

СЕКЦІЯ «КОМП'ЮТЕРНІ СИСТЕМИ І МЕРЕЖІ»

УДК 004.4'2

Дьячук Т.С.¹

¹старш. викл. НУ «Запорізька політехніка»

ОПАНУВАННЯ СТУДЕНТАМИ СУЧАСНОГО ІНСТРУМЕНТАРІЮ РОЗРОБКИ ПРИ НАВЧАННІ ПРОГРАМУВАННЮ

Оволодіння сучасними засобами розробки допомагає студентам вирішувати складні завдання та стати конкурентоспроможними на ринку праці у майбутньому. Сучасні інструменти допомагають створювати програмне забезпечення ефективніше та швидше завдяки високому рівню автоматизації та інтеграції, вони підвищують якість коду через використання сучасних практик та стандартів. При вивченні будь-якої дисципліни з програмування, потрібно також обов'язково приділяти увагу опануванню сучасного інструментарію розробки, адже програмування не обмежується лише знанням синтаксису мови. Нерідко виникають ситуації, коли людина, здавалося б, знає мову програмування, але стає безпорадною при необхідності працювати над складними проектами в команді з використанням сучасних засобів розробки. Для успішної роботи в таких умовах необхідно не тільки володіти технічними знаннями, але й вміти ефективно взаємодіяти з іншими учасниками проекту та використовувати різноманітні інструменти для спільної роботи.

На рисунку 1 представлено стек технологій для сучасної розробки на мові Котлін. Розглянемо їх більш детально. Отже, Kotlin - статично типізована, об'єктно-орієнтована мова програмування, яка працює поверх JVM, може застосовуватися для створення застосунків у багатьох проблемних областях та є сучасною та високопродуктивною альтернативою Java.

Gradle - потужний інструмент автоматизації збірки програмного забезпечення, який дозволяє встановлювати та керувати залежностями проекту, виконувати збірку та розгортання, управляти бібліотеками. Gradle є популярним інструментом у сучасній розробці програмного забезпечення, особливо в області Java та Android.

IntelliJ IDEA - інтегрована середовище розробки, що підтримує багато мов програмування та забезпечує широкі можливості для комфортної розробки. Вона відома своєю високою продуктивністю та широким спектром функцій, що допомагають розробникам писати код швидше та ефективніше. Має розширену підтримку рефакторингу, автоматичного завершення коду, аналізу коду на льоту, інтеграцію з системами контролю версій, тестування та іншими корисними функціями. Ця IDE підтримує роботу з різними

фреймворками і технологіями, що робить її популярним вибором серед розробників усього світу.

GitHub - це веб-платформа для спільної розробки програмного забезпечення, яка базується на системі контролю версій Git. Вона забезпечує можливості для зберігання, спільної роботи над кодом, відстеження змін, обговорень та управління проєктами. GitHub є однією з найпопулярніших платформ для розробки програмного забезпечення у світі та використовується мільйонами програмістів. GitHub надає механізм Pull requests, який дозволяє розробникам запропонувати зміни до коду в репозиторії. Це спосіб ініціювати обговорення та перевірку внесених змін перед їх об'єднанням з основною гілкою проєкту. GitHub Actions - інструмент автоматизації робочих процесів, який дозволяє створювати та розгорнути сценарії для автоматичного виконання дій у вашому репозиторії GitHub. Наприклад, ви можете налаштувати автоматичну збірку проєкту або запуск автоматичних тестів при відправці коду. GitHub Desktop - графічний клієнт, який має зручний інтерфейс для роботи з Git та GitHub.

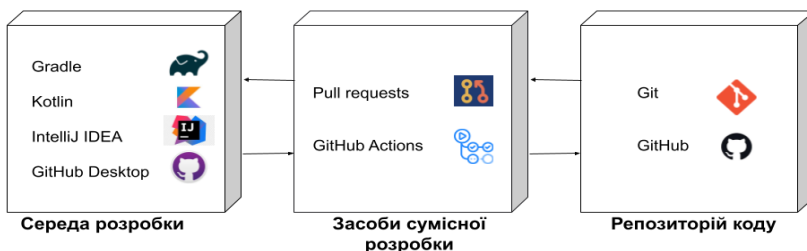


Рисунок 1 – Стек технологій для розробки на мові Kotlin.

Отже, всі ці засоби розробки було впроваджено та апробовано у рамках дисципліни «Основи програмування на Kotlin». Розроблено відповідні методичні вказівки та рекомендації. Для опанування та закріплення навичок спільної роботи, автоматичного тестування та розробки програмного забезпечення створено базовий репозиторій, доступний за посиланням <https://github.com/DiachT/KotlinLabsNUZP>. У цьому репозиторії є бібліотеки та початковий код, який можна використовувати для вивчення програмування та практичних завдань. Це допоможе здійснювати спільну роботу з іншими та вдосконалювати навички використання Git та GitHub.

Таким чином, хоча і виникають деякі труднощі при вивченні нових технологій, але набуті навички будуть дуже корисні в подальшому в професійній комерційній розробці, особливо навички спільної роботи у великих проєктах.