

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА
ПРИРОДОКОРИСТУВАННЯ**

Навчально-науковий інститут кібернетики, інформаційних технологій та інженерії

04-04-25S

СИЛАБУС

навчальної дисципліни

SYLLABUS

Інженерія програмного забезпечення		Software Engineering
Шифр за ОП	ОК 27	Code in Degree Programme
Освітній рівень: Бакалаврський (перший)		Level of Education: Bachelor's (first)
Галузь знань Інформаційні технології	12	Field of Knowledge Information Technology
Спеціальність Комп'ютерна інженерія	123	Field of Study Computer Engineering
Освітня програма: Комп'ютерна інженерія		Degree Programme: Computer Engineering

Силабус навчальної дисципліни *Інженерія програмного забезпечення* для здобувачів вищої освіти ступеня «бакалавр», які навчаються за освітньо-професійною програмою «Комп'ютерна інженерія», спеціальності «Комп'ютерна інженерія», 123. Рівне. НУВГП. 2024. 15 стор.

ОП на сайті університету: <https://ep3.nuwm.edu.ua/22990/>

Розробник силабусу: *Бойчура Михайло Володимирович, к.т.н., доцент кафедри обчислювальної техніки*

Силабус схвалений на засіданні кафедри обчислювальної техніки
Протокол № 14 від "14" червня 2024 року

В.о. завідувача кафедри: *Сидор А.І., к.т.н.*

Керівник (гарант) ОП: *Сидор А.І., к.т.н.*

Схвалено науково-методичною радою з якості ННІ КІТІ
Протокол № 7 від "17" червня 2024 року

Голова науково-методичної ради з якості ННІ: *Мартинюк П.М., д.т.н., професор.*


Попередня версія силабусу: <https://ep3.nuwm.edu.ua/26333>.

© Бойчура М.В., 2024
© НУВГП, 2024

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ	
Інженерія програмного забезпечення	
ЗАГАЛЬНА ІНФОРМАЦІЯ	
Ступінь вищої освіти	<i>Бакалавр</i>
Освітня програма	<i>Комп'ютерна інженерія</i>
Спеціальність	<i>123 Комп'ютерна інженерія</i>

Рік навчання, семестр	2-й рік, 1-й семестр, 1-й рік, 1-й семестр (для бакалаврів зі скороченим терміном навчання)
Кількість кредитів	5
Лекції:	24/4 години
Лабораторні заняття:	26/12 годин
Самостійна робота:	100/134 годин
Курсова робота:	Ні
Форма навчання	денна/заочна
Форма підсумкового контролю	Екзамен
Мови викладання	українська та англійська

ІНФОРМАЦІЯ ПРО РОЗРОБНИКА

Лектор	 <p>Бойчура Михайло Володимирович к.т.н., доцент кафедри обчислювальної техніки</p>
Вікіситет	http://wiki.nuwm.edu.ua/index.php/Бойчура Михайло Володимирович
ORCID	https://orcid.org/0000-0002-9073-4037
Канали комунікації	m.v.boichura@nuwm.edu.ua , Телеграм-група студентської групи

ІНФОРМАЦІЯ ПРО НАВЧАЛЬНУ ДИСЦИПЛІНУ

Мета та завдання

Відштовхуючись від результатів опитувань студентів щодо якості навчання та викладання і якості освітніх програм різних спеціальностей, існує досить великий попит на поглиблене опанування певної технології замість оглядового вивчення декількох окремих. Випускники ж звертають увагу на недостатній рівень здобутих протягом навчання м'яких навичок. Дана освітня компонента передбачає проходження повного циклу веб-розробки з використанням сучасних технологій та підходів: від створення проекту до його розгортання на веб-хостингу. У межах навчальної дисципліни студенти також матимуть можливість оволодіти тонкощами побудови порівняно великих і складних проектів у командах та розвинути низку м'яких навичок. В силабусі враховані результати навчання, отримані викладачем Бойчурою М.В. під час підвищення кваліфікації в ІТ-компанії SoftServe (курс .NET Advanced for Teachers, період: 04.01.2023–07.03.2023).

