

Список використаних джерел

1. Universal Design for Learning (UDL) Principles for Teaching Online. URL: <https://teachwithgive.org/resource/remote-learning-applying-universal-design-for-learning-udl-principles> (дата звернення: 30.10.2023).
2. UDL для онлайнного навчання. URL: <https://www.kqed.org/mindshift/56205/how-designing-accessible-curriculum-for-all-can-help-make-online-learning-more-equitable> (дата звернення: 23.10.2023).
3. Israel M., Lash T., Ray M. Universal Design for Learning within Computer Science Education. Creative Technology Research Lab. University of Illinois (дата звернення: 30.10.2023).

АСПЕКТИ ПРАКТИЧНОГО ВИКОРИСТАННЯ ФРЕЙМВОРКУ GRADLE ДЛЯ ЗБИРАННЯ ПРОЄКТІВ

Галас Анатолій Віталійович

здобувач другого рівня вищої освіти за спеціальністю 014.09 Середня освіта (Інформатика)
Тернопільський національний педагогічний університет імені Володимира Гнатюка,
galas_av@fizmat.tnpu.edu.ua

Василенко Ярослав Пилипович

викладач кафедри інформатики та методики її навчання,
Тернопільський національний педагогічний університет імені Володимира Гнатюка,
yava@fizmat.tnpu.edu.ua

Тема практичного використання фреймворку Gradle є дуже актуальною для розробників програмного забезпечення і має багато аспектів, які варто розглянути. Ось деякі ключові аспекти та переваги практичного використання Gradle:

– Автоматизація збирання та тестування: Gradle дозволяє легко автоматизувати процес збирання проєкту та виконання тестів. Це робить розробку більш ефективною і допомагає уникнути багатьох помилок.

– Управління залежностями: Gradle має потужну систему управління залежностями, яка спрощує процес додавання, оновлення та видалення залежностей для вашого проєкту. Ви можете легко використовувати бібліотеки і компоненти зі світу відкритого програмного забезпечення.

– Гнучкість конфігурації: Gradle надає гнучкість для конфігурації вашого проєкту. Ви можете використовувати DSL (Domain-Specific Language) для опису проєкту та задач, що спрощує конфігурацію.

– Підтримка багатьох мов програмування: Gradle підтримує не лише Java, але і багато інших мов програмування, включаючи Kotlin, Groovy, C++, Python і інші. Це означає, що ви можете використовувати Gradle для проєктів, написаних на різних мовах.

– Робота з Android: Для розробки Android-додатків і бібліотек Gradle має спеціалізований плагін, який спрощує процес побудови та розгортання додатків на платформі Android.

– Плагіни та розширення: Ви можете використовувати готові плагіни або створювати свої власні для розширення функціональності Gradle та відповідності специфічним вимогам вашого проєкту.

– Спільнота та документація: Gradle має активну спільноту розробників і широку базу документації, що допомагає вирішувати проблеми та отримувати підтримку.

– Інтеграція з іншими інструментами: Gradle легко інтегрується з іншими інструментами розробки, такими як системи контролю версій, середовища розробки та CI/CD системи.

Функціональність Gradle включає в себе:

- Зручний DSL для налаштування проєкту, що дозволяє легко описувати задачі та залежності.
- Можливість автоматизувати процес збирання проєкту та тестування.
- Гнучкість та розширюваність, завдяки можливості додавання власних плагінів.
- Підтримка багатьох мов програмування та платформ.
- Можливість працювати з різними збірками проєкту, такими як JAR, WAR, EAR тощо.
- Інтеграція з іншими інструментами для розробки, такими як IDE та системи контролю версій.

Gradle може бути використаний для автоматизації процесів тестування, побудови та пакування додатків різного типу. Нижче розглянемо кожен з цих процесів більш детально.

1. Автоматизація тестування.

Gradle може бути використаний для автоматизації тестування, в тому числі для запуску тестів один за одним або паралельно, побудови звіту про результати тестування, налаштування середовища для тестування та іншого. Для цього можна використовувати спеціальні плагіни Gradle, наприклад, Test Gradle Plugin.

2. Побудова додатків.

Gradle може бути використаний для побудови додатків різного типу, включаючи Java, Kotlin, Android та інші. Для цього в Gradle використовуються так звані «задачі» (tasks), які відповідають за конкретні кроки в процесі побудови додатку. Наприклад, задача build відповідає за побудову додатку. За допомогою Gradle також можна підключати залежності та налаштовувати параметри побудови додатку.

