



Інженерія програмного забезпечення

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	F (12) Інформаційні технології
Спеціальність	F7 (123) Комп'ютерна інженерія
Освітня програма	Комп'ютерні системи та мережі
Статус дисципліни	Нормативна
Форма навчання	очна(денна)
Рік підготовки, семестр	2 курс, весняний
Обсяг дисципліни	5 кредитів /150 год., лекції 30 год., практичні роботи 14 год, СРС 106 годин
Семестровий контроль/ контрольні заходи	Екзамен/Практичні роботи/Модульні контрольні роботи
Розклад занять	Згідно розкладу на весняний семестр поточного навчального року за адресою https://schedule.kpi.ua
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: старший викладач Васильєва Марія Давидівна, vasylieva.mariia@edu.kpi.ua Практичні роботи: старший викладач Васильєва Марія Давидівна, vasylieva.mariia@edu.kpi.ua
Розміщення курсу	На платформі дистанційного навчання «Сікорський»: https://classroom.google.com/c/ODY4MzMyNTc4MzQ3?cjc=owiaplv4

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Метою вивчення дисципліни “Інженерія програмного забезпечення” є надбання студентам теоретичних знань та практичних навичок, необхідних для ефективного планування, розробки, тестування, впровадження та супроводу програмних систем:

- вивчення процесів, методологій, інструментів та технологій, які забезпечують високу якість програмного забезпечення, його надійність, підтримуваність і масштабованість
- вивчення принципів проектування та побудови програмного забезпечення для комп'ютерних систем
- вивчення шаблонів проектування програмного забезпечення та мови UML
- вивчення методів та засобів розробки програмного забезпечення
- вивчення об'єктно-орієнтованого підходу до розробки програмного забезпечення.

Предметом дисципліни “Інженерія програмного забезпечення” є процеси, методи, моделі та інструменти, що застосовуються на різних етапах життєвого циклу програмного забезпечення.

Вивчення дисципліни забезпечує наступні загальні та фахові компетенції:

- ЗК01. Здатність до абстрактного мислення, аналізу та синтезу.
- ЗК02. Здатність вчитися і оволодівати сучасними знаннями.
- ЗК07. Вміння виявляти, ставити та вирішувати проблеми.

- ФК01. Здатність застосовувати законодавчу та нормативно-правову базу, а також державні та міжнародні вимоги, практики і стандарти з метою здійснення професійної діяльності в галузі комп'ютерної інженерії.
- ФК02. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.
- ФК03. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.
- ФК14. Здатність проектувати системи та їхні компоненти з урахуванням усіх аспектів їх життєвого циклу та поставленої задачі, включаючи створення, налаштування, експлуатацію, технічне обслуговування та утилізацію.
- ФК15. Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення.

У відповідності до вищезазначеного, студенти отримають наступні результати навчання:

- ПРН05. Мати знання основ економіки та управління проектами.
- ПРН06. Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.
- ПРН07. Вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності.
- ПРН09. Вміти застосовувати знання технічних характеристик, конструктивних особливостей, призначення і правил експлуатації програмно-технічних засобів комп'ютерних систем та мереж для вирішення технічних задач спеціальності.
- ПРН10. Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати, типове для спеціальності обладнання.
- ПРН12. Вміти ефективно працювати як індивідуально, так і у складі команди.
- ПРН14. Вміти поєднувати теорію і практику, а також приймати рішення та виробляти стратегію діяльності для вирішення завдань спеціальності з урахуванням загальнолюдських цінностей, суспільних, державних та виробничих інтересів.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

При вивченні дисципліни «Інженерія програмного забезпечення» доцільно використовувати знання, отримані при вивченні попередніх дисциплін:

- Програмування. Частина 1
- Програмування. Частина 2

Дисципліна є базовою для курсів:

- Інженерія та автоматизація розроблення програмного забезпечення. Мультидисциплінарна курсова робота
- Комп'ютерні системи

3. Зміст навчальної дисципліни

Вступ

Розділ 1. Методологія розробки ПЗ.

Тема 1.1. Основні поняття та проблеми розробки ПЗ.