Метою навчальної дисципліни «Інженерія програмного забезпечення» є поглиблене оволодіння студентами сучасними та актуальними підходами до веб-розробки на базі фреймворку ASP.NET Core MVC.

Основним **завданням** освітньої компоненти є набуття знань, умінь та навичок (компетентностей), необхідних для проектування та програмування порівняно великих і складних командних веб-проектів на базі фреймворку ASP.NET Core MVC, відповідно до кваліфікації бакалавра з комп'ютерної інженерії.

Оновлення

У попередні роки на ОК Інженерія програмного забезпечення викладались **патерни проектування**. У 2024 році **змінено концепцію** предмету відповідно до силабусів інших навчальних закладів, бачення викладача, рекомендацій студентів та стейкхолдерів. Зокрема, базуючись на матеріалах курсу .NET Advanced for Teachers, що проводився ІТ-компанією SoftServe, в межах даної освітньої компоненти поглиблено викладатиметься командна веб-розробка на основі фреймворку **ASP.NET Core MVC**.

Окрім того, зміна концепції даного предмету зумовлена введенням ОК **Патерни проектування** на спеціальності 123 «Комп'ютерна інженерія» магістерського рівня вищої освіти в НУВГП.

Завдання до лабораторних робіт побудовані таким чином, щоб в подальшому розроблені проекти могли стати хорошою базою для написання кваліфікаційних бакалаврських робіт.

Посилання на розміщення освітнього компонента на навчальній платформі Moodle, на платформі освітніх програм та їхніх освітніх компонентів

<https://exam.nuwm.edu.ua/course/view.php?id=2720>

Передумови вивчення (місце освітнього компоненту в структурно-логічній схемі)

Для засвоєння даної освітньої компоненти у повній мірі необхідно отримати основні знання та навички з наступних освітніх компонент: Вступ до спеціальності (ОК 10), Дискретна математика (ОК 11), Алгоритми та методи обчислень (ОК 12); рекомендується володіти базовими навичками програмування та пройти практичну підготовку з програмування, що забезпечують ОК 13 та ОК 14, відповідно.

Отримані навички, серед іншого, частково можуть використовуватись при читанні освітніх компонент: Програмування мобільних пристроїв (ВБ 8.4), Системне програмування (ОК 25) та при написанні «Кваліфікаційна робота» (ОК 37).

Компетентності

Z1. Здатність до абстрактного мислення, аналізу і синтезу.

P6. Здатність проектувати, впроваджувати та обслуговувати комп'ютерні системи та мережі різного виду та призначення.

P7. Здатність використовувати та впроваджувати нові технології, включаючи технології розумних, мобільних, зелених і безпечних обчислень, брати участь в модернізації та реконструкції комп'ютерних систем та мереж, різноманітних вбудованих і розподілених додатків, зокрема з метою підвищення їх ефективності.

P8. Готовність брати участь у роботах з впровадження комп'ютерних систем та мереж, введення їх до експлуатації на об'єктах різного призначення.

P15. Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення.

Програмні результати навчання (ПРН). Результати навчання (РН)*

N3. Знати новітні технології в галузі комп'ютерної інженерії.

N9. Вміти застосовувати знання технічних характеристик, конструктивних особливостей, призначення і правил експлуатації програмно-технічних засобів комп'ютерних систем та мереж для вирішення технічних задач спеціальності.

N10. Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати, типове для спеціальності обладнання.

N12. Вміти ефективно працювати як індивідуально, так і у складі команди.

N23. Здатність адаптуватись до нових ситуацій, обґрунтовувати, приймати та реалізовувати у межах компетенції рішення.

N25. Якісно виконувати роботу та досягати поставленої мети з дотриманням вимог професійної етики.

