



Інженерія програмного забезпечення

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>123 Комп'ютерна інженерія</i>
Освітня програма	<i>Комп'ютерні системи та мережі</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>Очна(денна)</i>
Рік підготовки, семестр	<i>2 курс, осінній семестр</i>
Обсяг дисципліни	<i>5 кредитів /150 год., лекції 54 год., практичні роботи 18 год, СРС 78 год.</i>
Семестровий контроль/ контрольні заходи	<i>Екзамен, календарний контроль</i>
Розклад занять	<i>Згідно розкладу на весняний семестр поточного навчального року за адресою http://rozklad.kpi.ua/</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лекції: старший викладач, Васильєва Марія Давидівна, vasylieva.mariia@ill.kpi.ua Практичні роботи: старший викладач, Васильєва Марія Давидівна, vasylieva.mariia@ill.kpi.ua</i>
Розміщення курсу	<i>На платформі дистанційного навчання «Сікорський»: https://classroom.google.com/c/NzcwMzQzMTCxMiQy?cjc=mnkvrxaf</i>

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 та 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:

- ЗК2. Здатність вчитися і оволодівати сучасними знаннями.
 - ФК2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.
 - ФК3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.
 - ФК18. Здатність розробляти, адаптувати, використовувати програмно-технічні засоби для збирання, оброблення та інтелектуального аналізу даних в комп'ютерних системах: паралельних, розподілених, хмарних, безпечних, інтелектуальних, розумних, Інтернет речей, зокрема, з використанням штучного інтелекту.
- У відповідності до вищезазначеного, студенти отримають наступні результати навчання:
- ПРН5. Мати знання основ економіки та управління проектами.
 - ПРН10. Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати, типове для спеціальності обладнання.
 - ПРН23. Вміти розроблювати та обслуговувати бази даних для IT-інфраструктур.

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 1 (Programming I)
- Програмування 2 (Programming II)
- Структури даних та алгоритми (DS and AI)

Дисципліна є базовою для курсів:

- Інженерія програмного забезпечення. Курсова робота
- Організація баз даних.

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 збірки проектів (Ant, Maven, Gradle).

Розділ 3. Проектування ПЗ з використанням шаблонів.

Тема 3.1. Графічна нотація UML. Документування програмного коду.

Тема 3.2. Принципи розробки програмного забезпечення.

Тема 3.3. Проектування архітектури ПЗ.

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

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

Тема 3.6. Структурні шаблони проектування 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 проектування Prototype, Singleton, Factory Method.

Тема 3.11. Породжувальні шаблони проектування Abstract Factory, Builder.

Розділ 4. Проектування клієнт-серверної архітектури.

Тема 4.1. Клієнт-серверна архітектура.

Тема 4.2. Веб-сервер.

Тема 4.3. Паттерн проектування MVC.

Розділ 5. Проектування інтерфейсу користувача.

Тема 5.1. Призначення та типи інтерфейсів користувача.

Тема 5.2. Принципи побудови графічного інтерфейсу користувача.

Тема 5.3. Моделі подій. Моделі промальовування 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. Моделювання програмного забезпечення.

Тема 7.1. Методології моделювання SADT, IDEF, DFD, ELM, OOAD. Мови моделювання.

Тема 7.2. Інформаційне моделювання. Діаграми сутність-зв'язок, класів.

Тема 7.3. Моделювання бізнес-процесів, організацій та цілей.

Тема 7.4. Поведінкове моделювання. Діаграми станів, діяльності, взаємодії, послідовності, часові.

Розділ 8. Менеджмент програмних проектів.

Тема 8.1. Задачі управління проектами. Трикутник обмежень.

Тема 8.2. Управління ресурсами. Планування графіку виконання проекту.

Тема 8.3. Управління ризиками програмного проекту.

Тема 8.4. Контроль та моніторинг стану проекту. Метрики контролю.

Тема 8.5. Організація роботи проектної команди. Ролі та зони відповідальності учасників команди.

3. 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.

2. 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

3. С.В.Баран. Розробка програмного забезпечення з використанням патернів проектування: Навчальний посібник. – Кривий Ріг: Державний університет економіки і технологій, 2023. – 203 с.

4. 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

4.2. Додаткова література

1. 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

2. Design Patterns: elements of reusable object-oriented software / Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Indianapolis: - Addison-Wesley, 1994. -417 p. ISBN: 0201633612.

3. Head First. Java. Легкий для сприйняття довідник / Кетті Сьєрра, Берт Бейтс. - Харків: ВД «Фабула», 2022. – 720 с. ISBN 978-617-522-033-7

Інженерія програмного забезпечення: навч. посібник / Д. П. Кучеров, Є. Б. Артамонов. – К. : НАУ, 2017. – 388 с.

