

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

«Сучасні технології програмування»



Ступінь освіти	бакалавр
Освітня програма	Інформаційні системи та технології
Тривалість викладання	13, 14 чверті
- лекції:	1 година
- практичні заняття:	2 години
Мова викладання	українська залік

Сторінка курсу в СДО НТУ «ДП»: <https://do.nmu.org.ua/course/view.php?id=6678>

Кафедра, що викладає

Інформаційних технологій та комп'ютерної інженерії



Викладач:

Дереза Андрій Юрійович

доцент кафедри ІТКІ, канд. техн. наук

Персональна сторінка

https://it.nmu.org.ua/ua/HR_staff/prepods/DerezaAndrii.php

E-mail:

Dereza.A.Yu@nmu.one

1. Анотація до курсу

Сучасні технології програмування відіграють ключову роль у професійному розвитку студентів і розробників-початківців. Знання актуальних технологій та методологій допомагає ефективно розробляти програмне забезпечення та суттєво збільшує шанси на успішну кар'єру.

Даний курс знайомить із:

- підходами до проектування коду SOLID, DRY (Don't Repeat Yourself) та KISS (Keep It Simple, Stupid).
- підходами до організації та командної взаємодії, включаючи систему контролю версій Git, методологій розробки Scrum та Kanban.
- різними архітектурними підходами до розробки веб додатків, такими як MVC (Model-View-Controller), MVVM (Model-View-ViewModel), та MVP (Model-View-Presenter) для створення добре структурованих та масштабованих додатків.
- мікросервісною архітектурою, яка дозволяє створювати та розгортати додатки як набір незалежних сервісів.
- підходами неперервної інтеграції та неперервного розгортання (Continuous Integration/Continuous Deployment).

Студент повинен отримати навички роботи з системою контролю версій, включаючи створення репозиторія, локального розгортання та його оновлення. Навчитись створювати REST сервіси та мікросервісну архітектуру. Вміти встановлювати, наставляти та використовувати інструменти CI/CD, наприклад TeamCity.

2. Мета та завдання курсу

Мета дисципліни –дати студентам теоретичні знання та практичні навички у використанні сучасних технологій програмування, методологій розробки та архітектури веб-додатків. Курс має на меті підготовку студентів до ефективної роботи в реальних проектах, використовуючи передові підходи до проектування коду, організації командної взаємодії, створення гнучкої архітектури додатків та впровадження практик Continuous Integration/Continuous Deployment (CI/CD).

Завданнями курсу є отримання теоретичних знань з основ ефективного проектування коду, організації та командної взаємодії, створення гнучкої та масштабованої архітектури веб-додатків. На практиці навчитись розробляти REST сервіси та використовувати популярні існуючі інструменти для підвищення ефективності процесу розробки: систему контролю версій Git, інструмент опису API Swagger та систему CI/CD TeamCity

3. Результати навчання

- Знати та вміти застосовувати принципи SOLID, DRY та KISS для написання чистого та підтримуваного коду.
- Розуміти основні принципи гнучких методологій розробки.
- Вміти спроектувати масштабовану веб-систему на основі мікросервісної архітектури та REST API.
- Вміти користуватися системою контролю версій Git для керування кодом та спільної роботи.
- Вміти налаштувати процес CI/CD для автоматизації тестування та розгортання програми.

4. Структура курсу

ЛЕКЦІЇ

Лекція 1	Тема 1. Знайомство з курсом: мета та задачі дисципліни. Основні терміни та визначення Мета і задачі дисципліни; Що таке технологія програмування ? Які технології є затребуваними на ринку
Лекція 2	Тема 2. Підходи до проектування коду SOLID; DRY (Don't Repeat Yourself); KISS (Keep It Simple, Stupid) та інші
Лекція 3	Тема 3. Підходи до організації та командної взаємодії Agile, Scrum, Kanban
Лекція 4	Тема 4. Архітектурні підходи до розробки веб додатків MVC (Model-View-Controller); MVVM (Model-View-ViewModel); MVP (Model-View-Presenter); Вісім категорій архітектур веб додатків (Monolithic, SPA, PWA, SOA, Serverless та інші)
Лекція 5	Тема 5. Мікросервісна архітектура Дванадцять факторів розробки додатку; Порівняння з монолітом та SOA; Патерни та антипатерни
Лекція 6	Тема 6. REST (Representational State Transfer) Принципи, методи, формати даних та структури; Переваги та недоліки

ЛАБОРАТОРНІ ЗАНЯТТЯ

Лабораторна робота 1	Система контролю версій Git
Лабораторна робота 2	Інструмент опису API Swagger
Лабораторна робота 3	Інструмент CI\CD TeamCity
Лабораторна робота 4	REST API та мікросервісна архітектура

5. Технічне обладнання та/або програмне забезпечення

Активованій акаунт університетської пошти (student.i.p.@nmu.one) на Офіс365, використання дистанційної платформи (<https://do.nmu.org.ua/>).

Пакети приладних програм: .NET Core (або еквівалент), Operation system's terminal\command line, Git, TeamCity On-Premise Professional edition.