СТРУКТУРА ТА ЗМІСТ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

№	Тема	Опис лекції	Опис лабораторного заняття
МОДУЛЬ 1. Змістовий модуль 1. Основи розробки веб-сайтів на базі фреймворку ASP.NET Core MVC			
1	Вступ. Огляд шаблону ASP.NET Core MVC. 2 год. лекцій 2 год. лабораторних РН8, РН9, РН14	Огляд платформи .NET. Переваги та недоліки ASP.NET Core. Структура проекту. Вступ до патерну MVC. Фреймворк Bootstrap.	Встановлення необхідних компонентів. Створення першого проекту. Знайомство з файлами _Layout.cshtml, Index.cshtml, HomeController.cs. Класи Bootstrap для роботи з кнопками, картками та текстовими полями.
2	Патерн MVC. 2 год. лекцій 2 год. лабораторних РН8, РН9, РН14	Особливості роботи з представленнями. Робота з класами. Створення моделі. Властивості. Модифікатори доступу. Створення контролера. Дії контролера. Взаємодія моделі, представлення та контролера.	Створення нового контролера. Передача чисел та рядків у представлення. Створення моделі. Передача моделі-списку у представлення та у зворотній бік.

3	Командна розробка веб-застосунків. 2 год. лекцій 2 год. лабораторних PH8, PH9, PH14	Особливості роботи у командах. Гра PlanningPoker. Маніфест Agile. Підходи Waterfall та Scrum. GitHub. Pull requests. Branches. Jira. Story Points. Task, Story, Bug. Графіки. Створення організації. Синхронізація GitHub та Jira.	Формування студентських команд. Командна розробка тестового проекту. Пошук та інтеграція веб-сайтів у проект. Робота з GitHub Desktop. Налаштування GitHub для командної розробки. Створення вітки develop. Створення віток виду id/task/description. Формування Pull request. Робота з Jira. Налаштування Jira. Задання тестових задач. Встановлення Story Points. Створення організації. Синхронізація GitHub та Jira.
4	CRUD операції. 2 год. лекцій 2 год. лабораторних PH8, PH9, PH14	Типи запитів. Клас DbContext. Створення таблиць бази даних. Навігаційні властивості. Ініціалізація таблиць даними. Зчитування даних з бази даних. Додавання рядка у базу даних. Оновлення даних у рядку бази даних. Видалення рядка бази даних. Мова LINQ. Фреймворк Entity Framework. Інструмент Migration.	Microsoft SQL Server 2019. Створення схеми даних (3 пов'язані таблиці). Встановлення навігаційних властивостей. Міграція. Ініціалізація таблиць даними. Додавання, зчитування, редагування та видалення даних. Фільтрація даних. Представлення скетчів проектів. Вибір TeamLeader'a.
5	Пошарова архітектура. Сервіси. 2 год. лекцій 2 год. лабораторних PH8, PH9, PH14	Відділення роботи з базою даних від UI. Відділення бізнес-логіки від UI. Створення сервісу. Дизайн-патерн Dependency Injection. Взаємодія декількох проектів.	Розділення єдиного проекту на 3: BusinessLogic, DataAccess, UI. Створення сервісів у BusinessLogic. Додавання сервісів засобами Dependency Injection. Встановлення необхідних пакетів. Представлення ступеня виконання проекту.
МОДУЛЬ 2.			
Змістовий модуль 2. Сучасні технології та підходи до розробки веб-сайтів на базі фреймворку ASP.NET Core MVC			
6	Asynchronous programming. Roles. 2 год. лекцій 2 год. лабораторних PH8, PH9, PH14	Стандартні та користувацькі асинхронні методи. Метод Task.Run(). Система Identity. Створення ролей. Призначення ролей. Ініціалізація ролей. Методи розширення. Ключове слово static. Атрибут Authorize.	Перехід від синхронності до асинхронності. Створення Story в Jira. Введення ролей. Призначення ролей. Розподіл функціоналу між ролями. Представлення ступеня виконання проекту.
7	Scaffolded items. 2 год. лекцій 2 год. лабораторних PH8, PH9, PH14	Система Identity. Адаптація Scaffolded items під потреби проекту. Архітектура Razor Page	Додавання тих Scaffolded items, які потребуються в межах проекту. Редагування Razor Pages під власні потреби. Налаштування параметрів автентифікації. Пов'язування Identity з електронною поштою. Представлення ступеня виконання проекту.

