

# АНОТАЦІЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ



## ДИСЦИПЛІНА ВІЛЬНОГО ВИБОРУ З КАФЕДРАЛЬНОГО КАТАЛОГУ «Новітні інформаційні сервіси»

Відеозвернення:	<a href="https://dn.nung.edu.ua/pluginfile.php/253388/mod_page/intro/%D0%9D%D0%BE%D0%B2%D1%96%D1%82%D0%BD%D1%96%20%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D1%96%20%D1%81%D0%B5%D1%80%D0%B2%D1%96%D1%81%D0%B8.mp4">https://dn.nung.edu.ua/pluginfile.php/253388/mod_page/intro/%D0%9D%D0%BE%D0%B2%D1%96%D1%82%D0%BD%D1%96%20%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D1%96%20%D1%81%D0%B5%D1%80%D0%B2%D1%96%D1%81%D0%B8.mp4</a> <a href="https://dn.nung.edu.ua/mod/page/view.php?id=130856&amp;forceview=1">https://dn.nung.edu.ua/mod/page/view.php?id=130856&amp;forceview=1</a>
Мова викладання:	Українська
Кількість студентів, які можуть вивчати дисципліну (мінімальна - максимальна):	25-100 ст.
Семестр, в якому викладається:	7
Окрім спеціальностей / ОП (за необхідності)	-
<b>Для спеціальностей/ОП (за необхідності)</b>	<b>121 Інженерія програмного забезпечення</b>
Кількість кредитів ЄКТС / академічних годин (вказати окремо лекції, лабораторні заняття, практичні заняття, самостійна робота тощо)	3 90 год (18 год. лекції, 24 год. лаб., 48 год самостійна робота) (2 лек, 10 лаб, 78 сам (заочна форма))
Форма підсумкового контролю та наявність індивідуальних завдань:	диференційований залік
Кафедра, що забезпечує викладання:	Інженерія програмного забезпечення
Викладач (викладачі), окремо за видами навантаження:	Лекції: Мельник Віталій Дмитрович, к.т.н., доцент Лабораторні: Мельник Віталій Дмитрович к.т.н., доцент; Козак Олексій Федорович, кандидат технічних наук, старший викладач.
Пререквізити:	Попередніх вимог для вивчення дисципліни немає. Перевагою буде наявність в здобувача знань основ програмування та алгоритмічного мислення.
Перелік компетентностей, яких набуде студент після опанування даної дисципліни:	<ul style="list-style-type: none"> <li>- здатність до абстрактного мислення, аналізу та синтезу;</li> <li>- здатність до пошуку, оброблення та аналізу інформації з різних джерел;</li> <li>- здатність аналізувати предметні області, ідентифікувати, класифікувати та формулювати вимоги;</li> <li>- здатність розробляти архітектури, модулі та компоненти програмних систем;</li> <li>- знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення;</li> <li>- уміти вибирати та використовувати відповідні задачі та методологію створення програмного забезпечення.</li> </ul>
Особливості навчання на курсі:	Практично-орієнтована дисципліна. Теорія – у мінімально-необхідному обсягу, більшість завдань організовано для реалізації практичного проекту.
Матеріально-технічне забезпечення:	Мультимедійні лекційні аудиторії 1102, А-11, А-12, А-13, А-14, оснащені проєктором, медіаплеєром, динаміками та екраном. Комп'ютерні класи (1418, 1419) з сучасним програмним забезпеченням для доступу до веб-ресурсів університету та кафедри та дистанційного навчання. Програмне забезпечення - .NET Framework, .Visual Studio Code.
Посилання на ЕНК на платформі Moodle (dn.nung.edu.ua):	<a href="https://dn.nung.edu.ua/course/view.php?id=4446">https://dn.nung.edu.ua/course/view.php?id=4446</a>

Посилання на інші матеріали за дисципліною (за наявності):

- <https://azure.microsoft.com/en-us/>
- <https://visualstudio.microsoft.com/>
- <https://learn.microsoft.com/uk-ua/dotnet/core/runtime-config/>
- <https://drawio-app.com/tutorials/>
- <https://www.visual-paradigm.com/tutorials/>
- <https://asana.com/guide/get-started/begin/quick-start>

Стислий опис дисципліни, в тому числі перелік тем теоретичного курсу, практичних та лабораторних занять, семінарів тощо:

Мета дисципліни «Новітні інформаційні сервіси» – отримання знань та вмінь на основі технології створення розподілених програмних додатків на базі платформи Microsoft .NET, а також теоретичних та практичних аспектів реалізації розподілених інформаційних систем та новітніх інформаційних сервісів.

Основними завданнями курсу є:

- засвоєння основних понять базових принципів розробки настільних та веб-застосунків й сервісів з використанням технології MS.NET Framework, .NET Core, Entity Framework, Windows Forms, ASP.NET Web API;

- набуття знань з питань використання можливостей технології ASP.NET MVC, їх переваги та недоліки.

**Перелік тем теоретичного курсу:**

- Програмна технологія «MS .NET Framework». Особливості архітектури та програмного середовища «.NET Framework»;
- Універсальна система типізації «Microsoft .NET» (Common Type System (CTS));
- Архітектурні рішення у «Microsoft .Net Framework», «Microsoft Cloud», збірка, керований та некерований код;
- Технології реалізації за стосунків «MS .NET Framework»;
- Технології та засоби реалізації застосувань у «MS .NET Framework» та C#. Технології та робота з графічними інтерфейсами, створення моделей;
- «MS ASP.NET Core MVC» як багатофункціональна платформа для створення web-додатків та API-інтерфейсів;
- Робота зі сховищами даних на основі «MS .NET-застосунків» з використанням бібліотек «ADO.NET». Архітектура «REST API» (Representational State Transfer) для хмарних сервісів та функціонування у розподіленому середовищі.

**Перелік тем лабораторних робіт:**

- Створення проекту та підключення бібліотек. Робота з типами даних та операторами. Використання колекцій. Робота з файлами. Демонстрація CLS;
- Аналіз та створення базових класів та бібліотек класів у «Common Language Runtime» (CLR) середовища «MS .NET Framework», забезпечення безпеки та керування пам'яттю під час виконання програми на прикладі консольного застосунку;
- Створення та аналіз основних та загальних типів даних (CTS) в «.NET», класів, структур, перелічень, простих інтерфейсів та делегатів;
- Створення на аналіз моделей розгортання хмарних рішень «Microsoft Cloud», алгоритмів збору сміття для вивільнення пам'яті та механізмів взаємодії виконуваного та не виконуваного коду;
- Створення консольних за стосунків в «C#» та та «.NET Framework»: використання класів з конструкторами, ініціалізаторами та деструкторами, створення та управління об'єктами;
- Використання платформи «ASP.NET Core MVC» для створення web-додатків та API-інтерфейсів з елементами маршрутизації, фільтрів, залежності та перевірки шаблонів;
- Архітектура «REST API» та її застосування для розробки web-сервісів. Набуття практичних навиків розробки web-сервісів та за стосунків.