6. Система оцінювання та вимоги

6.1. Навчальні досягнення здобувачів вищої освіти за результатами вивчення курсу оцінюватимуться за шкалою, що наведена нижче:

Рейтингова шкала	Інституційна шкала
90 – 100	відмінно
75-89	добре
60-74	задовільно
0-59	незадовільно

6.2. Здобувачі вищої освіти можуть отримати **підсумкову оцінку** з навчальної дисципліни на підставі поточного оцінювання знань за умови, якщо набрана кількість балів з поточного тестування та самостійної роботи складатиме не менше 60 балів.

Максимальне оцінювання:

Теоретична частина	Практична частина		Бонус	Разом
	При своєчасному складанні	При несвоєчасному складанні		
65	30	20	5	100

Практичні роботи приймаються за контрольними запитаннями до кожної з роботи. Теоретична частина оцінюється за результатами здачі білету диференційного заліку, який містить 2 питання.

6.3. Критерії оцінювання підсумкової роботи

Робота повинна містити розгорнуті відповіді на два питання білету. Якщо робота виконується у дистанційному режимі, то видача номеру білета проходить через систему MS Teams у зазначеній викладачем групі спілкування. В такому режимі виконана робота пишеться вручну, фотографується та відсилається не електронну пошту викладача у впродовж встановленого викладачем часу. За виконану роботу нараховуються бали:

65 балів – дана розгорнута відповідь на два питання.

50 балів – дана розгорнута відповідь на одне питання, але є помилки при розгляді іншого питання, або є несуттєві помилки у відповідях на два питання.

30 балів – два повна відповідь на одне питання або на два питання зі значними помилками.

20 балів – відповідь на одне питання із значними помилками.

0 балів – відповіді на питання відсутні або повністю невірні, або робота здана несвоєчасно.

6.4. Критерії оцінювання практичної роботи

З кожної практичної роботи здобувач вищої освіти отримує запитання з переліку контрольних запитань до роботи. Кількість вірних відповідей визначають кількість отриманих балів.

7. Політика курсу

7.1. Політика щодо академічної доброчесності

Академічна доброчесність здобувачів вищої освіти є важливою умовою для опанування результатами навчання за дисципліною і отримання задовільної оцінки з поточного та підсумкового контролів. Академічна доброчесність базується на засудженні практик списування (виконання письмових робіт із залученням зовнішніх джерел інформації, крім дозволених для використання), плагіату (відтворення опублікованих текстів інших авторів без зазначення авторства), фабрикації (вигадування даних чи фактів, що використовуються в освітньому процесі). Політика щодо академічної доброчесності регламентується положенням "Положення про систему запобігання та виявлення плагіату у Національному технічному університеті "Дніпровська політехніка". http://www.nmu.org.ua/ua/content/activity/us_documents/System_of_prevention_and_detection_of_plagiarism.pdf. У разі порушення здобувачем вищої освіти академічної доброчесності (списування, плагіат, фабрикація), робота оцінюється незадовільно та має бути виконана повторно. При цьому викладач залишає за собою право змінити тему завдання.

7.2. Комунікаційна політика

Здобувачі вищої освіти повинні мати активовану університетську пошту.

Усі письмові запитання до викладачів стосовно курсу мають надсилатися на університетську електронну пошту.

7.3. Політика щодо перескладання

Роботи, які здаються із порушенням термінів без поважних причин оцінюються на нижчу оцінку. Перескладання відбувається із дозволу деканату за наявності поважних причин (наприклад, лікарняний).

7.4 Політика щодо оскарження оцінювання

Якщо здобувач вищої освіти не згоден з оцінюванням його знань він може опротестувати виставлену викладачем оцінку у встановленому порядку.

7.5. Відвідування занять

Для здобувачів вищої освіти денної форми відвідування занять є обов'язковим. Поважними причинами для неявки на заняття є хвороба, участь в університетських заходах, академічна мобільність, які необхідно підтверджувати документами. Про відсутність на занятті та причини відсутності здобувач вищої освіти має повідомити викладача або особисто, або через старосту.

За об'єктивних причин (наприклад, міжнародна мобільність) навчання може відбуватись в он-лайн формі за погодженням з керівником курсу.

7.6. Бонуси

Наприкінці вивчення курсу та перед початком сесії здобувача вищої освіти буде запропоновано заповнити електронні анкети (Microsoft Forms Office 365), які буде розіслано на ваші університетські поштові скриньки. Заповнення анкет є важливою складовою вашої навчальної активності, що дозволить оцінити дієвість застосованих методів викладання та врахувати ваші пропозиції стосовно покращення змісту навчальної дисципліни «Сучасні технології програмування». За участь у анкетуванні здобувач вищої освіти отримує **5 балів**.

8. Рекомендовані джерела інформації

Базові

1. Роберт Мартін «Чистий код: створення, аналіз, рефакторинг». –К.: Вид.Фабула, 2019. - 416с.

2. Кен Швабер та Джефф Сазерленд «Scrum guide», 2020р, 17с.

URL: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Ukrainian.pdf>

3. Orderly Disruption Limited, Daniel S. Vacanti «Kanban guide», 2021р, 9с.

URL: <https://kanbanguides.org/wp-content/uploads/2021/01/Kanban-Guide-2020-12.pdf>

4. Alex Parker, онлайн-курс «IBM: Microservices and Serverless»

URL: <https://www.edx.org/learn/computer-programming/ibm-microservices-and-serverless>

5. Martin Fowler «Microservice Architecture: Technology Diversity».

URL: <https://martinfowler.com/articles/microservice-trade-offs.html>