8	Repository Pattern. Clean Architecture. 2 год. лекцій 2 год. лабораторних РН8, РН9, РН14	Суть патерну Repository. Інтеграція патерну Repository. Generic Types. Суть Clean Architecture. Інтеграція Clean Architecture у проект. Підключення необхідних пакетів. Модифікатори доступу.	Інтеграція патерну Repository у проект DataAccess. Generic Types. Асинхронність. Перейменування проектів у відповідності до концепції Clean Architecture. Налаштування зв'язків та модифікаторів доступу. Підключення необхідних пакетів. Налаштування Dependency Injection. Представлення ступеня виконання проекту.
9	Specifications. DTO. Mapping. 2 год. лекцій 2 год. лабораторних РН8, РН9, РН14	Випадки використання специфікацій. Lambda-вирази. Data Type Objects. Mapping об'єктів та списків об'єктів.	Інтеграція специфікацій у проект. Формування складних запитів до бази даних. Обмін даними між проектами на основі DTO. Mapping між моделями та їх DTO-аналогами. Mapping у випадку списків. Представлення ступеня виконання проекту.
10	Sessions. Serialization. Partial View. Web-site deployment. 2 год. лекцій 2 год. лабораторних РН8, РН9, РН14	Випадки використання сесій. Серіалізація. Структуризація засобами часткових представлень. Перенесення бази даних у глобальну мережу. Розгортання веб-сайту. Оптимізація кількості файлів веб-сайту.	Інтеграція сесій у веб-сайт. Застосування серіалізації. Використання часткових представлень. Перенесення бази даних у глобальну мережу. Розгортання веб-сайту. Оптимізація кількості файлів веб-сайту. Представлення ступеня виконання проекту.
11	Validation. Pagination. 2 год. лекцій 2 год. лабораторних РН8, РН9, РН14	Валідація на стороні сервера. Валідація на стороні клієнта. Атрибути валідації. Користувацькі атрибути. Tag-хелпери. Реалізація пагінації.	Забезпечення коректної валідації усіх даних. Введення пагінації даних. Представлення ступеня виконання проекту.
12	Підведення підсумків та подальші перспективи. 2 год. лекцій 2 год. лабораторних РН6, РН8	Етапи проведення співбесід у контексті фреймворку ASP.NET Core MVC. Формування резюме на основі пройденого матеріалу з даного предмету. Організаційні моменти представлення проектів. Принципи SOLID, Big Design Up Front, DRY, KISS тощо. Рекомендації Coding Conventions та відлагодження.	Узагальнення набутих навичок у вигляді резюме. Проведення тренувальних співбесід. Представлення ступеня виконання проекту.
13	Захист проектів. 2 год. лабораторних	–	Командне представлення проектів.

Форми, методи та технології навчання

Форми навчання	<ul style="list-style-type: none"> • очна (денна) з елементами дистанційного навчання; • заочна.
Форми навчального процесу	<ul style="list-style-type: none"> • навчальні заняття (лекції, лабораторні заняття, консультації); • самостійна робота здобувачів; • робота в наукових бібліотеках та мережі Інтернет; • контрольні заходи (поточна складова оцінювання, модульні контролі, підсумковий контроль).
Методи та технології навчання	<ul style="list-style-type: none"> • робота в малих групах (команді) та індивідуальна робота; • проектна технологія; • аналіз конкретних ситуацій (case study): ситуація-оцінка; • контекстне навчання; • проблемне навчання.
Процес навчання включає, зокрема, наступне	<ul style="list-style-type: none"> • написання комп'ютерних програм; • відлагодження програм; • Code Review; • слідування рекомендаціям Coding Conventions та принципу Big Design Up Front.
Засоби навчання	<ul style="list-style-type: none"> • відеозапис лекції; • презентація; • навчальний посібник; • конспект лекцій; • різні тьюторіали.

Інструменти, обладнання, програмне забезпечення

- середовище розробки: Microsoft Visual Studio 2022 (з компонентами для розробки веб-застосунків із застосуванням ASP.NET та SQL Server);
- Git термінал: GitHub Desktop;
- система керування проектами Jira;
- веб-хостинг somee.com.

