НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА ПРИРОДОКОРИСТУВАННЯ

Навчально-науковий інститут автоматики, кібернетики та обчислювальної техніки

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**

РІВНЕ – 2023

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. 2023. 13 стор.

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

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

Силабус схвалений на засіданні кафедри обчислювальної техніки
Протокол № 13 від "03" червня 2023 року

Завідувач кафедри: *Круліковський Б.Б., к.т.н., доцент*.

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

Схвалено науково-методичною радою з якості ННІ АКОТ
Протокол № 8 від "19" червня 2023 року

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

Попередня версія силабусу: відсутня.

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ	
Інженерія програмного забезпечення	
ЗАГАЛЬНА ІНФОРМАЦІЯ	
Ступінь вищої освіти	*Бакалавр*
Освітня програма	*Комп'ютерна інженерія*
Спеціальність	*123 Комп'ютерна інженерія*
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 години*
Лабораторні заняття:	*26 годин*
Самостійна робота:	*100 годин*
Курсова робота:	*Ні*
Форма навчання	*денна/заочна*
Форма підсумкового контролю	*Екзамен*
Мова викладання	*державна*
ІНФОРМАЦІЯ ПРО РОЗРОБНИКА	
Лектор 	*Бойчура Михайло Володимирович* *к.т.н., старший викладач кафедри обчислювальної техніки*
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	
Дуже часто проектування програмного забезпечення студенти здійснюють інтуїтивно, опускаючи доцільність побудови UML-діаграм, здійснення подальшого рефакторингу, ігнорують Coding Conventions, не орієнтуються у таких важливих принципах проектування, як	

SOLID, DRY, KISS тощо. Дана дисципліна орієнтована на те, щоб і заповнити «пробіли» у знаннях студентів, і навчити їх тонкощам побудови порівняно великих і складних проектів.

***Метою** дисципліни «Інженерія програмного забезпечення» є отримання студентами як базових, так і поглиблених сучасних знань і практичних навичок у проектуванні програмного забезпечення.*

*Основним **завданням** дисципліни є набуття знань, умінь та навичок (компетентностей), необхідних для проектування та програмування порівняно великих і складних додатків, відповідно до кваліфікації бакалавр з комп'ютерної інженерії.*

Оновлення

*У 2023 році **змінено концепцію** предмету відповідно до силабусів інших навчальних закладів, бачення нового викладача, рекомендацій студентів та стейкхолдерів. За рекомендаціями останніх глибоко вивчаються GoF-патерни на мові C#. Окрім цього, викладач, зважаючи на результати опитувань студентів, вирішив викладати вірні практики проектування програмного забезпечення (рефакторинг коду, принципи SOLID тощо). Окремі студенти групи КІ-21 поділились власним досвідом командної розробки проектів, побажаннями щодо вивчення деяких тем та посиланнями на ефективні відео-уроки. Всі ці речі враховано в силабусі.*

Завдання до лабораторних робіт спроектовані таким чином, щоб в подальшому розроблені проекти можна було розвивати в межах інших дисциплін.

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 освоєння даної дисципліни у повній мірі необхідно отримати основні знання та навички з дисциплін: Вступ до спеціальності (ОК 10), Дискретна математика (ОК 11), Алгоритми та методи обчислень (ОК 12); потрібно поглиблено знати програмування та пройти практичну підготовку з програмування, що забезпечують ОК 13 та ОК 14, відповідно.

Отримані навички, наприклад, частково можуть використовуватись при читанні дисциплін: Програмування мобільних пристроїв (ВБ 8.4) та Системне програмування (ОК 25).

Компетентності

Z1. Здатність до абстрактного мислення, аналізу і синтезу.

Р6. Здатність проектувати, впроваджувати та обслуговувати комп'ютерні системи та мережі різного виду та призначення.

Р7. 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.

Р8. Готовність брати участь у роботах з впровадження комп'ютерних систем та мереж, введення їх до експлуатації на об'єктах різного призначення.

