МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
ІМЕНІ ІГОРЯ СІКОРСЬКОГО

Теплоенергетичний факультет
Кафедра автоматизації проектування енергетичних процесів і систем

Дисертація Магістра
спеціальність 121 «Інженерія програмного забезпечення»

НА ТЕМУ: **«ПРОГРАМНІ ЗАСОБИ ПРИСКОРЕННЯ МОДУЛЬНОГО ТЕСТУВАННЯ»**

Магістр групи ТВз-71мп: Ігушкіна Т.С.
Науковий керівник: к.т.н., доцент Смаковський Д.С.

Київ – 2018

Рівні тестування

- **модульне** (unit testing) - найменші частини коду (класи, методи);
- **інтеграційне** (integration) - комбінація та взаємодія окремих модулів програми;
- **системне** (system) - система в цілому (БД, додатки, UI)
- **приймальне** (UAT) - відповідність до вимог

Модульне тестування

1. Чим більше коду покрито тестами,
   - тим стабільніший продукт;
   - менша вірогідність несправності;
   - легше рефакторити;
2. Модульні тести запускаються щоразу, коли вносяться зміни;
3. Чим старше/більше проект - тим більше модульних тестів накопичується

Актуальність

Модульне тестування, за досвідом практичного застосування, займає багато часу.

Тому актуальною є задача суттєво зменшити час на виконання модульного тестування.

Вирішенню саме цієї задачі присвячена дана робота.

Об’єкт дослідження - програмне забезпечення для модульного тестування;

Предмет - програмні засоби прискорення процесів модульного тестування;

Наукова новизна -

визначено, що послідовний запуск тестів не завжди є раціональним у використанні часових ресурсів. Тому *запропоновано* програмне рішення щодо процесів модульного тестування,
яке полягає у зменшенні часу на виконання модульних тестів завдяки паралельному їх виконанню у JUnit 5 за допомогою створення кастомної java-анотації (з можливістю відмітити всі або деякі тести кастомною анотацією) та розширенням функціоналу бібліотеки, який відсканує і запустить всі тести, позначені анотацією в багатопоточному режимі (одночасно).

JUnit

- бібліотека для модульного тестування програмного забезпечення на мові Java.
- де-факто стандарт модульного тестування у Java
- використовує механізм рефлексії для розпізнавання та запуску модульних тестів

Анотація

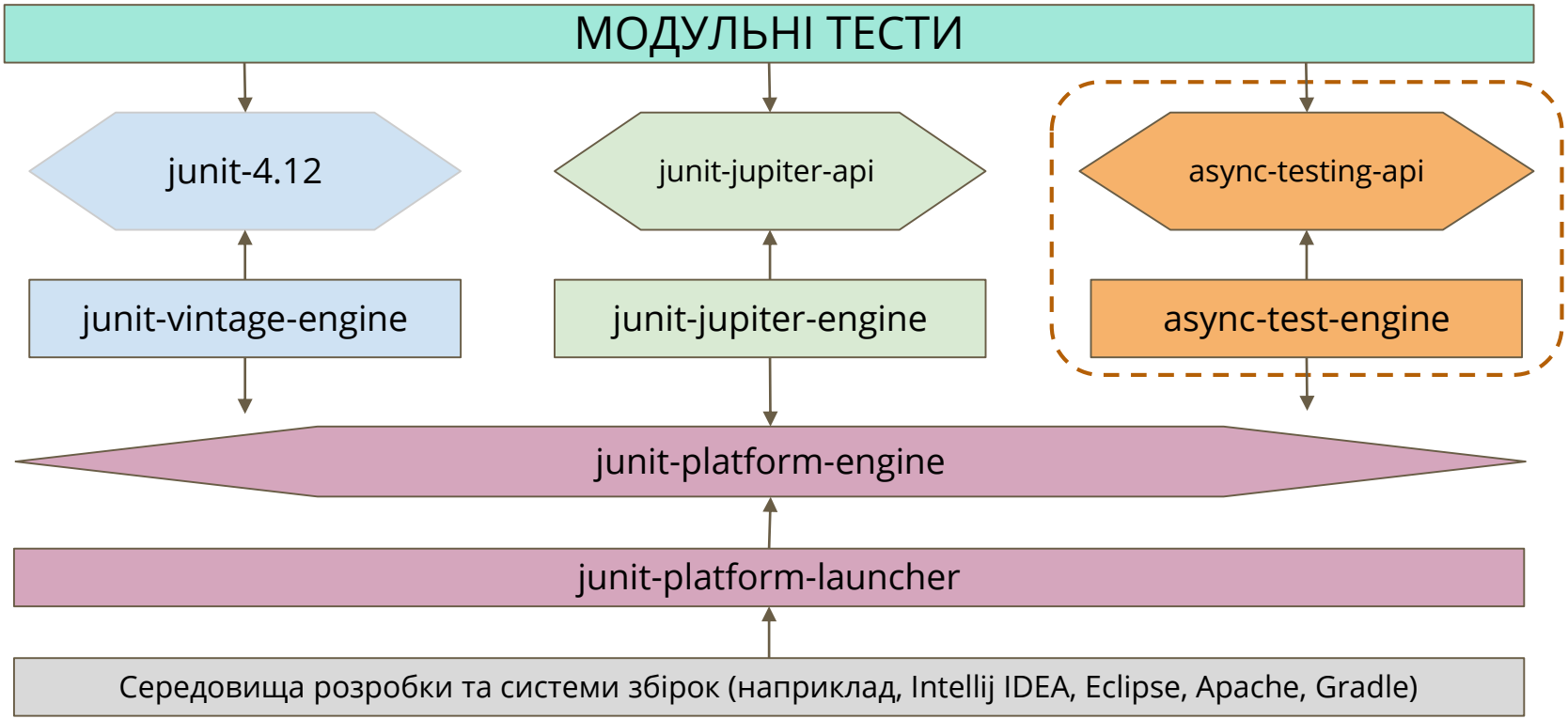
- спеціальна форма метаданих;
- тестові методи позначаються анотацією @Test;
- JUnit “розуміє”, що метод тестовий завдяки анотації.