Тема 1.2. Життєвий цикл ПЗ; міжнародні стандарти життєвого циклу ПЗ.

Тема 1.3. Моделі та методології розробки ПЗ.

Тема 1.4. Аналіз, специфікація, верифікація та валідація вимог до ПЗ. Функціональні та нефункціональні вимоги.

Розділ 2. Засоби та середовища створення ПЗ.

Тема 2.1. Інтегровані середовища розробки ПЗ.

Тема 2.2. Системи управління версіями документів, архітектурні особливості (CVS, SVN, Git).

Тема 2.3. Інструменти автоматизації зборки проєктів (утиліта make, системи CMake, Ant, Maven, Gradle).

Розділ 3. Проєктування ПЗ з використанням шаблонів.

Тема 3.1. Графічна нотація UML. Документування програмного коду.

Тема 3.2. Принципи розробки програмного забезпечення.

Тема 3.3. Проєктування архітектури ПЗ.

Тема 3.4. Шаблони проєктування ПЗ. Класифікація шаблонів проєктування. Графічна нотація.

Тема 3.5. Структурні шаблони проєктування Composite, Decorator, Proxy.

Тема 3.6. Структурні шаблони проєктування Flyweight, Adapter, Bridge, Facade.

Тема 3.7. Шаблони поведінки Iterator, Mediator, Observer.

Тема 3.8. Шаблони поведінки Strategy, Chain of Responsibility, Visitor.

Тема 3.9. Шаблони поведінки Memento, State, Command, Interpreter.

Тема 3.10. Породжувальні шаблони проєктування Prototype, Singleton, Factory Method.

Тема 3.11. Породжувальні шаблони проєктування Abstract Factory, Builder.

Розділ 4. Проєктування клієнт-серверної архітектури.

Тема 4.1. Клієнт-серверна архітектура.

Тема 4.2. Веб-сервер.

Тема 4.3. Патерн проєктування MVC.

Розділ 5. Проєктування інтерфейсу користувача.

Тема 5.1. Призначення та типи інтерфейсів користувача.

Тема 5.2. Принципи побудови графічного інтерфейсу користувача.

Розділ 6. Методи забезпечення та контролю якості ПЗ.

Тема 6.1. Якість ПЗ. Метрики і стандарти якості ПЗ.

Тема 6.2. Верифікація та валідація ПЗ.

Тема 6.3. Тестування ПЗ.

Тема 6.4. Оптимізація коду та рефакторинг.

Тема 6.5. Аспекти продуктивності ПЗ.

Тема 6.6. Інструменти автоматизації процесів тестування (JUnit, Selenium, JMeter).

Розділ 7. Менеджмент програмних проєктів.

Тема 7.1. Задачі управління проєктами. Трикутник обмежень.

Тема 7.2. Управління ресурсами. Планування графіку виконання проєкту.

Тема 7.3. Управління ризиками програмного проєкту.

Тема 7.4. Контроль та моніторинг стану проєкту. Метрики контролю.

Тема 7.5. Організація роботи проєктної команди. Ролі та зони відповідальності учасників команди.

4. Навчальні ресурси та матеріали

4.1. Базова література

1. Методичні вказівки до виконання лабораторних робіт з дисципліни «інженерія програмного забезпечення» [Електронний ресурс] : Методичні вказівки. для студ. Спеціальності 123 «Комп'ютерна інженерія» / А.І. Антонюк, А.О. Болдак; КПІ ім. Ігоря Сікорського. - Київ : КПІ ім. Ігоря Сікорського, 2022. - 51 с. [<https://ela.kpi.ua/handle/123456789/52000>]
2. Проєктування інформаційних систем: Загальні питання теорії проєктування ІС (конспект лекцій): навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки» / КПІ ім. Ігоря Сікорського; уклад.: О. С. Коваленко, Л. М. Добровська. – Київ : КПІ ім. Ігоря Сікорського, 2020. – 192с. [<https://ela.kpi.ua/handle/123456789/33651>]
3. Компоненти програмної інженерії 1. Вступ до програмної інженерії [Електронний ресурс] : навч. посіб. для здобувачів ступеня бакалавра за освіт. програмою «Інженерія програмного забезпечення інформаційних систем» спец. 121 Інженерія програмного забезпечення / КПІ ім. Ігоря Сікорського ; уклад.: М. О. Сидоров. – Київ : КПІ ім. Ігоря Сікорського, 2025. – 248 с. [<https://ela.kpi.ua/handle/123456789/77531>]
4. Марченко, О. І. Компоненти програмної інженерії. Частина 1. Вступ до програмної інженерії. Лабораторний практикум [Електронний ресурс] : навч. посіб. для здобувачів ступеня

