

УДК 004.43

Бабій В.– ст. гр. СІм-51

*Тернопільський національний технічний університет імені Івана Пулюя*

## **ОБ'ЄКТНО-ОРІЄНТОВАНА ПАРАДИГМА ПРОГРАМУВАННЯ. KOTLIN**

Науковий керівник: к.т.н., доцент Шингера Н.Я.

Babii V.

*Ternopil Ivan Pul'uj National Technical University*

## **THE OBJECT-ORIENTED PROGRAMMING PARADIGM. KOTLIN**

Supervisor: Shynhera N.

Ключові слова: об'єктно-орієнтоване програмування, Kotlin

Keywords: object-oriented programming, Kotlin

Об'єктно-орієнтоване програмування (ООП) – одна з парадигм програмування, яка розглядає програму як множину «об'єктів», що взаємодіють між собою. Основу ООП складають три основні концепції: інкапсуляція, успадкування та поліморфізм.

В наші дні в більшості випадків для написання складних програм, які мають вирішувати широке коло корпоративних завдань використовують об'єктно-орієнтований підхід програмування, оскільки в порівнянні із процедурним(функціональним) такий підхід має ряд переваг, описаних нижче.

1) Зменшення кількості помилок. За рахунок того, що можна розбити програму на велику кількість класів, у кожного класу можуть бути свої дані, до яких ніхто ніколи ззовні не зможе отримати доступу.

2) Прискорення налагодження програми. Більшість даних і методів строго прив'язані до свого класу й інші класи не мають до них доступу, пошук помилки, як правило, обмежується рамками лише одного класу, а не всієї програми.

3) Набагато прискорюється модифікація програми та нарощування функціоналу за рахунок написання нових класів.

Парадигму ООП на сьогоднішній день повністю підтримують такі мови, як Java, C#, C++, Python, Ruby та Objective-C, ActionScript 3, Swift, Kotlin.

Kotlin – об'єктно-орієнтована статично типізована мова програмування. Перший повний стабільний реліз публічно представлено 15 лютого 2016 року. Kotlin працює поверх Java Virtual Machine і має зворотну сумісність із Java але, в порівнянні з нею, володіє рядом переваг, серед яких:

1. Відсутність потреби під час оголошення змінної вказувати тип значення, яке вона буде містити. Тип буде визначений автоматично.
2. Відсутність необхідності закінчувати кожен оператор символом “;”.
3. Гнучкість оголошення методів.
4. Простіша і ефективніша перевірка на null, elvis-оператор.
5. Лаконічність та зручність циклу for.

Як підсумок, можна сказати, що Kotlin є досить перспективною мовою програмування. Враховуючи факт зворотної сумісності із Java, в майбутньому Kotlin цілком може стати її заміною при написанні кросплатформених додатків.