4. 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

5. Theo Mandel. The Elements of User Interface Design. - John Wiley & Sons, 2005. - 468 p. ISBN: 0471162671

6. Bruce Eckel. Thinking in Java, 4th ed. – Printice hall, 2006. – 1057 p. ISBN: 0131872486

7. Об'єктно-орієнтоване програмування: [Підручник] / В.В. Бублик. – К.: ІТ-книга, 2015. – 624 с.

8. Бандура В.В., Храбатин Р.І. Архітектура та проектування програмного забезпечення: конспект лекцій. — Івано-Франківськ: ІФНТУНГ, 2012. — 240 с.

9. Проектування інформаційних систем: Загальні питання теорії проектування 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.

10. Поморова О.В., Говорущенко Т.О. Проектування інтерфейсів користувача: навч.посібник - Хмельницький: ХНУ, 2011. - 206 с.

11. Проектування та моделювання програмного забезпечення сучасних інформаційних систем / Г. В. Табунщик, Т.І. Каплієнко, О.А. Петрова – Запоріжжя : Дике Поле, 2016. – 250 с.

12. 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.

4.3. Інформаційні ресурси

1. Дистанційні курси на платформі дистанційного навчання «Сікорський» в середовищі Google Workspace for Education: Інженерія програмного забезпечення.
<https://classroom.google.com/c/NzcwMzQzMTCxMjQy?cjc=mnkvrxaf>
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 54 години, практичний робіт 18 годин, самостійної роботи 78 годин.

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. Інтегровані середовища розробки ПЗ. Розширення функціональності IDE (додавання плагінів). Системи управління версіями документів, архітектурні особливості (CVS, SVN, Git).
3.	Автоматизація збірки ПЗ. Інструменти автоматизації зборки проектів (утиліта make, системи CMake, Ant, Maven, Gradle).

4.	Графічна нотація UML. Документування програмного коду. Уніфікована мова модулювання (UML). Діаграми UML. Засоби автоматизації UML моделювання. Документування за допомогою JavaDoc.
5.	Принципи розробки програмного забезпечення. Принципи DRY, KISS, YAGNI. SOLID принципи.
6.	Проектування архітектури ПЗ. Шаблони проектування ПЗ. Класифікація шаблонів проектування. Шаблон Composite: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Composite. Шаблон Decorator: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Decorator.
7.	Структурні шаблони проектування: Proxy та Flyweight. Шаблон Proxy: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Proxy. Шаблон Flyweight: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Flyweight.
8.	Структурні шаблони проектування: Adapter, Bridge та Facade. Шаблон Adapter: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Adapter. Шаблон Bridge: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Bridge. Шаблон Facade: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Facade.
9.	Шаблони поведінки: Iterator та Mediator. Призначення шаблонів поведінки. Шаблон Iterator: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Iterator. Шаблон Mediator: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Mediator.
10.	Шаблони поведінки: Observer та Strategy. Шаблон Observer: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Observer. Шаблон Strategy: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Strategy.
11.	Шаблони поведінки: State та Memento. Шаблон State: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону State. Шаблон Memento: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Memento.
12.	Шаблони поведінки: Command та Visitor. Шаблон Command: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Command. Шаблон Visitor: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Visitor.
13.	Шаблони поведінки: Interpreter та Chain of Responsibility. Шаблон Interpreter: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Interpreter. Шаблон Chain of

	Responsibility: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Chain of Responsibility.
14.	Породжувальні шаблони Prototype, Factory Method та Abstract Factory. Призначення шаблонів, що породжують. Шаблон Prototype: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Prototype. Шаблон Factory Method: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Factory Method. Шаблон Abstract Factory: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Abstract Factory.
15.	Шаблони Singleton та Builder. Шаблон Singleton: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Singleton. Шаблон Builder: мотивація, структура, учасники, відносини, шляхи застосування та результат використання. Приклад реалізації шаблону Builder.
16.	Клієнт-серверна архітектура. Модель OSI. Мережеві протоколи. Клієнт-серверна архітектура. Протокол http.
17.	Веб-сервер. Поняття веб-сервера. Сервлет. Контейнер сервлетів (Tomcat). Cookie та Session. Шаблонізатори (Thymeleaf).
18.	Паттерн проектування MVC. Архітектурний паттерн Model View Controller. Складові частини MVC. Взаємодія між компонентами. Застосування MVC у популярних фреймворках (Spring MVC).
19.	Графічний інтерфейс користувача. Призначення та типи інтерфейсів користувача. Принципи побудови графічного інтерфейсу користувача. Побудова графічного інтерфейсу користувача на основі Java Swing. Технологія JavaFX та її особливості.
20.	Якість та тестування ПЗ. Якість ПЗ. Метрики і стандарти якості ПЗ. Верифікація та валідація ПЗ. Тестування ПЗ.
21.	Автоматизація процесів тестування. Модульне тестування (unit testing, JUnit). Підхід TDD. Інструменти автоматизації тестування веб-додатків (Selenium). Паттерн PageObject. Тестування швидкодії (JMeter).
22.	Постійна інтеграція ПЗ Принципи постійної інтеграції ПЗ. Continuous Integration. Continuous Delivery. Налаштування CI/CD процесів (GitHub Actions, Jenkins).
23.	Інформаційне моделювання та моделювання бізнес-процесів Методології моделювання SADT, IDEF, DFD, ELM, OOAD. Мови моделювання. Інформаційне моделювання. Діаграми сутність-зв'язок, класів. Моделювання бізнес-процесів, організацій та цілей.
24.	Управління проектами Задачі управління проектами. Трикутник обмежень. Управління змістом та якістю проекту.
25.	Контроль та моніторинг стану проекту Контроль та моніторинг стану проекту. Метрики контролю. Організація роботи проектною команди. Ролі та зони відповідальності учасників команди.