бакалавра за спеціальністю 121 «Інженерія програмного забезпечення» / Марченко О. І., Вітковська І. І. ; КПП ім. Ігоря Сікорського. – Київ : КПП ім. Ігоря Сікорського, 2022. – 50 с. [<https://ela.kpi.ua/handle/123456789/56535>]

4.2. Додаткова література

1. Design Patterns: elements of reusable object-oriented software / Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Indianapolis: - Addison-Wesley, 1994. -417 p. ISBN: 0201633612.
2. Інженерія програмного забезпечення: навч. посібник / Д. П. Кучеров, Є. Б. Артамонов. – К. : НАУ, 2017. – 388 с.
3. Mark Grand. Patterns in Java: A Catalog of Reusable Design Patterns Illustrated with UML, Volume 1, 2nd Edition. - Wiley Publishing, 2002. - 480 p. ISBN: 0471258393
4. Проектування та моделювання програмного забезпечення сучасних інформаційних систем / Г. В. Табунщик, Т.І. Каплієнко, О.А. Петрова – Запоріжжя : Дике Поле, 2016. – 250 с.
5. Навчальний посібник з дисципліни «Технології розробки програмного забезпечення» для студентів спеціальності 123 «Комп'ютерна інженерія» / Дегтярьова Л.М., Гроза П.М., Сомов С.В. – Полтава: ПолтНТУ, 2017. – 218 с.
6. Інженерія програмного забезпечення: Посібник для студентів вищих навчальних закладів /І. Л. Бородкіна, Г. О. Бородкін. – К.: Центр учбової літератури, 2020. – 204 с. ISBN 978-611-01-1232-1
7. С.В.Баран. Розробка програмного забезпечення з використанням патернів проектування: Навчальний посібник. – Кривий Ріг: Державний університет економіки і технологій, 2023. – 203 с.
8. Head First. Патерни проектування / Ерік Фрімен, Елізабет Робсон, Кеті Сьєрра і Берт Бейтс; пер. з англ. Г. Якубовська — Харків : ВД «Фабула», 2020. — 672 с. ISBN 978-617-09- 6159-4
9. Чиста архітектура: Мистецтво створення програмного забезпечення. Видання друге / пер. з англ. І.Бондар-Терещенко. – Харків. Вид-во «Ранок»: Фабула, 2023. – 368 с. ISBN 978-617-09-5286-8

4.3. Інформаційні ресурси

1. Дистанційний курс на платформі дистанційного навчання «Сікорський» в середовищі Google Workspace for Education: Інженерія програмного забезпечення. <https://meet.google.com/pta-kuod-kva> . На ресурсі розміщено додаткові теоретичні матеріали та приклади виконання практичних задач, презентації та відеозаписи лекцій, перелік питань та завдання до самостійного опрацювання, матеріали для підготовки до іспиту.
2. <https://se-111.blogspot.com/>
3. <https://refactoring.guru/uk>

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компоненту)

Розподіл навчального часу за видами занять і завдань з дисципліни згідно з робочим навчальним планом. На вивчення дисципліни виділено 150 годин, 5 кредитів, лекцій 30 годин, практичних робіт 14 годин, самостійної роботи 106 годин.

Лекційні заняття

Для досягнення мети навчальної дисципліни слід зосередитись в лекційному матеріалі, теми лекційних занять наведено в таблиці 1.