Порядок оцінювання програмних результатів навчання/результатів навчання

Студент може отримати 100 балів, враховуючи наступну розбаловку:

1. модульні контролі: 40 балів;
2. поточний контроль: 26-36 бали;
3. командне довгострокове завдання: 24 балів;
4. додаткові бали: 0-10 балів.

Розподіл балів:

1. за модульні контрольні роботи:

- модульний контроль №1 (20 балів):
 - Рівень 1 – 19 запитань по 0.5 балів за кожне.
 - Рівень 2 – 6 запитань по 0.9 балів за кожне.
 - Рівень 3 – 3 запитання по 1.7 балів за кожне.
- модульний контроль №2 (20 балів):
 - Рівень 1 – 19 запитань по 0.5 балів за кожне.
 - Рівень 2 – 6 запитань по 0.9 балів за кожне.
 - Рівень 3 – 3 запитання по 1.7 балів за кожне.

2. за поточний контроль (26-36 бали):

Передбачено по 3 бали за перші 12 лабораторних робіт; у випадку правильного виконання відповідних завдань оцінка лінійно залежить від відсотка розуміння коду. За певних умов можливе повне перенесення Story Points, вказаних у Jira, в електронний журнал. У якості альтернативи студенти можуть виконувати командні проекти з використанням інших мов чи технологій (відмінних від ASP.NET Core MVC), але за умови попереднього узгодження деталей з викладачем.

3. командне довгострокове завдання (24 балів):

На останньому лабораторному занятті передбачається захист виконаних командних проектів у форматі відкритої пари. Для отримання максимальної оцінки студенти повинні якісно підготуватись до представлення проекту, реалізувати низку актуального функціоналу (патернів, технологій, архітектурних підходів тощо), показати наповненість Jira і GitHub, розподілити між собою ролі при виступі, обґрунтувати актуальність проекту, використовувати англійську мову, у випадку дистанційного представлення повинні включати відеокамеру; оцінюється також ступінь завершеності проекту, командна робота тощо. Викладач кожному члену команди задаватиме запитання щодо розуміння програмного коду.

4. додаткові бали за вагому громадську та студентську активності (0-10 балів):

Виставляється до 10 балів за волонтерство, олімпіади, спартакіади, конкурси, конференції, написання статей, активну студентську діяльність, конкретні пропозиції з удосконалення змісту освітньої компоненти тощо.

Поєднання навчання та досліджень

Кожен студент навіть для отримання 60 балів повинен розробити досить серйозний та великий проект. Передбачається, що він потребуватиме попереднього аналізу предметної області, дослідження актуальності, постановки завдань дослідження, формулювання об'єкта та предмета, розробки алгоритму тощо. Таким чином, розроблені студентами проекти можуть бути придатними для публікування результатів.

Вивчення даної освітньої компоненти може бути хорошим фундаментом для написання кваліфікаційної бакалаврської роботи.

Рекомендована література (основна, допоміжна)

Основна література

1. Lock A. *ASP.NET Core in Action*. 3rd ed. Shelter Island: Manning, 2023. 984 p.
2. Freeman A. *Pro ASP.NET Core 7*. 10th ed. Shelter Island: Manning, 2023. 1256 p.
3. Danylko J.R. *ASP.NET 8 Best Practices: Explore techniques, patterns, and practices to develop effective large-scale .NET web apps*. Birmingham: Packt Publishing, 2023. 256 p.
4. Smith J.P. *Entity Framework Core in Action*. 2nd ed. Shelter Island: Manning, 2021. 624 p.
5. Мартін Р. Чистий Agile. Назад до основ. Харків: Фабула, 2021. 224 с.
6. Омельчук Л.Л., Русіна Н.Г. Інструментальні середовища та технології програмування. Лабораторний практикум. Навчальний посібник Одеса: Айс Принт, 2020. 175 с.
7. Мартін Р. Чистий код. Створення, аналіз і рефакторинг. Харків: Фабула, 2019. 368 с.
8. Troelsen A., Japikse P. *Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming*. 11th ed. New York: Apress, 2022. 1705 p.
9. Richter J. *CLR via C#*. 4th ed. Redmond: Microsoft Press, 2012. 1594 p.