Р15. 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, обґрунтувати, 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. Методологія розробки програмного забезпечення на прикладі мови C#.Net			
1	Основні поняття та характеристика дисципліни Інженерія програмного забезпечення. Знайомство з мовою C#.Net. 2 год. лекцій 2 год. лабораторних N3, N12, N25	Наведення визначень основних понять. Класифікація інформаційних систем. Методології розробки програмного забезпечення. Моделі розробки програмного забезпечення.	Огляд платформи .Net. Знайомство з інтерфейсом Microsoft Visual Studio 2022, базовим синтаксисом мови C#, класами Console та Math, інтерполяцією, плейсхолдерами, деякими рекомендаціями Coding Conventions, набуття (повторення) базових

			практичних навичок відлагодження програми.
2	Процес створення інформаційної системи. Спільні та відмінні речі в C++ і C#. 2 год. лекцій 2 год. лабораторних N10, N12, N25	Життєвий цикл інформаційної системи. Трудомісткість стадій створення інформаційної системи. Структура проектної документації. Учасники процесу створення інформаційної системи. Методи та засоби створення інформаційної системи. Технологія створення інформаційної системи.	Закріплення навичок відлагодження програми та практичних навичок використання IntelliSense. Ключове слово var. Масиви. Цикли. Оператор вибору switch. Модифікатори ref, out, in. Параметри за замовчуванням. Ключове слово params. Локальні функції. Ключове слово enum. Посилкові та значеннєві типи. Області видимості змінних та констант. Методи. Перевантаження методів. Конструктори. Ініціалізатори. Деконструктори. Класи. Структури. Поля. Ключове слово this. Модифікатори public, private.
3	Технологія підготовки загальних рішень щодо створення інформаційної системи. Спільні та відмінні речі в C++ і C#. 2 год. лекцій 2 год. лабораторних N10, N12, N25	Формування вимог до інформаційної системи. Розробка концепції інформаційної системи. Технічне завдання. Передпроектна документація. Обстеження інформаційної системи.	
4	Технологія техноробочого проектування інформаційних систем. Вступ до об'єктно-орієнтованого програмування. 2 год. лекцій 2 год. лабораторних N10, N12, N25	Технічний проект. Робоча документація. Визначення структури інформаційної системи. Розробка постановки задач. Класи. Структури. Властивості. Наслідування. Поліморфізм. Абстрактні класи.	Ключове слово static. Доступ до констант класу. Упакування та розпакування. Практика користування принципами об'єктно-орієнтованого програмування. Проектування та розробка великого додатку. Модифікатор protected. Структурування коду. Створення власних бібліотек dll. Властивості. Абстрактні класи. Абстрактні методи. Інтерфейси.
5	Поглиблене вивчення об'єктно-орієнтованого програмування на C#.Net. Розробка першого великого проекту. UML-діаграми. 2 год. лекцій 2 год. лабораторних N3, N9, N12, N23, N25	Інкапсуляція. Абстракція. Абстрактні методи. Інтерфейси. Закріплення наслідування, поліморфізму, абстрактних класів. Види UML-діаграм.	
МОДУЛЬ 2. Змістовий модуль 2. Патерни проектування			
6	Вірні практики проектування. Патерн «Одинак». UML-діаграми класів. 2 год. лекцій 2 год. лабораторних N3, N12, N23, N25	Вірні практики проектування (SOLID, DTY, KISS, Big Design Up Front). Огляд поведінкових, структурних і породжуючих GoF-патернів. UML-діаграми класів. Патерн «Одинак»: шаблон коду, переваги та недоліки, UML-діаграма класів.	Модифікація розробленого додатку із використанням патернів. Реалізація поведінкового («Хранитель»), структурного («Фасад») і породжуючого («Одинак») GoF-патернів. Побудова UML-діаграми класів.
7	Патерни «Фасад», «Хранитель». UML-діаграми класів. 2 год. лекцій	Патерни «Фасад» і «Хранитель»: приклади практичного застосування, переваги, недоліки,	