Таблиця 1
Лекційні заняття

№ лекції	Назва теми лекції та перелік основних питань
1.	Життєвий цикл ПЗ. Основні поняття та проблеми розробки ПЗ. Життєвий цикл ПЗ. Моделі життєвого циклу ПЗ. Міжнародні стандарти життєвого циклу ПЗ

2.	Інтегровані середовища розробки ПЗ. Автоматизація збірки ПЗ. Інтегровані середовища розробки ПЗ. Розширення функціональності IDE (додавання плагінів). Системи управління версіями документів, архітектурні особливості (CVS, SVN, Git). Інструменти автоматизації зборки проєктів (утиліта make, системи CMake, Ant, Maven, Gradle).
3.	Графічна нотація UML. Документування програмного коду. Уніфікована мова модулювання (UML). Діаграми UML. Засоби автоматизації UML моделювання. Документування за допомогою JavaDoc.
4.	Принципи розробки програмного забезпечення. Принципи DRY, KISS, YAGNI. SOLID принципи.
5.	Проектування архітектури ПЗ. Шаблони проектування ПЗ. Класифікація шаблонів проектування. Шаблони Composite, Decorator та Proxy. Мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблонів.
6.	Структурні шаблони проектування: Flyweight, Adapter, Bridge та Facade. Мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблонів.
7.	Шаблони поведінки: Iterator, Mediator, Observer та Strategy. Призначення шаблонів поведінки. Мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблонів.
8.	Шаблони поведінки: State, Memento, Command та Template. Мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблонів.
9.	Шаблони поведінки: Visitor, Interpreter та Chain of Responsibility. Мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблонів.
10.	Породжувальні шаблони Prototype, Factory Method, Abstract Factory, Singleton та Builder. Призначення породжувальних шаблонів. Мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблонів.
11.	Клієнт-серверна архітектура. Веб-сервер. Модель OSI. Мережеві протоколи. Клієнт-серверна архітектура. Протокол http. Поняття веб-сервера. Сервлет. Контейнер сервлетів (Tomcat). Cookie та Session. Шаблонізатори (Thymeleaf).
12.	Паттерн проектування MVC. Архітектурний паттерн Model View Controller. Складові частини MVC. Взаємодія між компонентами. Застосування MVC у популярних фреймворках (Spring MVC).
13.	Якість та тестування ПЗ. Автоматизація процесів тестування. Якість ПЗ. Метрики і стандарти якості ПЗ. Верифікація та валідація ПЗ. Тестування ПЗ. Модульне тестування (unit testing, JUnit). Підхід TDD. Інструменти автоматизації тестування веб-додатків (Selenium). Паттерн PageObject. Тестування швидкодії (JMeter).
14.	Постійна інтеграція ПЗ Принципи постійної інтеграції ПЗ. Continuous Integration. Continuous Delivery. Налаштування CI/CD процесів (GitHub Actions, Jenkins).
15.	Управління проєктами. Контроль та моніторинг стану проєкту Задачі управління проєктами. Трикутник обмежень. Управління змістом та якістю проєкту. Організація роботи проєктної команди. Ролі та зони відповідальності учасників команди.

Практичні роботи

Мета практичних робіт - набуття вмінь та навичок застосування методів і засобів інженерії програмного забезпечення на практиці, ознайомлення з сучасними технологіями розробки, тестування та супроводу програмних систем. У ході виконання робіт студенти вчаться працювати з інструментами для проєктування, моделювання та керування вимогами, середовищами розробки (IntelliJ IDEA, Eclipse, Visual Studio Code), системами контролю версій (Git), а також з технологіями для створення веб- та десктопних застосунків. Практичні завдання спрямовані на закріплення теоретичних знань і формування досвіду командної роботи над програмними проєктами.

Практична робота включає:

1. Розробку та аналіз алгоритму рішення задачі;
2. Декомпозицію задачі;
3. Застосування шаблонів проєктування;
4. Роздруківку програми;
5. UML діаграма класів;
6. Результати виконання програми;
7. Документування практичної роботи та розробленої програми.