Допоміжна література

1. Коноваленко І.В. Платформа .NET та мова програмування C# 8.0: навчальний посібник. Тернопіль: ФОП Паляниця В. А., 2020. 320 с.
2. Freeman A. *Pro ASP.NET Core 6: Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages*. 9th ed. New York: Apress, 2022. 1286 p.
3. Marcotte C.-H. *An Atypical ASP.NET Core 6 Design Patterns Guide*. 2nd ed. Birmingham: Packt Publishing, 2022. 678 p.
4. Shellman M., Afyouni H., Pratt P.J., Last M.Z. *A Guide to SQL*. 10th ed. Kentucky: Cengage Learning, 2020. 320 p.
5. Bootstrap · The most popular HTML, CSS, and JS library in the world. URL: <https://getbootstrap.com/> (Last accessed: 18.11.2023).
6. ASP.NET Core | Open-source web framework for .NET. URL: <https://dotnet.microsoft.com/en-us/apps/aspnet> (Last accessed: 18.11.2023).
7. Introduction to Identity on ASP.NET Core | Microsoft Learn. URL: <https://learn.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-8.0&tabs=visual-studio> (Last accessed: 18.11.2023).
8. Manifesto for Agile Software Development. URL: <https://agilemanifesto.org/> (Last access: 14.01.2023).

Інформаційні ресурси в Інтернеті

1. <https://www.youtube.com/@mvboichura> (YouTube канал викладача даного предмету)
2. <https://www.youtube.com/watch?v=phyV-OQNeRM&list=PLLWMQd6PeGY1C3YLCyAUIDJrPS9S3WdLK&index=11> (Introduction to ASP.NET MVC in C#)
3. <https://www.youtube.com/watch?v=h7alzCkmbI8&list=PLLWMQd6PeGY2GVsQZ-u3DPXqwwKW8MkiP> (An Introduction To The C# Learning Cycle)
4. <https://www.youtube.com/watch?v=lkmvnjypENw> (Clean Architecture with ASP.NET Core 6)
5. <https://www.youtube.com/playlist?list=PLqWD37Mj2te0xiDW-Q58ZIG7XZry-1-bk> (Jira Tools)

Перелік соціальних, «м'яких» навичок (soft skills)

<p><i>Вміння комунікувати.</i></p>	<ul style="list-style-type: none"> • здатність спілкуватися державною мовою як усно, так і письмово; • вміння спілкуватись та писати із використанням англomовної професійної термінології; • навички усного спілкування; • навички письмового спілкування; • вміння писати зрозумілий код.
<p><i>Вміння сумісно працювати.</i></p>	<ul style="list-style-type: none"> • навички управління проектами; • здатність планувати свій час у сенсі співставлення вимог, власних знань, здібностей і дедлайнів; • здатність працювати в команді; • навички міжособистісних відносин; • вміння надавати рекомендації іншим у коректній формі.
<p><i>Здатність до аналізу та синтезу.</i></p>	<ul style="list-style-type: none"> • вміння критично мислити; • здатність знаходити вихід із складних ситуацій; • здатність до навчання; • вміння комплексного вирішення проблем.
<p><i>Здатність застосовувати знання у практичних ситуаціях.</i></p>	

Дедлайни та перескладання

Дедлайн здачі лабораторних робіт – останній четвер перед початком сесії. Здача лабораторних робіт відбувається на парі або під час консультації, дата та час якої гнучко узгоджується між студентом та викладачем.

На здачу кожного з модульних контролів студенту надається одна спроба. Перший модуль здається на будь-якій лекції у жовтні, а другий – на передостанній чи останній парі. Перездача окремого модульного контролю передбачена лише за виключних обставин.

У випадку, якщо студент набрав хоча б 60 балів, то він має право отримати дану підсумкову оцінку «автоматом»; в усіх інших випадках студент повинен йти на іспит.

У разі, якщо по закінченню сесії здобувач не набрав 60 балів, його відправляють на комісію для розгляду результатів підсумкового контролю, яка рекомендує здобувачу із академічною заборгованістю покращити результати підсумкового контролю шляхом: повторного складання підсумкового контролю, повторного вивчення освітньої компоненти, повторного курсу навчання.

