



## РОЗРОБКА І ТЕСТУВАННЯ ПРОГРАМ

### Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 – Інформаційні технології
Спеціальність	122 Комп'ютерні науки, 124 Системний аналіз
Освітня програма	Системи і методи штучного інтелекту, Системний аналіз і управління
Статус дисципліни	вибіркова
Форма навчання	очна (денна)
Рік підготовки, семестр	3 рік, 5-й семестр (осінній)
Обсяг дисципліни	120 годин / 4 кредити ЄКТС (лекції – 36 год., практичні заняття – 18 год., СРС – 66 год.)
Семестровий контроль/ контрольні заходи	залік, фронтальні опитування, МКР
Розклад занять	<a href="http://rozklad.kpi.ua/">http://rozklad.kpi.ua/</a>
Мова викладання	українська
Інформація про керівника курсу / викладачів	лекції та практичні проводить: канд. техн. наук, доцент, доцент кафедри ШІ Бендюг Владислав Іванович , bendiuh.vladyslav@iit.kpi.ua
Розміщення курсу	<a href="https://classroom.google.com/c/NjlxNzEzMzM5MzU4?cjc=ota2yzy">https://classroom.google.com/c/NjlxNzEzMzM5MzU4?cjc=ota2yzy</a> Код класу <b>ota2yzy</b>

### Програма навчальної дисципліни

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

Мета дисципліни	Метою опанування дисципліни є формування відповідного рівня знань і досвіду застосування методологій розробки та керування життєвим циклом програмного забезпечення, а також основ та підходів до тестування програмного забезпечення.
Компетентності	Дисципліна сприяє формуванню у студентів таких компетентностей: <ul style="list-style-type: none"><li>• Здатність до абстрактного мислення, аналізу та синтезу;</li><li>• Здатність застосовувати знання у практичних ситуаціях;</li><li>• Здатність вчитися і оволодівати сучасними знаннями;</li><li>• Здатність проводити бізнес-аналіз з використанням сучасних технологій бізнес-інтелекту, аналітичних інструментів та управляти ІТ-проектами.</li><li>• Здатність проектувати, розробляти та тестувати програмне забезпечення із використанням різних парадигм програмування: структурного, об'єктно-орієнтованого, функціонального за</li></ul>

	відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління ними.
<b>Результати навчання</b>	<p><b>Результати</b> навчання охоплюють</p> <ul style="list-style-type: none"> <li>• ознайомлення з принципами розробки ПЗ;</li> <li>• ознайомлення з моделями розробки та життєвим циклом ПЗ;</li> <li>• ознайомлення з моделями гнучкої розробки ПЗ;</li> <li>• ознайомлення з принципами тестування ПЗ та класифікацією методів тестування;</li> <li>• вміння формувати якісні вимоги до створення ПЗ та проводити тестування вимог;</li> <li>• знання класифікацій тестування ПЗ та ознайомлення з техніками тестування;</li> <li>• ознайомлення з чек-листами та тест-кейсами та принципами створення якісних чек-листів та тест-кейсів;</li> <li>• вміння формувати звіти про дефекти та знання життєвого циклу дефекту ПЗ;</li> <li>• ознайомлення з принципами планування та формування звітності при тестуванні ПЗ;</li> <li>• ознайомлення з основами автоматизованого тестування.</li> </ul>
<b>Інформаційне забезпечення</b>	силабус, рейтингова система оцінювання, презентаційні та відеоматеріали до практичних занять, презентаційні та відеоматеріали до лекцій, Google Клас дисципліни на платформі дистанційного навчання Сікорський

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

Вивчення дисципліни базується на знанні студентами основних понять програмування, математики, економіки й спрямоване на вироблення в них навичок системного підходу до вирішення задач проектування та створення програмного забезпечення, знання принципів, підходів та життєвого циклу тестування програмного забезпечення.

### Попередні дисципліни та розділи

- Алгоритми і структури даних
- Програмування та алгоритмічні мови
- Об'єктно-орієнтоване програмування
- Операційні системи

### Наступні дисципліни та розділи

- Системний аналіз
- Методи штучного інтелекту
- Текстова аналітика
- Веб-орієнтована розробка програмного забезпечення
- Основи баз даних та ін.