Таблиця 2
Тематика практичних робіт

№ з/п	Назва практичної роботи (комп'ютерного практикуму)	Кількість ауд. годин
1.	Проєкт програмного додатку. Збірка проєкту за допомогою інструментів автоматизації зборки проєктів (Ant, Maven, Gradle).	2
2.	Графічна нотація UML. Документування за допомогою JavaDoc.	2
3.	Структурні шаблони проєктування ПЗ. Composite, Decorator, Proxy, Flyweight, Adapter, Bridge, Facade.	2
4.	Шаблони поведінки. Iterator, Mediator, Observer, Strategy, Chain of Responsibility.	2
5.	Шаблони поведінки. Visitor, Memento, State, Command, Interpreter.	2
6.	Породжувальні шаблони. Prototype, Singleton, Factory Method, Abstract Factory, Builder.	2
7.	Шаблон проєктування MVC.	2
	Разом:	14

6. Самостійна робота здобувача вищої освіти

Види самостійної роботи для студентів денної форми навчання (106 годин):

- підготовка до аудиторних занять, виконання поточних домашніх завдань та опрацювання матеріалів лекцій (0,5 годин x 30 годин = 15 годин);
- виконання індивідуального завдання до практичних робіт, оформлення звіту до практичної роботи (1 година x 14 годин практичних занять (7 практичних робіт) = 14 годин);
- підготовка та виконання контрольних робіт (4 години);
- підготовка до екзамену (8 годин);
- самостійне опрацювання теоретичних тем, розгортання програмного оточення та завантаження вихідних кодів програмного забезпечення для виконання практичних робіт, робота з системою контролю версій Git (67 години).

Таблиця 3
Теми для самостійного опрацювання

№ з/п	Назва теми, що виноситься на самостійне опрацювання	Кількість годин
1.	Установка і використання систем управління проєктами. Установка і конфігурування SVN	4
2.	Установка і конфігурування систем збірки проєктів Maven та Gradle	4
3.	Встановлення та налаштування засобів автоматизації UML моделювання.	2
4.	Шаблони проєктування GRASP. Реалізації Composite та Decorator в класах Java SDK	4
5.	Реалізації Proxy та Flyweight в класах Java SDK	3
6.	Реалізації Adapter, Bridge та Facade в класах Java SDK	3
7.	Реалізації Iterator та Mediator в класах Java SDK	3
8.	Реалізації Observer та Strategy в класах Java SDK	3
9.	Реалізації State та Memento в класах Java SDK	3
10.	Реалізації Command та Visitor в класах Java SDK	3
11.	Реалізації Interpreter та Chain of Responsibility в класах Java SDK	3
12.	Реалізації Prototype, Factory Method та Abstract Factory в класах Java SDK	3
13.	Реалізації Singleton та Builder в класах Java SDK	3
14.	Використання класів Java для роботи з протоколами зв'язку, такими як HTTP і HTTPS.	4
15.	Паттерн Front Controller.	2
16.	Паттерн проєктування MVC. Реалізація MTV у Django	4
17.	Фреймовий та віконний інтерфейси користувача. Реалізація легких компонентів на прикладі SWING, JavaFX	2
18.	Сертифікація ПЗ в Україні.	2
19.	Автоматизована ініціалізація елементів сторінки за допомогою PageFactory.	2
20.	Установка і використання сервера постійної інтеграції	2
21.	Rational Unified Process	2
22.	Засоби автоматизації управління проєктами	2
23.	Робота команди з гнучкою методологією	2
	Разом	65

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Дедлайни. Для виконання практичних робіт встановлюються дедлайни. Дедлайни уточнюються і оприлюднюються викладачем на початку семестру.

Заохочувальні бали. Заохочувальні бали виставляються за підготовку повідомлення з презентацією та доповідь на відповідній лекції по одній із тем СРС дисципліни. Доповідь попередньо узгоджується з викладачем. Кількість заохочувальних балів не більше 5.