Неформальна та інформальна освіта

Студенти мають право на часткове або повне перезарахування предмету за умови написання ними відповідної заяви та надання документів, які підтверджують ті результати навчання, які здобувач отримав (див. положення <https://ep3.nuwm.edu.ua/28363/>). Зокрема студенти можуть самостійно проходити онлайн-курси на таких навчальних платформах, як Prometheus, Coursera, edX, EdEra, FutureLearn та інших, для наступного перезарахування результатів навчання. Проте доцільно попередньо узгодити з викладачем відповідність обраного онлайн-курсу суті освітньої компоненти. Деякий перелік підходящих курсів наведено нижче:

- Coursera – Backend Development using ASP.Net (Розробка бекенду за допомогою ASP.Net);
- Class Central – Learn ASP.NET Core MVC (.NET 6) - Full Course (Вивчайте ASP.NET Core MVC (.NET 6) - повний курс);
- Coursera – ASP.NET for Experienced Developers Specialization (Спеціалізація ASP.NET для досвідчених розробників).

Зручний пошук курсів доступний тут: <https://www.classcentral.com/>.

Окрім того, якщо з'являються обставини для здобуття неформальної чи інформальної освіти від викладачів-практиків, то пропонуються ці можливості для студентів; рекомендуються відео-уроки практикуючих програмістів з YouTube тощо.

Правила академічної доброчесності

Задля запобігання академічної недоброчесності вимагається наступне:

- кожна студентська команда повинна виконувати власний проект;
- кожне завдання у проекті студент виконує самостійно, відмічаючи це у Jira та GitHub;
- студент отримує хоча б якусь оцінку лише за умови розуміння програмного коду виконаного завдання;
- студентам забороняється: плагіатити, самоплагіатити, фабрикувати, фальсифікувати, списувати, обманювати або будь-яким чином намагатись вплинути на викладача.

Залежно від виду та ступеня порушення викладач може накладати наступні санкції:

- усне або письмове зауваження від викладача;
- попередження про можливість притягнення до академічної відповідальності;
- зниження чи анулювання результатів оцінювання навчального завдання здобувача вищої освіти;
- повторне виконання навчального завдання;
- виконання іншого навчального завдання;
- призначення додаткового навчання з питань академічної доброчесності;
- призначення додаткових контрольних заходів (додаткові індивідуальні навчальні завдання, тести тощо);
- подання клопотання на ім'я ректора з метою порушення формальної процедури розгляду питання про притягнення студента до відповідальності.

За списування під час проведення модульного контролю чи підсумкового контролю студент позбавляється подальшого права здавати матеріал і у нього виникає академічна заборгованість.

Документи стосовно академічної доброчесності (про плагіат, порядок здачі курсових робіт, кодекс честі студентів, документи Національного агентства стосовно доброчесності) наведені на сторінці «Якість освіти» сайту НУВГП – <http://nuwm.edu.ua/sp/akademichna-dobrochesnistj>.

Вимоги до відвідування

Санкції за пропуски пар не передбачені. Студент має право самостійно вивчити необхідний для здачі модульних контролів та лабораторних робіт матеріал, який в повному обсязі дублюється викладачем одночасно на платформі Moodle та/або у групі з даного предмету в месенджері Telegram. Також викладач розміщує відеозаписи пар на YouTube. У разі необхідності проведення консультації – викладач йде назустріч.

Відвідування пари допускається із використанням власного ноутбука. Студенти не повинні порушувати дисципліну на парі.

Для студентів, які знаходяться на індивідуальному плані навчання, надаються індивідуальні завдання.

Автор
Доцент ОТ

Михайло БОЙЧУРА

Затверджено

Проректор з науково-педагогічної та навчальної роботи

Валерій СОРОКА



документ підписаний КЕП
Номер документа СИЛ №756
Підписувач Сорока Валерій Степанович
Підписувач (дані КЕП):
Сертифікат 58E2D9E7F900307B04000000807E2D0054327D00