестирование

Python Advanced

После урока обязательно



Повторите этот урок в видео формате на
[ITVDN.com](http://itvdn.com)



Проверьте как Вы усвоили данный материал
на [TestProvider.com](http://testprovider.com)

Типизированный Python (модуль typing)

Python Advanced

Статическая типизация

- Однозначность.
- Исключение ошибок.
- Полезна в больших проектах.
- Низкая скорость разработки.

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

Динамическая типизация

- Простота.
- Высокая скорость разработки.
- Универсальность.
- Ошибки во время выполнения.

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

Типизация в Python

- Поддержка типизации на уровне подсказок (type hints).
- Модуль typing для создания составных типов.
- Реализация проверки типов сторонними пакетам (mypy).
- Типизация на различных версиях Python.
- Поддержка типов в IDE.

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



Модульное тестирование

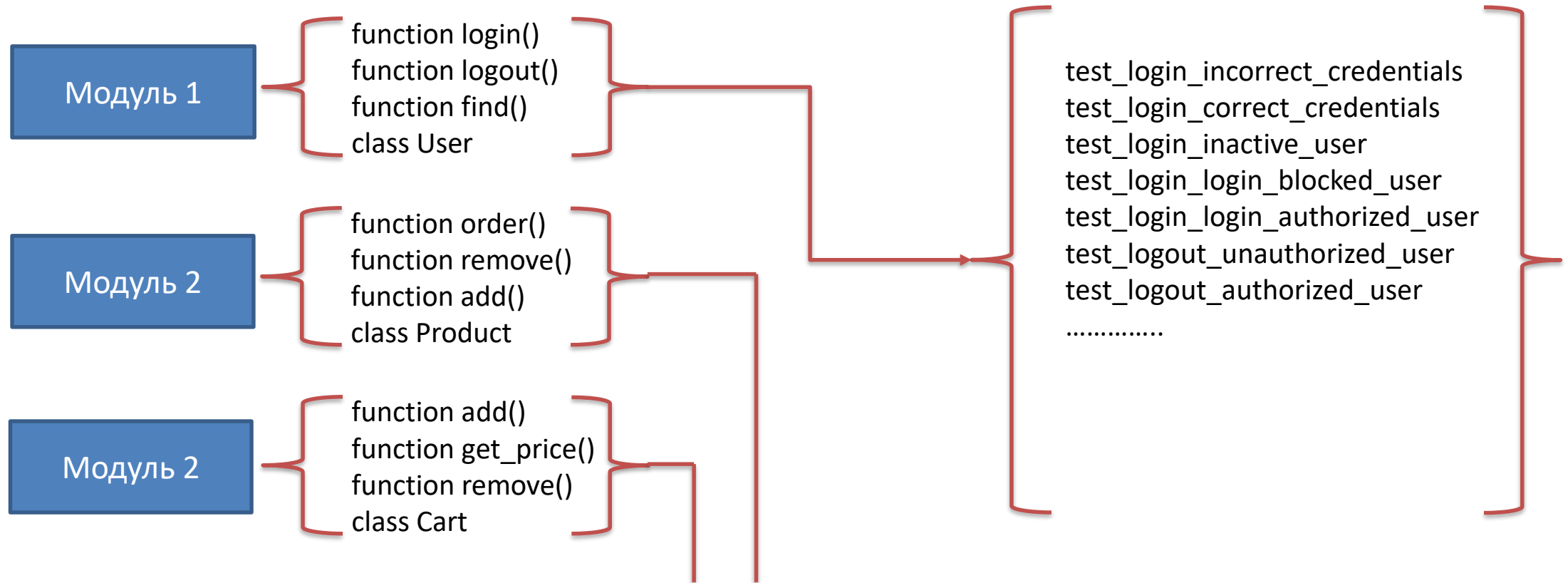
Модульное тестирование

Основные понятия

- Виды тестирования
- Поддержка актуального состояния тестов
- Дополнительные затраты времени
- Код который тестирует код
- Подход «Разработка через тестирование» или TDD

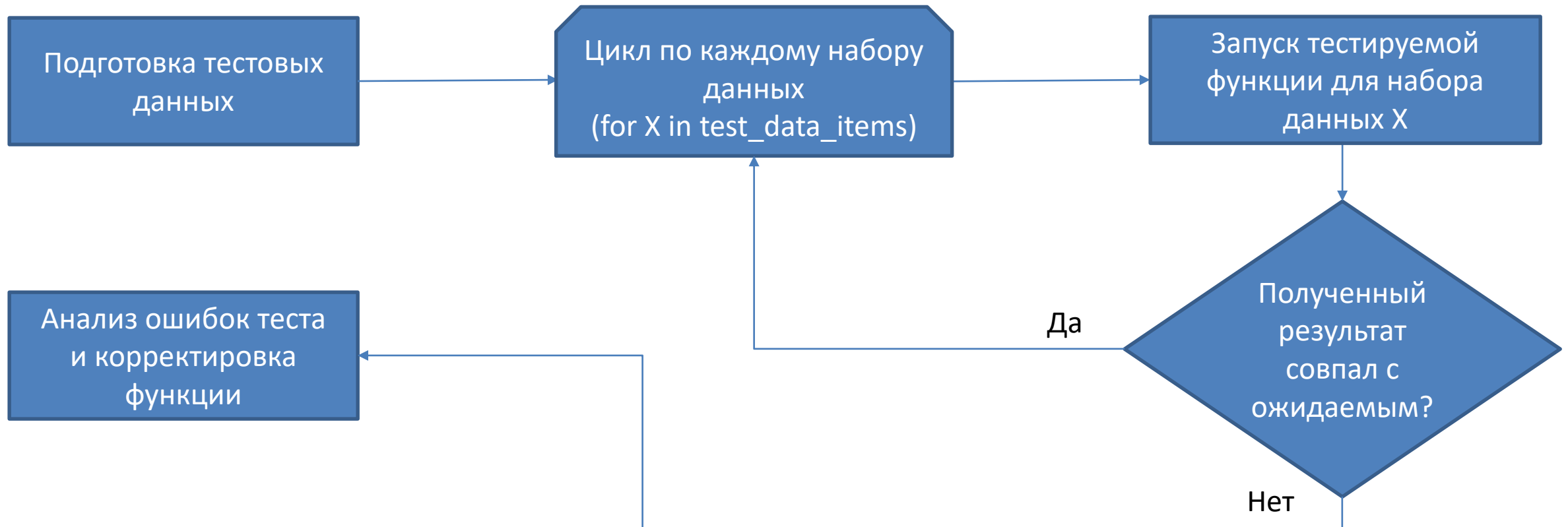
Модульное тестирование

Как работают unit-тесты



Модульное тестирование

Как работают unit-тесты



Модульное тестирование

Модульное тестирование в Python

- Стандартный модуль unittest
- Пакет pytest
- Тестирование в Django
- Заглушки



Python Advanced

Q&A

Информационный видеосервис для разработчиков программного обеспечения