3. Пакування додатків.

Gradle може бути використаний для пакування додатків в різні формати, включаючи JAR, WAR, APK та ін. Для цього використовуються спеціальні плагіни Gradle, наприклад, War Gradle Plugin або Android Gradle Plugin. Ці плагіни дозволяють пакувати додатки відповідно до вимог платформи, для якої вони призначені.

Gradle є досить популярним фреймворком для збирання проєктів, і багато відомих компаній використовують його для своїх розробок. Ось кілька прикладів відомих проєктів, які використовують Gradle:

1. Android: Google використовує Gradle для збирання Android-додатків. Інструмент Android Studio, який розробляється Google для розробки Android-додатків, інтегрує Gradle як систему збирання проєктів.

2. Spring Framework: Spring, один із найпопулярніших фреймворків для розробки додатків на Java, використовує Gradle для збирання своїх проєктів.

3. Netflix: Netflix використовує Gradle для збирання різних проєктів, включаючи внутрішні інструменти та бібліотеки.

4. LinkedIn: LinkedIn також використовує Gradle для збирання свого програмного забезпечення та для автоматизації різних завдань у проєктах.

5. Twitter: Twitter використовує Gradle для збирання та автоматизації внутрішніх проєктів.

6. Adobe: Adobe використовує Gradle для розробки і збирання свого програмного забезпечення.

7. Gradle: Команда Gradle, яка розробляє сам фреймворк, використовує Gradle для розробки та збирання фреймворку Gradle.

Ці приклади свідчать про широку популярність Gradle серед розробників і компаній з різних галузей, і його успішне використання для різних видів проєктів, від Android-додатків до веб-програмного забезпечення.

Висновки. Використання Gradle є актуальним і важливим для розробників, оскільки цей фреймворк допомагає автоматизувати багато аспектів процесу розробки, зробити його більш ефективним і забезпечити більшу стабільність проєктів. Фреймворк Gradle допомагає полегшити і прискорити розробку проєктів, зменшити витрати часу на рутинні завдання та підвищити якість програмного забезпечення.

Список використаних джерел

1. Василенко Я. П., Галас А. В. Технологічні особливості використання фреймворку Gradle для збирання проєктів. Сучасні цифрові технології та інноваційні методики навчання: досвід, тенденції, перспективи. Матеріали XI Міжнародної науково-практичної інтернет-конференції (м. Тернопіль, 6 квітня, 2023). С. 43–46.

2. Berglund T. and McCullough, M. Building and Testing with Gradle. O'Reilly Media, 2011. 110 p.

3. Kunal D. A. Gradle Essentials. Dabir Abhinandan Kunal. Packt Publishing Ltd, 2015. 176 p.

4. Основи життєвого циклу Gradle. Офіційна документація Gradle. URL: https://docs.gradle.org/current/userguide/build_lifecycle.html (дата звернення: 01.11.2023).

ІТ-СТУДІЇ: СУЧАСНІ ПІДХОДИ У ВИВЧЕННІ ІНФОРМАТИКИ

Джаган Ангеліна Валеріївна

здобувач другого рівня вищої освіти за спеціальністю 014.09 Середня освіта (Інформатика)
Тернопільський національний педагогічний університет імені Володимира Гнатюка,
dzhagan_av@fizmat.tnpu.edu.ua

Шмигер Галина Петрівна

кандидат біологічних наук, доцент кафедри інформатики та методики її навчання,
Тернопільський національний педагогічний університет імені Володимира Гнатюка,
shmyger@fizmat.tnpu.edu.ua

Визначення сучасних підходів: зокрема ІТ-студій та їх ролі в інформатиці. Один із новітніх підходів для вивчення інформатичної галузі у поєднанні з сучасними засобами навчання.

Важливим на сьогодні є підхід за допомогою якого вивчають інформатику в середній та старшій школі. Новий підхід оновленого курсу інформатики, яких включає низку різних завдань та вправ у поєднанні із теорією та практикою.

Навчальні програми змінилися, технології рухаються вперед та звісно ж і програми з інформатики не залишились осторонь. ІТ студії – це не лише про інформацію чи про комп'ютер в загальному але і про розвиток дитини – як