## 3. Зміст навчальної дисципліни

### РОЗДІЛ 1 ПРИНЦИПИ ТА МЕТОДОЛОГІЇ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### Тема 1.1 Розробка і тестування програм

**Зміст.** Що таке тестування ПЗ. Історія розвитку тестування ПЗ. Життєвий цикл розробки ПЗ (SDLC). Керування життєвим циклом програм (ALM)

#### Тема 1.2 Принципи розробки програмного забезпечення

**Зміст.** Принципи розробки DRY, SOLID, KISS, YAGNI

#### Тема 1.3 Моделі розробки програмного забезпечення

**Зміст.** WATERFALL, V-MODEL, ITERATIVE MODEL, INCREMENTAL MODEL, SPIRAL MODEL, PROTOTYPING MODEL, BIG BANG MODEL

#### **Тема 1.4** *Методології гнучкої розробки програмного забезпечення*

**Зміст.** *AGILE методології розробки ПЗ (AGILE MODELING, AGILE TESTING, RAD, EXTREME PROGRAMMING (XP), LEAN, DEVOPS, KANBAN, SCRUMBAN)*

#### **Тема 1.5** *Agile та SCRUM*

**Зміст.** *Agile алгоритми: Scrum. Що таке Scrum? Основні переваги Scrum. Правило 3-5-3 в Scrum. Цінності та принципи Scrum. Ролі в Scrum. Velocity & Capacity. Історії користувача. Poker планування. Події в Scrum. Артефакти в Scrum. Scrum of Scrums. Переваги та обмеження Scrum. Scrum vs інші методології*

### **РОЗДІЛ 2** *ОСНОВИ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ*

#### **Тема 2.1** *Основні принципи тестування ПЗ*

**Зміст.** *Життєвий цикл тестування. Критерії тестування. Основні засади тестування. Тестування документації та вимог. Джерела та шляхи виявлення вимог*

#### **Тема 2.2** *Формування вимог тестування*

**Зміст.** *Рівні і типи вимог. Властивості якісних вимог. Техніки тестування вимог*

#### **Тема 2.3** *Класифікація тестування*

**Зміст.** *Схема класифікації тестування. Класифікація за запуском коду на виконання. Класифікація за ступенем автоматизації. Класифікація за рівнем деталізації застосунку (за рівнем тестування). Загальна класифікація за рівнем тестування. Класифікація за спаданням ступеню важливості тестованих функцій (за рівнем функціонального тестування). Класифікація за принципами роботи із застосунком. Класифікація за природою застосунку. Класифікація за фокусуванням на рівні архітектури застосунку. Класифікація за залученням кінцевих користувачів. Класифікація за ступенем формалізації. Класифікація за цілями і задачами. Класифікація за цілями і підходами. Класифікація за цілями і підходами. Класифікація за моментом виконання (хронологією). Альтернативні та додаткові класифікації тестування*

#### **Тема 2.4** *Тестовий випадок*

**Зміст.** *Чек-листи. Тест-кейс. Властивості тест-кейсів. Мета тест-кейсу та його життєвий цикл. Атрибути (поля) тест-кейсу. Шаблон тест-кейсу. Правила оформлення теми тест-кейса. Система відстеження помилок. Інструментальні засоби керування тестуванням. Властивості якісних тест-кейсів. Набори тест-кейсів*

#### **Тема 2.5** *Дефекти*

**Зміст.** *Помилки, дефекти, збої, відмови тощо. Типи дефектів програмного забезпечення. Звіт про дефект та його життєвий цикл. Атрибути (поля) звіту про дефект. Зразок звіту про помилку/дефект*

#### **Тема 2.6** *Планування та звітність*

**Зміст.** *Тест-план. Звіт про результати тестування. Логіка побудови звіту про результати тестування.*

## **4. Навчальні матеріали та ресурси**

### **Базова література**

1. Тестування програмного забезпечення. Базовий курс. С. С. Куліков. EPAM Systems, 2015–2023. URL: <https://svyatoslav.biz/software testing book/>
2. Технології розробки та тестування програм: Навчальний посібник / Дідковська М.В. – ТОВ «Вік-Принт», 2014. – 344 с.
3. SDLC: System Development Life Cycle. URL: <https://svitla.com/blog/system-development-life-cycle>

