



СУЧАСНІ ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ ТА КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Третій (доктор філософії)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>122 Комп'ютерні науки</i>
Освітня програма	<i>КОМП'ЮТЕРНІ НАУКИ</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>Очна (денна)</i>
Рік підготовки, семестр	<i>3 курс, осінній семестр</i>
Обсяг дисципліни	<i>4 кредити (120 годин)</i>
Семестровий контроль/ контрольні заходи	<i>Екзамен</i>
Розклад занять	<i>Перший та другий тиждень: Лекція/лабораторна робота, четвер, 10-25, on-line</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	Лектор: <i>к.т.н. Кравцов Григорій Олексійович, контактні дані: hryhoriy.kravtsov@gmail.com</i> Практичні : <i>к.т.н. Кравцов Григорій Олексійович, контактні дані: hryhoriy.kravtsov@gmail.com</i>
Розміщення курсу	

Програма навчальної дисципліни

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

Дисципліна «Сучасні технології програмування та комп'ютерного моделювання» (ІОЗ) є нормативною дисципліною навчального плану підготовки докторів філософії з спеціальності «комп'ютерні науки» і грає важливу роль у підготовці фахівців.

Метою навчальної дисципліни є формування у аспірантів компетентностей та підготовка науковця, здатного вирішувати складні задачі і практичні проблеми використання новітніх технологій та засобів проектування інформаційних систем та здійснювати професійну діяльність з моделювання та проектування інформаційних систем.

Метою кредитного модуля є формування у аспірантів загальних та спеціальних (фахових, предметних) компетентностей:

- ЗК01 - Здатність до абстрактного мислення, аналізу та синтезу.
- ЗК02 - Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
- ЗК03 - Здатність працювати в міжнародному контексті.
- ЗК04 - Здатність розв'язувати комплексні проблеми комп'ютерних наук на основі

системного наукового світогляду та загального культурного кругозору із дотриманням принципів професійної етики та академічної доброчесності.

– СК01 - Здатність виконувати оригінальні дослідження, досягати наукових результатів, які створюють нові знання у комп'ютерних науках та дотичних до них міждисциплінарних напрямках і можуть бути опубліковані у провідних наукових виданнях з комп'ютерних наук та суміжних галузей.

– СК02 - Здатність застосовувати сучасні методології, методи та інструменти експериментальних і теоретичних досліджень у сфері комп'ютерних наук, сучасні цифрові технології, бази даних та інші електронні ресурси у науковій та освітній діяльності.

– СК03 - Здатність виявляти, ставити та вирішувати дослідницькі науково-прикладні задачі та/або проблеми в сфері комп'ютерних наук, оцінювати та забезпечувати якість виконуваних досліджень.

– СК04 - Здатність ініціювати, розробляти і реалізовувати комплексні інноваційні проекти у галузі комп'ютерних наук та дотичні до неї міждисциплінарних проектах, демонструвати лідерство під час їх реалізації.

– СК05 - Здатність здійснювати науково-педагогічну діяльність у вищій освіті у сфері комп'ютерних наук.

– СК06 - Здатність аналізувати та оцінювати сучасний стан і тенденції розвитку комп'ютерних наук та інформаційних технологій.

Програмні результати навчання:

Згідно з вимогами програми навчальної дисципліни аспіранти після засвоєння кредитного модуля мають продемонструвати такі результати навчання:

– ПРН01. Мати передові концептуальні та методологічні знання з комп'ютерних наук і на межі предметних галузей, а також дослідницькі навички, достатні для проведення наукових і прикладних досліджень на рівні останніх світових досягнень з відповідного напрямку, отримання нових знань та/або здійснення інновацій.

– ПРН02. Вільно презентувати та обговорювати з фахівцями і нефахівцями результати досліджень, наукові та прикладні проблеми комп'ютерних наук державною та іноземною мовами, оприлюднювати результати досліджень у наукових публікаціях у провідних міжнародних наукових виданнях.

– ПРН03. Формулювати і перевіряти гіпотези; використовувати для обґрунтування висновків належні докази, зокрема, результати теоретичного аналізу, експериментальних досліджень і математичного та/або комп'ютерного моделювання, наявні літературні дані.

– ПРН04. Розробляти та досліджувати концептуальні, математичні і комп'ютерні моделі процесів і систем, ефективно використовувати їх для отримання нових знань та/або створення інноваційних продуктів у комп'ютерних науках та дотичних міждисциплінарних напрямках.