Практичні роботи. Перед виконанням кожної практичної роботи студент повинен підготувати індивідуальне завдання та оформити результати у вигляді протоколу. Готовий протокол необхідно завантажити до Classroom не пізніше ніж за день до початку відповідного практичного заняття. За потреби до протоколу додаються посилання на репозиторій Git із програмним кодом, а також інші матеріали, необхідні для перевірки та захисту роботи, зокрема відеозаписи виконання завдання (наприклад, під час дистанційного навчання або виконання віддаленого захисту роботи). Студенти, які не підготували та не подали протокол у встановлений термін, до захисту практичної роботи не допускаються. Захист складається з двох етапів: спочатку студент демонструє та обгрунтовує результати виконання індивідуального завдання, після чого проходить перевірку теоретичних знань у формі усного опитування або тестування. У дистанційному форматі навчання оцінювання здійснюється за допомогою тестових завдань, що охоплюють матеріал практичної роботи. Водночас викладач може додатково провести усне опитування, якщо виникають сумніви щодо самостійності виконання роботи або необхідно уточнити окремі аспекти отриманих результатів. Підсумкова оцінка за практичну роботу формується з урахуванням якості виконаного завдання, повноти та правильності оформлення протоколу та результатів захисту. Якщо студент не виконав або не захистив індивідуальне завдання, до проходження усного опитування або тестування він не допускається.

Бали, які студент може отримати за виконання кожної практичної роботи наведено в таблиці 4 оцінювання семестрових робіт, розділ 8 силабуса.

Поточна контрольна робота. Поточна контрольна робота не приймається поза встановленого терміну.

Екзаменаційна робота пишеться без застосування допоміжних засобів (мобільні телефони, планшети та ін.); результат пересилається до відповідної директорії Google – диску через Google форму. Бали за екзаменаційну роботу додаються до балів за практичні роботи та модульну контрольну роботи і складають семестрову рейтингову оцінку.

Загальною умовою допуску до семестрового контролю (екзамену) є зарахування всіх практичних робіт та рейтинг не менше 40 балів.

Правила відвідування занять. Відвідування занять є вільним, бали за присутність на лекціях та практичних заняттях не додаються, відповідно штрафні бали не передбачаються. Втім, вагома частина рейтингу формується через активну участь у заходах на практичних заняттях.

Дистанційний режим навчання. У разі запровадження обмежень на відвідування університету, пов'язаних з введенням карантину або режиму воєнного стану в державі, освітній процес здійснюється у дистанційному режимі відповідно до Положення про дистанційне навчання в КПІ ім. Ігоря Сікорського (<https://osvita.kpi.ua/index.php/node/188>), Регламенту організації освітнього процесу в дистанційному режимі (<https://profkom.kpi.ua/reglament-organizatsiyi-osvitnogo-protsesu-v-distantsiynomu-rezhimi>) та Регламенту проведення семестрового контролю в дистанційному режимі (<https://osvita.kpi.ua/node/148>). У режимі дистанційного навчання заняття відбуваються у вигляді онлайн-конференції на платформах Google Meet, Zoom. Посилання на конференцію видається на початку семестру і розміщується в Google Classroom. З метою забезпечення якісної підготовки здобувачів, дистанційний курс дисципліни розміщено на Платформа дистанційного навчання «Сікорський» (<https://www.sikorsky-distance.org>). Результати оцінювання висвітлюють у АС «Електронний кампус» на особистій сторінці здобувача (<https://ecampus.kpi.ua>), а також в Google Classroom.

Правила поведінки на заняттях. На заняттях слід дотримуватись норм етичної поведінки визначених у Кодексі честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (<https://kpi.ua/code>). На території університету здобувачі мають поводити себе відповідно до Правил внутрішнього розпорядку (<https://kpi.ua/admin-rule>).

Визнання результатів навчання, набутих у неформальній/інформальній освіті. Положення про визнання результатів навчання, набутих у неформальній / інформальній освіті (<https://osvita.kpi.ua/index.php/node/179>) регламентує визнання результатів навчання, набутих у неформальній/інформальній освіті:

- у разі навчання на неформальній освіті, яку студент обрав самостійно і хоче зарахувати результати навчання в рамках дисципліни, проводиться процедура валідації, що передбачає подання здобувачем заяви на ім'я декана, декларації підтверджувальних документів; рішення про визнання чи не визнання приймається комісією у складі завідувача кафедри, викладача, гаранта освітньо-професійної програми.