4. Principles and Practices of Software Development. Daniel Huttenlocher, Daniel Spoonhower URL: <http://www.cs.cornell.edu/~dph/papers/principles>
5. Software Engineering URL: <https://www.geeksforgeeks.org/software-engineering/?ref=lbp>
6. Software Engineering Tutorial. URL: <https://www.sitesbay.com/software-engineering/index>
7. SDLC Methodologies: From Waterfall to Agile. Oliver Trunkett. URL: <https://www.virtasant.com/blog/sdlc-methodologies>
8. Agile-маніфест розробки програмного забезпечення. URL: <https://agilemanifesto.org/iso/uk/manifesto.html>
9. Scrum Metrics for Hyperproductive Teams: How They Fly like Fighter Aircraft. Scott Downey, Jeff Sutherland. URL: <https://www.agilealliance.org/wp-content/uploads/2016/01/ScrumMetricsAgile2012.pdf>
10. The 2020 Scrum Guide. URL: <https://scrumguides.org/scrum-guide.html>
11. Scrum Guides. URL: <https://www.visual-paradigm.com/scrum/>

### **Додаткова література**

*(факультативно / ознайомлення)*

1. Розробка вимог до програмного забезпечення. Карл Вігерс, Джой Бітті. 2021. 736 с.
2. Тестування. Фундаментальна теорія. URL: <https://dou.ua/forums/topic/13389/>
3. Техніки тест-дизайну для «чайників». URL: <https://dou.ua/forums/topic/44882/>
4. The History of Software Testing. URL: <http://www.testingreferences.com/testinghistory.php>
5. What are the Software Development Models? URL: <http://istqbexamcertification.com/what-are-the-software-development-models>
6. The Growth of Software Testing, David Gelperin, Bill Hetzel. URL: [https://www.researchgate.net/publication/234808293\\_The\\_growth\\_of\\_software\\_testing](https://www.researchgate.net/publication/234808293_The_growth_of_software_testing)
7. Принципи SOLID, про які має знати кожен розробник. URL: <https://medium.com/webbdev/solid-4ffc018077da>
8. SOLID: 5 принципів об'єктно-орієнтованого програмування. Samuel Oloruntoba. URL: [https://www.digitalocean.com/community/conceptual\\_articles/s-o-l-i-d-the-first-five-principles-of-object-oriented-design-ru](https://www.digitalocean.com/community/conceptual_articles/s-o-l-i-d-the-first-five-principles-of-object-oriented-design-ru)
9. Your Complete Guide To Rapid Application Development (RAD) URL: <https://marutitech.com/rapid-application-development/>
10. What Is the Software Development Life Cycle? URL: <https://phoenixnap.com/blog/software-development-life-cycle>
11. Quick and dirty Scrum guide: ceremonies and artifacts. Alaina Ferretti. URL: <https://www.ariadgroup.com/en/blog/all-about-scrum-agile/details-about-scrum-agile-components>
12. Extreme Programming. URL: <http://www.extremeprogramming.org/>
13. The Plague of Aimlessness, James Whittaker. URL: <https://testing.googleblog.com/2009/06/7-plagues-of-software-testing.html>
14. The Plague of Repetitiveness, James Whittaker. URL: <http://googletesting.blogspot.com/2009/06/by-james.html>
15. The Plague of Amnesia, James Whittaker. URL: <http://googletesting.blogspot.com/2009/07/plague-of-amnesia.html>
16. The Plague of Boredom, James Whittaker. URL: <http://googletesting.blogspot.com/2009/07/plague-of-boredom.html>
17. The Plague of Homelessness, James Whittaker. URL: <http://googletesting.blogspot.com/2009/07/plague-of-homelessness.html>
18. The Plague of Blindness, James Whittaker. URL: <http://googletesting.blogspot.com/2009/07/plague-of-blindness.html>