– ПРН05. Планувати і виконувати експериментальні та/або теоретичні дослідження з комп'ютерних наук та дотичних міждисциплінарних напрямків з використанням сучасних інструментів, критично аналізувати результати власних досліджень і результати інших дослідників у контексті усього комплексу сучасних знань щодо досліджуваної проблеми.

– ПРН06. Застосовувати сучасні інструменти і технології пошуку, оброблення та аналізу інформації, зокрема, статистичні методи аналізу даних великого обсягу та/або складної структури, спеціалізовані бази даних та інформаційні системи.

– ПРН07. Розробляти та реалізовувати наукові та/або інноваційні інженерні проекти, які дають можливість переосмислити наявне та створити нове цілісне знання та/або професійну практику і розв'язувати значущі наукові та технологічні проблеми комп'ютерної науки з

дотриманням норм академічної етики і врахуванням соціальних, економічних, екологічних та правових аспектів.

– ПРН08. Визначати актуальні наукові та практичні проблеми у сфері комп'ютерних наук, глибоко розуміти загальні принципи та методи комп'ютерних наук, а також методологію наукових досліджень, застосувати їх у власних дослідженнях у сфері комп'ютерних наук та у викладацькій практиці.

– ПРН09. Вивчати, узагальнювати та впроваджувати в навчальний процес інновації комп'ютерних наук.

– ПРН10. Відшукувати, оцінювати та критично аналізувати інформацію щодо поточного стану та трендів розвитку, інструментів та методів досліджень, наукових та інноваційних проєктів з комп'ютерних наук.

– ПРН11. Організувати і здійснювати освітній процес у сфері комп'ютерних наук, його наукове, навчально-методичне та нормативне забезпечення, застосувати ефективні методики викладання навчальних дисциплін.

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

Пререквізити: здатність застосовувати знання у практичних ситуаціях, навички комп'ютерного моделювання та використання інформаційних і комунікаційних технологій. Цей курс базується на таких дисциплінах:

- Сучасні проблеми і тенденції розвитку комп'ютерних наук та інформаційних технологій.
- Додаткові розділи вищої математики.

Постреквізити: Перелік напрямків діяльності, що забезпечуються: педагогічна практика, захист дисертаційної роботи.

3. Зміст навчальної дисципліни

Розділ 1. Вступ до дисципліни.

Тема 1.1. Вступ до проєктування інформаційних систем.

Тема 1.2. Огляд інструментів та технологій проєктування.

Тема 1.3. Вибір інструментів для проєктування.

Тема 1.4. Інтеграція інструментів у процесі розробки.

Розділ 2. Моделювання та проєктування інформаційних систем. Інструментарій, моделі, підходи, технології.

2.1 Методології проєктування.

2.2. Архітектурні підходи.

2.3. Мови моделювання та їх використання.

2.4. Технології для архітектурного проєктування.

2.5. Інструменти для дизайну інтерфейсу користувача.

2.6. Інструменти для проєктування баз даних.

Модуль 3: Інструментарій для програмування та розробки інформаційних систем.

3.1. Вибір мов програмування та фреймворків.

3.2. Інструменти розробки та оточення.

3.3. Інструментарій для Web-розробки.

3.4. Інструменти для збору та управління вимогами.

3.5. Інтеграція та обмін даними між системами.

3.6. Інструменти для автоматизованого тестування.

3.7. Техніки та інструменти для тестування продуктивності.

3.8. Засоби версійного контролю.

3.9. Автоматизована збірка та розгортання.