	2 год. лабораторних N9, N12, N23, N25	покрокова реалізація, UML-діаграми класів.	
8	Патерн «Фабричний метод». UML-діаграми класів. 2 год. лекцій 2 год. лабораторних N12, N23, N25	Патерн «Фабричний метод»: приклад практичного застосування, переваги, недоліки, покрокова реалізація, UML-діаграма класів.	Модифікація розробленого додатку із використанням патернів. Огляд породжуючого патерна «Фабричний метод» і поведінковаго патерна «Спостерігач». Закріплення навичків побудови UML-діаграми класів.
9	Патерн «Спостерігач». UML-діаграми класів. 2 год. лекцій 2 год. лабораторних N12, N23, N25	Патерн «Спостерігач»: приклади практичного застосування, переваги, недоліки, покрокова реалізація, UML-діаграми класів.	
10	Робота з проектами інших розробників. Патерн «Адаптер». 2 год. лекцій 2 год. лабораторних N12, N23, N25	Патерн «Адаптер»: приклади практичного застосування, переваги, недоліки, покрокова реалізація, зв'язок з іншими патернами, UML-діаграми класів.	Дослідження та удосконалення проектів одногрупників. Розвиток навичок читання коду інших розробників. Більш глибоке дослідження та застосування структурних патернів проектування.
11	Робота з проектами інших розробників. Патерни «Декоратор», «Проксі». 2 год. лекцій 2 год. лабораторних N9, N12, N23, N25	Патерни «Декоратор» та «Проксі»: приклади практичного застосування, переваги, недоліки, покрокові реалізації, зв'язки з іншими патернами, UML-діаграми класів.	
12	Об'єднання декількох великих консольних додатків у єдиний desktop-додаток. 2 год. лекцій 4 год. лабораторних N3, N9, N12, N23, N25	Проектування та програмування великих додатків із застосуванням UML-діаграми класів. Елементи керування: Button, GroupBox, Label, CheckBox, RadioButton, TextBox, ListBox, ComboBox.	Робота з великими консольними додатками. Перетворення консольних додатків на desktop-додатки засобами Windows Forms.

Форми, методи та технології навчання

Форми навчання	• *очна (денна) з, можливо, елементами дистанційного навчання;* • *заочна.*
Форми навчального процесу	• *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	• *робота в малих групах (команді) та*

технології навчання	*індивідуальна робота;* • *проектна технологія;* • *аналіз конкретних ситуацій (case study): ситуація-оцінка;* • *контекстне навчання;* • *проблемне навчання.*
Процес навчання включає, зокрема, наступне	• *написання комп'ютерних програм;* • *відлагодження програм;* • *Code Review;* • *слідування рекомендаціям Coding Conventions та принципу Big Design Up Front.*
Засоби навчання	• *відео-запис лекції;* • *презентація;* • *підручник;* • *конспект лекцій;* • *різні туторіали.*

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.*

Порядок оцінювання програмних результатів навчання/результатів навчання

Студент може отримати 100 балів, враховуючи наступну розбаловку:

1. *модульні контролі: 40 балів;*
2. *поточний контроль: 50-60 балів;*
3. *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 лабораторні роботи (50-60 балів):*

Передбачено по 4 бали за перші 5 лабораторних робіт та по 5 балів – за решту; у випадку правильного виконання лабораторної роботи оцінка лінійно залежить від відсотка розуміння коду. Як

альтернатива, студенти можуть виконувати завдання на інших мовах програмування чи застосовувати інші патерни за умови попереднього узгодження деталей з викладачем.

3. *додаткові бали за вагому громадянську та студентську активність (0-10 балів):*

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

Поєднання навчання та досліджень

Кожен студент навіть для отримання 60 балів повинен розробити досить серйозний та великих проект, який характеризується високим рівнем абстракції, побудувати відповідну UML-діаграму класів, дослідити та вірно реалізувати низку патернів проектування. Розроблений проект потребує 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. Більше того, створене програмне забезпечення у перспективі може бути адаптоване під мобільні додатки, веб-додатки, крос-платформну парадигму, бази даних тощо. Відповідні предмети вивчаються на старших курсах.

Вивчення даної дисципліни може бути хорошим фундаментом для початку та продовження наукових досліджень аж до етапів захисту бакалаврської і магістерської робіт.

Рекомендована література (основна, допоміжна)

Основна література

1. *Сидоров М.О. Вступ до інженерії програмного забезпечення: курс лекцій. Київ: Видавництво Національного авіаційного університету «НАУ-друк», 2010. 112 с.*
2. *Табунщик Г.В., Каплієнко Т.І., Петрова О.А. Проектування та моделювання програмного забезпечення сучасних інформаційних систем. Запоріжжя : Дике Поле, 2016. – 250 с.*
3. *Gamma E., Vlissides J., Johnson R., Helm R. Design Patterns: Elements of Reusable Object-Oriented Software. Boston: Addison-Wesley. 1994. 395 p.*
4. *Будай А. Дизайн-патерни — просто, як двері. 2012. 90 с. [електронний ресурс]. URL: https://learn.ztu.edu.ua/pluginfile.php/632/mod_resource/content/1/DesignPatterns_AndriyBuday.pdf (Last access: 14.01.2023).*
5. *Коноваленко І.В. Платформа .NET та мова програмування C# 8.0: навчальний посібник. Тернопіль: ФОП Паляниця В. А., 2020. 320 с.*