- за рішенням викладача окремі змістовні модулі або теми дисципліни можуть бути зараховані на підставі результатів проходження здобувачем онлайн-курсу або іншого елементу неформальної освіти; перелік завдань та елементів неформальної освіти визначається викладачем заздалегідь і розміщується на платформі дистанційного навчання «Сікорський», про що студенти повідомляються на початку семестру; у цьому випадку здобувач звільняється від виконання відповідних завдань і отримує максимальний бал згідно з рейтинговою системою оцінювання.

Політика використання штучного інтелекту. Використання штучного інтелекту регламентується «Політикою використання штучного інтелекту для академічної діяльності в КПІ ім. Ігоря Сікорського» (<https://osvita.kpi.ua/node/1225>). Усі завдання, як під час виконання навчальних завдань з дисципліни, так і індивідуальні завдання, мають бути результатом власної оригінальної роботи здобувача. Використання ШІ для автоматичної генерації відповідей без подальшого їх аналізу та доопрацювання заборонено.

Порушення академічної доброчесності: використання плагіату. У разі виявлення фактів використання плагіату під час виконання навчальних завдань, студент втрачає право на подальше виконання поточних завдань у межах навчальної дисципліни до з'ясування обставин порушення.

Студент зобов'язаний аргументовано пояснити причини використання плагіату, а також засвідчити готовність дотримуватись принципів академічної доброчесності в подальшому навчанні (**письмово у вигляді пояснювальної записки**); повторно виконати завдання або, за потреби, виконати додаткове завдання відповідно до бачення викладача з урахуванням вимог щодо самостійності.

Зазначені дії регламентуються Кодексом честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (<https://kpi.ua/code>).

8. Види контролю та рейтингова система оцінювання результатів навчання (РСО)

Семестровий рейтинг студента з дисципліни розраховується, виходячи із 100-бальної шкали. Семестровий рейтинг складається із стартової (поточної) оцінки R_C та екзаменаційної R_E . Стартовий рейтинг складається з балів, що студент отримує за виконання 7 практичних робіт $R_{ПР}$, поточної контрольної роботи $R_{ПКР}$ та заохочувальних балів.

$$R_C = R_{ПР} + R_{ПКР}.$$

Максимальна кількість балів за практичні роботи складає 42 балів, тобто $R_{ПР} = 42$. Виконання модульних контрольних робіт оцінюється в 9 балів за кожну контрольну роботу, тобто $R_{ПКР} = 18$.

Критерії оцінювання практичних робіт наступні:

- Протокол: Відповідність протоколу завданню практичної роботи та вимогам до оформлення:
0 балів - протокол відсутній або не відповідає завданню практичної роботи.
0,5 бали - протокол відповідає завданню, але матеріал викладено неповно, результати виконання не відображені.
1 бал - протокол повністю відповідає завданню практичної роботи та встановленим вимогам до оформлення, всі посилання коректно відображаються, матеріал викладено повно, надані результати виконання роботи (скріншоти, діаграми тощо).
- Виконання завдання: написаний код, розроблені діаграми повністю відповідають завданню практичної роботи, програма працює коректно, без помилок та відповідає очікуваним результатам, студент вільно орієнтується у власній розробці, вміє пояснити обраний підхід та фрагменти реалізації, послідовно демонструє результати виконання.
- Опитування/тест: для зарахування теоретичної частини роботи студент відповідає на декілька обраних викладачем теоретичних питань з переліку питань наведеного у практичній роботі або у випадку дистанційного захисту складає тест за тематикою практичної роботи.