4. Навчальні матеріали та ресурси

Базова

1. Мінухін С. В. Методи і моделі проектування на основі сучасних CASE–засобів. Навчальний посібник / С.В. Мінухін, О.М. Беседовський, С. В. Знахур. – Харків: Вид. ХНЕУ, 2008. – 272 с. (укр. мов.)
2. Xu A. System Design Interview – An insider's guide. - Independentlypublished, 2020. – 320 с.
3. Klepmann M. Designing Data-Intensive Applications. - O'ReillyMedia, 2017. – 590 с.
4. Авраменко В.С., Авраменко А.С. Проектування інформаційних систем: навчальний посібник / В.С. Авраменко, А.С. Авраменко. – Черкаси: Черкаський національний університет ім. Б. Хмельницького, 2017. – 434 с.: іл.
5. Навчальна платформа Coursera, Secure Software Design [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.coursera.org/programs/program-natsional-nii-tiekhnichnii-univiersitiet-ukrayini-kiyivs-kii/specializations/secure-software-design>.
6. Навчальна платформа Coursera, Information Systems [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.coursera.org/programs/program-natsional-nii-tiekhnichnii-univiersitiet-ukrayini-kiyivs-kii/specializations/information-systems>.
7. Навчальна платформа Udemy, Software Architecture & System Design Practical Case Studies [Електронний ресурс]. – Режим доступу до ресурсу: <https://ua.udemy.com/course/software-architecture-system-design-practical-case-studies>.
8. Навчальна платформа Udemy, Software Architecture & Design of Modern Large Scale Systems [Електронний ресурс]. – Режим доступу до ресурсу: <https://ua.udemy.com/course/software-architecture-design-of-modern-large-scale-systems/>.
9. Навчальна платформа Udemy, Software Architecture & Technology of Large-Scale Systems [Електронний ресурс]. – Режим доступу до ресурсу: <https://ua.udemy.com/course/developer-to-architect/>.
10. Навчальна платформа Wintellectnow, Architecting Distributed CloudApplications [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.wintellectnow.com/Home/SeriesDetail?seriesId=architecting-distributed-cloud-applications>.
11. Навчальна платформа Pluralsight.com, DevOps: TheBig Picture [Електронний ресурс]. – Режим доступу до ресурсу: <https://app.pluralsight.com/library/courses/devops-big-picture/table-of-contents>.
12. Навчальна платформа Pluralsight.com, Implementing DevOps intheRealWorld [Електронний ресурс]. – Режим доступу до ресурсу: <https://app.pluralsight.com/library/courses/implementing-devops-real-world/table-of-contents>.

Додаткова

13. Richards M., Ford N. Fundamentals of Software Architecture: A Comprehensive Guide to Patterns, Characteristics, and Best Practices. - O'Reilly Media, 2020. – 396 с.
14. Wixom D., Roth R. System Analysis and Design. - John Wiley&Sons, Inc., 2012. – 596 с.

Навчальний контент

5. Методика опанування навчальної дисципліни(освітнього компонента)

№ з/п	Назва теми лекції та перелік основних питань (дидактичні матеріали: презентація PowerPoint)
-------	---

1	2
1	<p align="center">Розділ 1. Вступ до дисципліни.</p> <p>Лекція 1. Огляд інструментів та технологій проектування. Основні концепції та терміни. Роль і значення інформаційних систем в організаціях. Основні етапи життєвого циклу інформаційних систем. Роль інструментів у процесі проектування. Огляд CASE-інструментів. Визначення та роль CASE-інструментів в процесі розробки. Популярні CASE-інструменти: VisualParadigm, SparxSystemsEnterpriseArchitect. Функціональність CASE-інструментів. Моделювання UML-діаграм. Підтримка процесів аналізу та проектування в CASE-середовищі. Інструменти для моделювання процесів. Огляд інструментів для моделювання процесів: Lucidchart, Draw.io, StarUML. Практичні аспекти вибору інструментів. Спільна робота команди. Важливість інтеграції інструментів для колективної розробки. Порівняння та аналіз інструментів для вибору оптимального варіанту. Література: 1-3. Самостійна робота: Ознайомлення з CASE-інструментами для проектування інформаційних систем. Підготовка до лабораторної роботи.</p>
	<p>Розділ 2. Моделювання та проектування інформаційних систем. Інструментарій, моделі, підходи, технології.</p>
2	<p>Лекція 2. Методології проектування. Архітектурні підходи. Мови моделювання та їх використання. Методології проектування. Водопадна модель. Методології гнучкого розроблення. Основні принципи Agile. Характеристики Scrum та Kanban. Вибір методології в залежності від проекту. Методології спрямовані на об'єктно-орієнтоване проектування. Моделі керування проектами (PMI, PRINCE2). Вибір моделі керування проектом залежно від типу та масштабу проекту. Архітектурні підходи. Монолітні та розподілені архітектури. Мікросервісна архітектура. Клієнт-серверна архітектура. Мови моделювання та їх використання. UML. BPMN. Література: 1-3. Самостійна робота: Створення моделей бізнес-процесів інформаційної системи для своєї дисертації. Підготовка до лабораторної роботи.</p>
3	<p>Лекція 3. Технології для архітектурного проектування. ArchiMate для архітектурного моделювання. Огляд ArchiMate. Призначення та основні концепції. Види діаграм та їх призначення. Використання ArchiMate в архітектурному моделюванні. TOGAF та його роль в архітектурному проектуванні. Огляд TOGAF. Мета та завдання TOGAF. Структура TOGAF Architecture Development Method (ADM). Роль TOGAF в архітектурному проектуванні. Забезпечення стандартизації та високої якості архітектурних рішень. Інтеграція TOGAF в процес розробки програмного забезпечення. Література: 1-4. Самостійна робота: Використання архітектурного моделювання при підготовці дисертаційної роботи. Підготовка до лабораторної роботи</p>
4	<p>Лекція 4. Інструменти для дизайну інтерфейсу користувача. Інструменти для дизайну інтерфейсу користувача. Sketch та Figma для UX/UI дизайну. Огляд Sketch та його використання. Інтерфейс та основні функції Sketch. Створення дизайну інтерфейсу за допомогою Sketch. Figma: Інтерактивний UX/UI дизайн. Переваги та особливості використання Figma. Спільна робота та взаємодія в команді. Інтеграція дизайну та розробки. Оптимальне використання інструментів. Підвищення ефективності через інтеграцію Sketch та Figma з розробкою. Взаємодія дизайнерів та розробників у процесі проектування. Література: 2, 5-6. Самостійна робота: використання інструментальних засобів дизайну інтерфейсу користувача при розробленні інформаційної системи. Підготовка до лабораторної роботи</p>
5	<p>Лекція 5. Інструменти для проектування баз даних. Сучасні інструменти для моделювання ER-діаграми. Lucidchart, Draw.io, SmartUML, PlantUML. UML-діаграми для баз даних. Застосування UML-діаграм для моделювання баз</p>

