



Git-технологія командної розробки проєктів

Спеціальності: E2 Екологія, E4 Науки про Землю, F2 Інженерія програмного забезпечення, F3 Комп'ютерні науки, F4 Системний аналіз та наука про дані, F5 Кібербезпека та захист інформації, F6 Інформаційні системи і технології, F7 Комп'ютерна інженерія, G5 Електроніка, електронні комунікації, приладобудування та радіотехніка, G6 Інформаційно-вимірювальні технології, G7 Автоматизація, комп'ютерно-інтегровані технології та робототехніка, G18 Геодезія та землеустрій, G22 Біомедична інженерія, J6 Авіаційний транспорт

Рівень вищої освіти	другий (магістерський)
Статус дисципліни	вибіркова (Технічна дисципліна за вибором)
Обсяг дисципліни	90 годин / 3 кредити ЄКТС
Мова викладання	українська
Що буде вивчатися (предмет вивчення)	Застосування сучасних інструментальних засобів для аналізування стану і розроблення програмного забезпечення з використанням системи контролю версій Git та GitHub – веб-платформа для хостингу та управління версіями репозиторіїв Git. GitHub надає інструменти для спільної роботи над проєктами, відстеження змін у коді, управління завданнями та обговорення, а також автоматизації робочих процесів
Чому це цікаво/треба вивчати (мета)	Надання знань про використання Git-технологій у життєвому циклі розроблення програмного забезпечення. У контексті розробки програмного забезпечення GitHub дозволяє розробникам зберігати свої проєкти в репозиторіях, які можуть бути скопійовані (клонівані) на локальні машини розробників. Це дозволяє їм вносити зміни в код, створювати нові гілки, фіксувати зміни в комітах і надсилати їх на GitHub. Потім інші розробники можуть переглядати зміни, залишати коментарі та пропонувати свої зміни шляхом створення Pull Requests. Коли зміни перевірено та схвалено, вони можуть бути включені (злиття) в основну гілку проєкту. GitHub також пропонує додаткові функціональні можливості, такі як управління завданнями через GitHub Issues, ведення обговорень у GitHub Discussions, автоматизація процесів за допомогою GitHub Actions та публікація релізів проєкту за допомогою GitHub Releases. Це робить GitHub потужним інструментом для спільної розробки, спільного навчання та управління проєктами
Як можна користуватися набутими знаннями і уміннями (компетентності)	– Розробка програмного забезпечення: Git та GitHub широко використовуються розробниками для управління та контролю версій вихідного коду. Вони дозволяють ефективно працювати з командами розробників, керувати змінами, вирішувати конфлікти, створювати гілки для нових функціональностей чи виправлень, а також проводити код-рев'ю. – Колаборація та відкритий вихідний код: GitHub є популярною платформою для спільної роботи над проєктами з відкритим вихідним кодом. Він надає інструменти для вкладу в проєкти іншими розробниками, обговорення завдань, управління запитами на внесення змін та перегляду історії змін. Знання Git і GitHub дозволяє брати активну участь у спільнотах розробників і робити свій внесок у проєкти. – Управління проєктами: GitHub пропонує функціональності для управління проєктами, такі як створення завдань та відстеження прогресу через GitHub Issues, ведення дискусій та обговорень через GitHub Discussions, а також використання проєктних дошок для планування та управління

	завданнями. Знання Git та GitHub дозволяє ефективно організовувати та відстежувати роботу над проєктами. – Портфоліо та резюме розробника: Мати досвід роботи з Git та GitHub є цінним активом для розробників при пошуку роботи. Вони можуть використовувати GitHub для хостингу своїх проєктів, надання прикладів свого коду та демонстрації своїх навичок розробки. – Автоматизація робочих процесів: GitHub пропонує можливості автоматизації за допомогою GitHub Actions. Це дозволяє налаштовувати та запускати різні робочі процеси, такі як складання, тестування, розгортання та інші, на основі певних подій чи розкладу. Знання Git та GitHub дозволяє створювати та налаштовувати автоматичні робочі процеси для підвищення ефективності розробки. <i>В цілому, Git та GitHub широко застосовуються в різних сферах розробки програмного забезпечення та співпраці над проєктами. Вони допомагають упорядкувати роботу з кодом, керувати версіями, полегшити командну роботу та зробити процеси розробки більш прозорими та ефективними</i>		
Пререквізити	Бажано «Основи програмування» та «Об'єктно-орієнтоване програмування»		
Кореквізити	«Архітектура та проектування програмного забезпечення» та «Якість програмного забезпечення та тестування»		
Організація навчання	Види занять: лекції, практичні заняття. Форми здобуття освіти: денна, заочна, дистанційна, дуальна. Форми контролю: модульний контроль, залік		
Кафедра	Кафедра інженерії програмного забезпечення		
Факультет	Програмної інженерії та бізнесу		
Викладач		ПІБ	Кузнєцова Юлія Анатоліївна
		Посада	доцент
		Вчене звання	доцент
		Науковий ступінь	кандидат технічних наук
		e-mail	y.kuznetsova@khai.edu
		Персональна сторінка	https://se.khai.edu/kuznetsova/
Посилання на електронні матеріали курсу	https://mentor.khai.edu/course/view.php?id=8881		
Посилання на силабус			