Допоміжна література

1. *Стрюк А.М. Інженерія програмного забезпечення: перші 50 років становлення та розвитку. Computer Science & Software Engineering: proceedings of the 1 st Student Workshop (CS&SE@SW 2018), Kryvyi Rih, Ukraine, November 30, 2018. С. 11–36. URL: https://ceur-ws.org/Vol-2292/paper01.pdf (Last access: 14.01.2023).*
2. *Refactoring and Design Patterns. URL: https://refactoring.guru/ (Last access: 14.01.2023).*
3. *C# Coding Conventions | Microsoft Learn: https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/coding-style/coding-conventions (Last access: 14.01.2023).*
4. *Стандарт ECMA по C#: https://www.ecma-international.org/wp-content/uploads/ECMA-334_6th_edition_june_2022.pdf (Last access: 14.01.2023).*

Інформаційні ресурси в Інтернеті

1. *https://www.youtube.com/watch?v=Em094oeUE6c&list=PLbKZJf0OWepc4Ew9DPsY2T_8WJ6cqDMoL (Low Level Design Patterns | GoF 23 Design Patterns).*
2. *https://www.youtube.com/watch?v=tv-_1er1mWI (Intro to Design Patterns Teaser - step by step).*
3. *https://www.youtube.com/playlist?list=PL5neT6krvZKY2aq1pMF6V9ycr-DTu7arp (SMD /OOAD | Software Design & Modeling | Object Oriented Analysis & Design | UML | Unified Approach).*
4. *https://www.youtube.com/watch?v=dhnsegiPXoo&list=PLLWMQd6PeGY3ob0Ga6vn1czFZfW6e-FLr (Design Patterns & Principles).*

ПОЛІТИКИ ВИКЛАДАННЯ ТА НАВЧАННЯ

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

Вміння комунікувати	• *здатність спілкуватися державною мовою як усно, так і письмово;* • *вміння спілкуватись та писати із використанням англомовної професійної термінології;* • *навички усного спілкування;* • *навички письмового спілкування;* • *вміння писати зрозумілий код.*
Вміння сумісно працювати	• *вміння управляти часом;* • *навички управління проектами;* • *здатність планувати свій час у плані співставлення вимог, власних знань, здібностей і*

	дедлайнів; • *здатність працювати в команді;* • *навички міжособистісних відношень;* • *вміння надавати рекомендації іншим у коректній формі.*
Здатність до аналізу та синтезу	• *здатність критично мислити;* • *знаходити вихід з складних ситуацій;* • *здатність до навчання;* • *комплексне рішення проблем;* • *критичне мислення.*
Здатність застосовувати знання у практичних ситуаціях	

Дедлайни та перескладання

Дедлайн здачі лабораторних робіт – за 4 робочі дні до іспиту з даного предмету. 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 лекції. 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 здобувач не набрав 60 балів у результаті проходження іспиту (чи отримання оцінки «автоматом»), його відправляють на комісію з ліквідації академічної заборгованості. Якщо і тоді здобувач не набирає необхідних балів, то передбачається повторний курс.

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/18660/). Зокрема студенти можуть самостійно проходити онлайн-курси на таких навчальних платформах, як Prometheus, Coursera, edEx, edEra, FutureLearn та інших, для наступного перезарахування результатів навчання. Проте доцільно попередньо узгодити з викладачем відповідність обраного онлайн-курсу суті навчальної дисципліни. Деякий перелік підходящих курсів наведено нижче:

- *Pluralsight – C++ Design Patterns (Шаблони проектування C++);*
- *Pluralsight – Design Patterns in C# (Шаблони проектування в C#);*
- *Coursera – Design Patterns (Шаблони проектування).*

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:

- *кожен студент у групі виконує завдання згідно запропонованого йому варіанту або пропонує свою тему, яку обов'язково узгоджує з викладачем;*
- *студент отримує хоча б якусь оцінку лише за умови розуміння коду програми;*
- *студентам забороняється: плагіатити, самоплагіатити, фабрикувати, фальсифікувати, списувати, обманювати та будь-яким чином впливати на викладача, включаючи спроби хабарництва.*

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
Старший викладач — Михайло БОЙЧУРА

Затверджено

Проректор з науково-педагогічної та навчальної роботи — Валерій СОРОКА

документ підписаний КЕП
Номер документа СИЛ №525 від [sDateTime_SignWriteAgree_Last]
Підписувач Сорока Валерій Степанович
Підписувач (дані КЕП): [oSignECP.sSigner_Sert]
Сертифікат 58E2D9E7F900307B04000000807E2D0054327D00