	<p>даних. Інструменти підтримки: VisualParadigm, SparxSystemsEnterpriseArchitect. Інструменти для нормалізації. Використання DbSchema для визначення нормальних форм. Ручна та автоматична генерація SQL-скриптів. Аналіз та нормалізація баз даних в NavicatDataModeler. Вбудовані функції для аналізу та нормалізації. Інтеграція інструментів у процесі розробки. Інтеграція баз даних у розробку. MySQLWorkbench. Експорт та імпорт моделей баз даних. Підтримка спільної роботи над проектами. Oracle SQL Developer. Інтеграція з іншими інструментами для обміну моделями баз даних.</p> <p>Література: 1-9.</p> <p>Самостійна робота: використання інструментальних засобів проєктування баз даних при розробленні інформаційної системи. Підготовка до лабораторної роботи</p>
	<p>Розділ 3. Інструментарій для програмування та розробки інформаційних систем.</p>
6	<p>Лекція 6. Оцінка вимог проєкту. Вибір IDE. Інструментарій для Web-розробки.</p> <p>Обґрунтування вибору мови програмування. Оцінка вимог проєкту. Адаптація мови до функціональних вимог. Швидкість та продуктивність. Підтримка бібліотек та фреймворків. Тип проєкту та його складність. Вибір мови з урахуванням типу застосунка та його розміру. Спільнота та екосистема. Фреймворки для розробки веб-застосунків та мобільних застосунків. Front-end фреймворки: Angular, React, Angular, Vue.js - порівняння та вибір. Back-end фреймворки: Play, Django, Flask, RubyonRails - характеристики та використання. Інструменти розробки та оточення. IDE. Огляд найпопулярніших IDE для різних мов програмування: VisualStudioCode, IntelliJ IDEA, PyCharm. Адаптація до вимог розробки.</p> <p>Системи автоматизованої збірки (CI/CD). Інструментарій для Web-розробки. Технології веб-серверів. Вибір та конфігурація веб-сервера: Apache, Nginx - порівняння та використання.</p> <p>Література: 6-9.</p> <p>Самостійна робота: використання IDE при проєктуванні інформаційних систем. Підготовка до лабораторної роботи</p>
7	<p>Лекція 7. Інструменти для збору та управління вимогами. Інтеграція та обмін даними між системами. Інструменти для автоматизованого тестування. Техніки та інструменти для тестування продуктивності.</p> <p>Інструменти для збору та управління вимогами. Огляд популярних систем управління проєктами. Jira. Функціонал та можливості для ефективного управління вимогами. Trello. Принцип роботи та використання для невеликих проєктів. Redmine. Інтеграція та основні функції для збору вимог. Техніки визначення та документування вимог. Методології збору вимог. Експертні інтерв'ю, анкетування, спостереження. Використання кейсів та сценаріїв для документування. Інтеграція та обмін даними між системами. ETL-інструменти (Extract, Transform, Load). Огляд основних ETL-інструментів: Informatica, Microsoft SSIS, ApacheNifi. Застосування в реальних проєктах для обміну даними. RESTful API та веб-служби. Інструменти для автоматизованого тестування Selenium. Функціонал та використання для автоматизації тестів JUnit та TestNG. Основні функції та переваги використання в юніт-тестуванні. Інтеграція тестування у CI/CD. Організація автоматичного тестування в процесі CI/CD. Jenkins, GitLab CI - інтеграція та автоматизація тестового процесу. Техніки та інструменти для тестування продуктивності Apache JMeter. Створення та виконання тестів для оцінки продуктивності LoadRunner. Визначення навантаження та оцінка реакції системи. Аналіз продуктивності за допомогою Profilers.</p> <p>Література: 6-9.</p> <p>Самостійна робота: використання інструментів автоматизованого тестування та CI/CD. Підготовка до лабораторної роботи.</p>
8	<p>Лекція 8. Засоби версійного контролю.</p> <p>Засоби версійного контролю. Огляд систем версійного контролю. Git. Основні концепції та переваги використання. Відгалуження та злиття гілок в Git. А</p> <p>Література: 9-12.</p> <p>Самостійна робота: використання інструментарію CI/CD для розгортання інформаційної</p>