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 проектування, 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 проектами.

Практична робота включає:

1. розробку та аналіз математичного алгоритму рішення задачі;
2. декомпозицію задачі;
3. застосування шаблонів проектування;
4. роздруківку програми;
5. UML діаграми взаємодії класів;
6. результати виконання програми;
7. документування практичної роботи та зробленої програми.

Таблиця 2

№ з/п	Назва практичної роботи (комп'ютерного практикуму)	Кількість ауд. годин
1.	Проект програмного додатку. Збірка проекту за допомогою Ant.	3
2.	Графічна нотація UML. Документування за допомогою JavaDoc.	3
3.	Структурні шаблони проектування ПЗ. Composite, Decorator, Proxy. Flyweight, Adapter, Bridge, Facade.	3
4.	Шаблони поведінки. Iterator, Mediator, Observer, Strategy, Chain of Responsibility.	3
5.	Шаблони поведінки. Visitor, Memento, State, Command, Interpreter.	3
6.	Породжувальні шаблони. Prototype, Singleton, Factory Method, Abstract Factory, Builder.	3
	Разом:	18

6. Самостійна робота здобувача вищої освіти

Види самостійної роботи для студентів денної форми навчання (78 годин):

- підготовка до аудиторних занять, виконання поточних домашніх завдань та опрацювання матеріалів лекцій (0,5 годин x 54 годин = 26 годин);
- виконання індивідуального завдання до практичних робіт, розв'язок задач, оформлення протоколу, підготовка та оброблення проведення розрахунків за первинними даними, отриманими на практичних заняттях, оформлення звіту до практичної роботи (1 година x 18 годин практичних занять (6 практичних робіт) = 18 годин);
- підготовка та виконання поточної контрольної роботи (2 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 (24 години).

Таблиця 3

№ з/п	Назва теми, що виноситься на самостійне опрацювання	Кількість годин
-------	---	-----------------

1.	Установка і використання систем управління проектами. Установка і конфігурування SVN	1
2.	Установка і конфігурування систем збірки проектів Maven та Gradle	1
3.	Встановлення та налаштування засобів автоматизації UML моделювання.	1
4.	Шаблони проектування GRASP. Реалізації Composite та Decorator в класах Java SDK	1
5.	Реалізації Proxy та Flyweight в класах Java SDK	1
6.	Реалізації Adapter, Bridge та Facade в класах Java SDK	1
7.	Реалізації Iterator та Mediator в класах Java SDK	1
8.	Реалізації Observer та Strategy в класах Java SDK	1
9.	Реалізації State та Memento в класах Java SDK	1
10.	Реалізації Command та Visitor в класах Java SDK	1
11.	Реалізації Interpreter та Chain of Responsibility в класах Java SDK	1
12.	Реалізації Prototype, Factory Method та Abstract Factory в класах Java SDK	1
13.	Реалізації Singleton та Builder в класах Java SDK	1
14.	Використання класів Java для роботи з протоколами зв'язку, такими як HTTP і HTTPS.	1
15.	Паттерн Front Controller.	1
16.	Паттерн проектування MVC. Реалізація MTV у Django	2
17.	Фреймовий та віконний інтерфейси користувача. Реалізація легких компонентів на прикладі SWING	1
18.	Сертифікація ПЗ в Україні.	1
19.	Автоматизована ініціалізація елементів сторінки за допомогою PageFactory.	1
20.	Установка і використання сервера постійної інтеграції	1
21.	Rational Unified Process	1
22.	Засоби автоматизації управління проектами	1
23.	Робота команди з гнучкою методологією	1
	Разом	24