Приклад тестового методу:

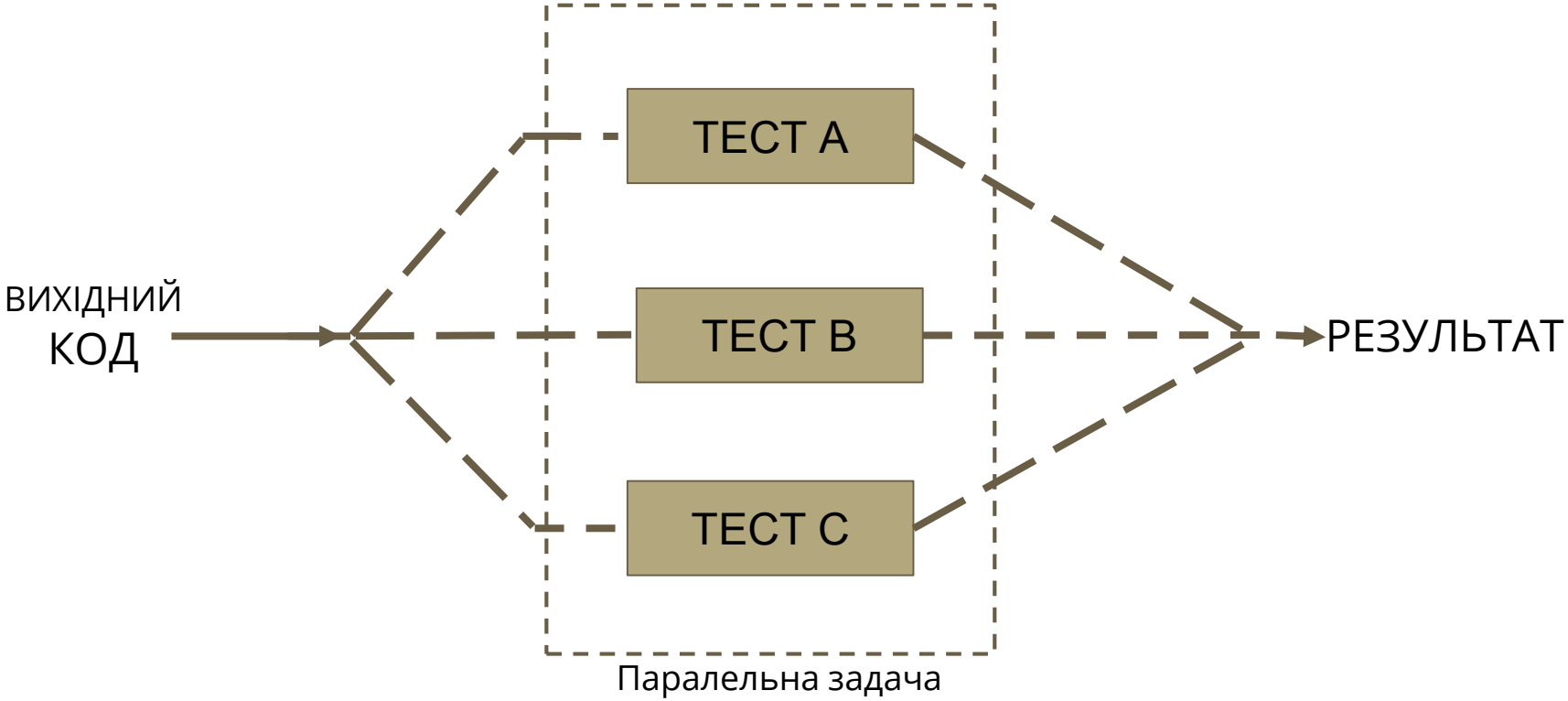
```
@Test
public void testMultiply() {
    MyClass tester = new MyClass();
    assertEquals("10 x 5 must be 50", 50, tester.multiply(10, 5));
}
```

Послідовне виконання тестів

JUnit 5, розширений за допомогою AsyncTestEngine



Організація паралельного тестування



Знімок екрану під час досліджень - послідовне виконання

Знімок екрану під час досліджень - паралельне виконання

Порівняння результатів

	JUnit5	JUnit5+AsyncTest
час на виконання 1222 тестів	26 сек 217 мсек	7 сек 988 мсек

Список технологій, що були використані при реалізації програмного застосунку

Висновки

Якість програмного продукту завжди була і є актуальною, а досягти цієї якості допомагають тести. Першими зазвичай виконуються модульні тести, які спроможні виключити багато дефектів на ранніх стадіях написання коду, тому важливо, щоб код був покритий тестами. Однак сучасні популярні інструменти не пропонують рішення, яке б допомогло запускати модульні тести у паралельному режимі.

Для того щоб виконання тестів було раціональним, ми можемо використати можливості Java, серед яких – багатопоточність. Таким чином, тести, з використанням запропонованого рішення, можна буде запускати одночасно, а також вибрати тести для паралельного запуску, позначивши їх анотацією. Це зробить роботу ефективнішою, використання ресурсів - раціональнішим.

ДЯКУЮ ЗА УВАГУ!