	системи при підготовці дисертаційної роботи. Підготовка до лабораторної роботи
9	Лекція 9. Інструментарій автоматизованої збірки та розгортання. Автоматизована збірка та розгортання. Інтеграція засобів автоматизованої збірки в процес розробки. Jenkins. Налаштування та використання для автоматизації збірки. Travis CI. Інтеграція з GitHub та автоматизація тестування. GitLab CI. Використання в інтегрованому середовищі GitLab. Деплоймент у хмарний сервіс Heroku.

Лабораторні заняття

№ з/п	Назва лабораторної роботи (комп'ютерного практикуму)	Кількість ауд. годин
1	Лабораторна робота 1. Використання StarUML для створення моделі інформаційної системи в нотації UML. Література: 1-3.	2
2	Лабораторна робота 2. Використання LucidChart для створення моделі бізнес-процесів інформаційної системи в нотації BPMN. Література: 1-3.	2
3	Лабораторна робота 3. Використання Archi для створення архітектури проєкту інформаційної системи в парадигмі ArchiMate. Література: 1-4.	2
4	Лабораторна робота 4. Використання Figma для створення прототипу інтерфейсу користувача для інформаційної системи. Література: 2, 5-6.	2
5	Лабораторна робота 5. Використання MySQL Workbench для створення моделі бази даних для інформаційної системи. Література: 1-9.	2
6	Лабораторна робота 6. Огляд сучасних IDE для розробки програмного забезпечення інформаційних систем. Література: 6-9.	2
7	Лабораторна робота 7. Використання систем управління проєктами Jirama Trello для створення проєкту інформаційної системи. Література: 6-9.	2
8	Лабораторна робота 8. Використання Git для управління версійністю програмного забезпечення при створенні проєкту інформаційної системи. Література: 9-12.	2
9	Лабораторна робота 9. Використання BurpSuite для визначення безпеки коду проєкту інформаційної системи. Література: 5-7.	2

6. Самостійна робота аспіранта

№ з/п	Назви тем і питань, що виносяться на самостійне опрацювання та посилання на навчальну літературу	Кількість годин СРС
-------	--	---------------------

1	2	2
1	Методології гнучкого розроблення[1-3]	8
2	Мови моделювання та їх використання [1-3]	8
3	Технології для архітектурного проєктування [1-4]	8
4	Інструменти для дизайну інтерфейсу користувача [2, 5-6]	8
5	Інструменти для проєктування баз даних [1-9]	8
6	Інструменти для збору та управління вимогами [6-9]	8
7	Інструменти для автоматизованого тестування [1-11]	12
8	Автоматизована збірка та розгортання[9-12]	12
9	OWASP ZAP, BurpSuite[5-7]	12

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Відвідування занять

Рекомендоване відвідування лекцій, але штрафних санкцій за пропуски лекцій не передбачено. Відвідування занять комп'ютерного практикуму необхідно в обсязі, достатньому для виконання вимог викладача щодо виконання і своєчасної здачі лабораторних робіт та індивідуального завдання.