Критерії оцінювання усної відповіді:

Повнота та правильність відповіді, володіння фаховою термінологією	Студент повністю розкриває теоретичне питання, правильно використовує терміни та поняття предметної області.	1 бал
Логічність та послідовність відповіді	Студент структуровано викладає матеріал, встановлює причинно-наслідкові зв'язки між поняттями.	1 бал
Розуміння матеріалу та вміння застосовувати знання	Студент здатен наводити приклади, пояснювати принцип роботи методів та засобів, що використовуються в практичній роботі.	1 бал

Таблиця 4

Деталізація балів за поточні роботи за семестр наведені в наступній таблиці

Назва заняття	Форма контролю	Кількість балів	Мінімальна кількість балів для зарахування	Всього балів
Практична робота 1	Протокол	1	4	6
	Виконання завдання	4		
	Опитування/тест	3		
Практична робота 2	Протокол	1	4	6
	Виконання завдання	4		
	Опитування/тест	3		
Практична робота 3	Протокол	1	4	6
	Виконання завдання	4		
	Опитування/тест	3		
Практична робота 4	Протокол	1	4	6
	Виконання завдання	4		
	Опитування/тест	3		
Практична робота 5	Протокол	1	4	6
	Виконання завдання	4		
	Опитування/тест	3		
Практична робота 6	Протокол	1	4	6
	Виконання завдання	4		
	Опитування/тест	3		
Практична робота 7	Протокол	1	4	6
	Виконання завдання	4		
	Опитування/тест	3		
Модульна контрольна робота 1		9	6	9
Модульна контрольна робота 2		9	6	9

Всього балів		60	40	60
--------------	--	----	----	----

Максимальна кількість балів за екзамен дорівнює **R_E =40 балів**.

Екзамен проводиться у вигляді автоматизованого тестування на платформі Google Workspace for Education, складається з тестових питань та практичного завдання (40 балів).

Опис практичного завдання: Студент повинен розробити структурно-функціональну схему функціонального вузла комп'ютера за варіантом. Під час виконання завдання враховуються:

- правильність виконання завдання;
- коректність програмної реалізації;
- обґрунтування вибору обраного методу реалізації;
- якість програмного коду;
- уміння пояснити реалізоване рішення.

Критерії оцінювання практичного завдання:

Критерій	4 бали	3 бали	2 бали	1 бал	0 балів
Правильність виконання завдання	Завдання виконано повністю та без помилок	Є незначні помилки, що не впливають на результат	Виконано більшу частину завдання, наявні помилки	Виконано лише окремі частини завдання	Завдання не виконано
Коректність програмної реалізації	Код працює коректно для всіх передбачених випадків. Код структурований, читабельний, відповідає принципам ООП та SOLID	Наявні дрібні недоліки в реалізації	Код працює, але потребує незначного доопрацювання	Код містить суттєві помилки, його важко читати та підтримувати	Код не працює або зовсім не відповідає вимогам
Вибір та застосування шаблону проєктування (в практичних роботах на реалізацію шаблону проєктування)	Шаблон обрано правильно, вибір обґрунтовано	Шаблон обрано правильно, але реалізовано з незначними недоліками	Шаблон частково відповідає завданню	Вибір шаблону недостатньо обґрунтований або реалізація некоректна	Шаблон не використано або використано неправильно
Пояснення рішення	Студент повністю пояснює логіку роботи та прийняті рішення	Пояснення правильне, але недостатньо повне	Пояснення неповне	Пояснення містить суттєві неточності	Студент не може пояснити рішення

Календарна атестація студентів з дисципліни проводиться за значенням поточного рейтингу студента на час атестації. Якщо значення цього рейтингу не менше 50 % від максимально можливого на час атестації, студент вважається атестованим. В іншому випадку в атестаційній відомості виставляється «неатестовано».

Необхідною умовою допуску до екзамену студента є виконання і захист всіх практичних робіт з сумою балів не менше ніж 40 балів.

Кількість балів, що отримує студент за семестр визначається формулою

$$R = R_C + R_E.$$

Максимальна кількість балів за семестр не перевищує $R = 100$.

З урахуванням одержаної суми балів кінцева оцінка визначається наступною таблицею:

Таблиця 5. Визначення семестрової оцінки

<i>Кількість балів</i>	<i>Оцінка</i>
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно

Робочу програму навчальної дисципліни (силабус):

Складено, старший викладач кафедри ОТ Васильєва Марія Давидівна.

Ухвалено кафедрою ОТ (протокол № 18 від 24.06.2026)

Погоджено методичною комісією ФІОТ (протокол № 12 від 26.06.2026)