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Для виконання практичних робіт та поточної контрольної роботи встановлюються дедлайни. Виконання практичних робіт поза встановлених термінів супроводжуються штрафними балами, які вираховуються із оцінки за протокол. В разі бажання студента здати ще одну роботу, студент має стати в кінець черги. За одне заняття можна здати лише 2 роботи. Для зарахування практичної

роботи студент повинен виконати індивідуальне завдання і оформити його у вигляді протоколу. Першим етапом студент захищає результати отримані під час виконання індивідуального завдання до практичної роботи, на другому етапі – захищає теорію шляхом усного опитування або тестування. Бали отримані за виконання практичної роботи, за усне опитування (або тест) та за протокол входять в оцінку за практичну роботу. Виконання всіх практичних робіт є обов’язковими для допуску до семестрового екзамену. Поточна контрольна робота виконується самостійно за індивідуальним завданням та не приймається поза встановлені терміни. Контрольна робота пишеться на лекційному занятті без застосування допоміжних засобів (мобільні телефони, планшети та ін.); результат пересилається до відповідної директорії Google-диску через Google форму.

Студент допускається до виконання поточної контрольної роботи за умови здачі всіх практичних робіт, необхідних для допуску.

Бали за екзаменаційну роботу додаються до балів за практичні роботи та контрольну роботу і складають семестрову рейтингову оцінку.

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.

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

Семестровий рейтинг студента з дисципліни розраховується, виходячи із 100-бальної шкали. Семестровий рейтинг складається із стартової (поточної) оцінки RS та екзаменаційної RE. Стартовий рейтинг складається з балів, що студент отримує за виконання 9 практичних робіт РП, 2 поточних контрольних робіт РПКР.

Максимальна кількість балів за практичні роботи складає 60 балів, тобто РП = 60. Критерії оцінювання практичних робіт наступні:

- своєчасність підготовки протоколу до практичного заняття, повнота виконання теоретичного або практичного завдання в протоколі: 0 – 2 бали (із оцінки за протокол можуть бути нарахovanі штрафні бали за несвоєчасну здачу практичної роботи від 0 – 2 балів);
- коректність функціонування розроблених моделей на програмному забезпеченні, пояснення написаного коду та демонстрація роботи програми: 0 – 2(4) бали;
- опитування/тестування за тематикою практичної роботи, захист одержаних в роботі результатів, відповіді на додаткові теоретичні питання викладача: 0 – 2(3) бали.

Максимальна кількість балів за поточну КР: **РПКР = 10 балів.**

Оцінка за поточну КР знижується за:

- некоректне оформлення схем та креслень, відсутність розрядності регістрів, шин, недотримання ГОСТів;
- відсутність коментарів в програмному коді та оформленні алгоритмів;
- відсутність коментарів та пояснень під час розрахунків.

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	2	6	10

	Виконання завдання	4		
	Опитування/тест	4		
Практична робота 2	Протокол	2	6	10
	Виконання завдання	4		
	Опитування/тест	4		
Практична робота 3	Протокол	2	6	10
	Виконання завдання	4		
	Опитування/тест	4		
Практична робота 4	Протокол	2	6	10
	Виконання завдання	4		
	Опитування/тест	4		
Практична робота 5	Протокол	2	6	10
	Виконання завдання	4		
	Опитування/тест	4		
Практична робота 6	Протокол	2	6	10
	Виконання завдання	4		
	Опитування/тест	4		
Поточна контрольна робота 1		10	6	10
Поточна контрольна робота 2		10	6	10
Всього балів		80	51	80

Максимальна кількість балів за екзамен дорівнює **RE = 20 балів**.

Екзаменаційний білет містить тестові завдання (теоретичного та практичного характеру) та практичне завдання за тематикою лекцій та практичних робіт, що виконувались протягом семестра.

Календарна атестація студентів (на 7 та 13 тижнях семестрів) 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 «не атестований».

Кількість балів, що отримує студент за семестр визначається формулою

$$R = R_{\text{ПР}} + R_{\text{ПКР}} + R_E = R_S + R_E$$

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

З урахуванням одержаної суми балів кінцева оцінка визначається наступною таблицею:

Таблиця 5

<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
Не виконані умови допуску	Не допущено

9. Нормативні положення політики навчальної дисципліни (освітнього компонента)

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 платформі Google Meet або Zoom. Посилання на конференцію видається на початку семестру.

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, проходиться 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; 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; 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>).

Робочу програму навчальної дисципліни (силабус):

Складено, старший викладач кафедри ОТ Васильєва Марія Давидівна.

Ухвалено кафедрою обчислювальної техніки (протокол No 12 від 23.06.2025).

Погоджено методичною комісією ФІОТ (протокол No 11 від 27.06.2025).