Пропущені контрольні заходи

Лабораторні роботи можна здавати у відведений за розкладом час лабораторних занять, як до, так і після встановленого терміну здачі конкретної роботи. Додаткові години для здачі індивідуального завдання призначаються викладачем в межах часу лабораторних занять. За відсутності поважних причин пропуску (медична довідка тощо) штрафні бали не нараховуються.

Процедура оскарження результатів контрольних заходів

Здобувачі мають можливість підняти будь-яке питання, яке стосується процедури контрольних заходів та очікувати, що воно буде розглянуто згідно із наперед визначеними процедурами.

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

Поточний контроль: виконання індивідуального завдання на проєктування інформаційних систем, виконання комп'ютерних практикумів.

Календарний контроль: проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

Семестровий контроль: залік.

Умови допуску до семестрового контролю: мінімально позитивна оцінка за індивідуальне завдання /зарахування усіх практичних робіт/семестровий рейтинг більше 30 балів.

Рейтинг аспіранта з дисципліни складається з двох складових: стартової – призначена для оцінювання заходів поточного контролю впродовж семестру та екзаменаційної – призначена для оцінювання окремих запитань (завдань) на екзамені і формується з балів, що він отримує за:

- 1) індивідуальне завдання на проєктування інформаційної системи;
- 2) виконання комп'ютерних практикумів;
- 3) відповідь на екзамені.

1. Індивідуальне завдання

Ваговий бал – 25. Кожний аспірант виконує індивідуальне завдання щодо розробки інформаційної системи (програмного забезпечення) в рамках свого дисертаційного дослідження, яке передбачає використання всієї низки програмно-технічного інструментарію, що вивчається в рамках курсу.

2. Робота на комп'ютерних практичних заняттях

Ваговий бал – 5. Максимальна кількість балів на всіх практичних заняттях дорівнює $5 \cdot 9 = 45$ балів.

3. Відповідь на заліку

Кількість балів по відповіді на кожне питання визначається викладачем з врахуванням складності питання та якості відповіді. Максимальна кількість балів 30.

Штрафні та заохочувальні бали за:

- виконання завдань із удосконалення дидактичних матеріалів з дисципліни надається від 2 до 5 заохочувальних балів.

Розрахунок шкали (R) рейтингу

Сума вагових балів контрольних заходів протягом семестру складає:

$$RC = 70 + 30 = 100 \text{ балів.}$$

Для отримання аспірантом відповідних оцінок (ECTS та традиційних) його рейтингова оцінка *R* переводиться згідно з таблицею відповідності рейтингових балів оцінкам за університетською шкалою.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

<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. Додаткова інформація з дисципліни (освітнього компонента)

- можливість зарахування сертифікатів проходження дистанційних чи онлайн курсів за тематикою дисципліни «Сучасні технології програмування та комп'ютерного моделювання»;
- інша інформація для аспірантів щодо особливостей опанування навчальної дисципліни:

Застосовуються стратегії активного і колективного навчання, які визначаються наступними методами і технологіями:

- 1) кредитно-модульна технологія навчання;
- 2) особистісно-орієнтовані (розвиваючі) технології, засновані на активних формах і методах навчання («аналіз ситуацій» ділові, імітаційні ігри, дискусія, експрес-конференція, навчальні дебати.);
- 3) інформаційно-комунікаційні технології, що забезпечують проблемно-дослідницький характер процесу навчання та активізацію самостійної роботи аспірантів (електронні презентації для лекційних занять, використання аудіо-, відео-підтримки навчальних занять, розробка і застосування на основі комп'ютерних і мультимедійних засобів творчих завдань, доповнення традиційних навчальних занять засобами взаємодії на основі мережевих комунікаційних можливостей).

Робочу програму навчальної дисципліни (силабус):

Складено к.т.н. Кравцовим Григорієм Олексійовичем

Ухвалено: Вченою радою ІПМЕ ім. Г.Є. Пухова НАН України (протокол №10 від 26.09.2024 р.)