19. The 7th Plague and Beyond, James Whittaker. URL:  
<http://googletesting.blogspot.com/2009/09/7th-plague-and-beyond.html>
20. The Plague of Entropy, James Whittaker. URL:  
<http://googletesting.blogspot.com/2009/09/plague-of-entropy.html>
21. What is Waterfall model advantages, disadvantages and when to use it? URL:  
<http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it>
22. What is V-model advantages, disadvantages and when to use it? URL:  
<http://istqbexamcertification.com/what-is-v-model-advantages-disadvantages-and-when-to-use-it>
23. Using V Models for Testing, Donald Firesmith. URL:  
[https://insights.sei.cmu.edu/sei\\_blog/2013/11/using-v-models-for-testing.html](https://insights.sei.cmu.edu/sei_blog/2013/11/using-v-models-for-testing.html)
24. Project Lifecycle Models: How They Differ and When to Use Them. URL:  
<http://www.businessesolutions.com/islm.htm>
25. Software Testing Life Cycle. URL: <http://softwaretestingfundamentals.com/software-testing-life-cycle/>
26. 7 Principles of Software Testing. URL: <https://www.interviewbit.com/blog/principles-of-software-testing/>
27. What is documentation testing in software testing? URL:  
<http://istqbexamcertification.com/what-is-documentation-testing/>
28. Requirements Gathering vs. Elicitation. Laura Brandenburg. URL: <http://www.bridging-the-gap.com/requirements-gathering-vs-elicitation/>
29. Software Requirements Engineering: What, Why, Who, When, and How? Linda Westfall. URL:  
[http://www.westfallteam.com/Papers/The\\_Why\\_What\\_Who\\_When\\_and\\_How\\_Of\\_Software\\_Requirements.pdf](http://www.westfallteam.com/Papers/The_Why_What_Who_When_and_How_Of_Software_Requirements.pdf)

### **Інформаційні ресурси**

1. ISTQB. All terms. URL: [https://glossary.istqb.org/en\\_US/search?term=&exact\\_matches\\_first=true](https://glossary.istqb.org/en_US/search?term=&exact_matches_first=true)
2. International Software Testing Qualifications Board Glossary. URL:  
<http://www.istqb.org/downloads/glossary.html>
3. Scrum-Institute. URL: <https://www.scrum-institute.org/>
4. TRY QA. URL: <https://tryqa.com/>
5. Ресурси про тестування за версією QA-спільноти DOU. URL:  
[https://dou.ua/forums/topic/36445/?from=similar\\_topics](https://dou.ua/forums/topic/36445/?from=similar_topics)
6. Принцип хорошого коду KISS ("keep it simple, stupid", "keep it short and simple") URL:  
<https://youtu.be/rix-fkrlog4>
7. Принцип хорошого коду YAGNI ("You aren't gonna need it") URL: <https://youtu.be/Ot2eB07rjcl>
8. Принцип хорошого коду DRY (dont repeat yourself) URL: <https://youtu.be/NWemaqNMCesQ>
9. SOLID принципи: SRP (Принцип одноосібної відповідальності, Single Responsibility Principle) URL: <https://youtu.be/O4uhPCEDzSo>
10. SOLID принципи: OCP (Откритості/закритості (Open Closed Principle) URL:  
<https://youtu.be/x5OtQikOG-Q>
11. SOLID: Принцип подстановки Барбара Лісков/ LSP (The Liskov Substitution Principle) URL:  
<https://youtu.be/NqvwYcjrwdw>
12. Тестування програмного забезпечення. URL:  
[https://uk.wikipedia.org/wiki/Тестування\\_програмного\\_забезпечення](https://uk.wikipedia.org/wiki/Тестування_програмного_забезпечення)
13. Тестування програмного забезпечення. URL: <https://software-testing.org/>
14. Огляд процесу розробки програмного забезпечення. URL:  
<https://habr.com/ru/articles/255991/>

15. What are the Software Development Models? URL: <https://tryqa.com/what-are-the-software-development-models/>
16. Жизненный цикл ПО. Каскадная модель (Waterfall). URL: <https://xbsoftware.ru/blog/zhiznennyi-tsykl-po-kaskadnaya-model-waterfall/>
17. Agile фан-клуб. Харківський AGILE Фан-Клуб. URL: <https://agile.kh.ua/>
18. When to use Scrum? Waterfall vs. Scrum vs. Kanban vs. Scrumban. URL: <https://dusted.codes/when-to-use-scrum-waterfall-vs-scrum-vs-kanban-vs-scrumban>
19. 33 тестера. URL: <https://33testers.blogspot.com/2013/07/blog-post.html>
20. Jira Software. URL: <https://www.atlassian.com/software/jira>
21. Trello. URL: <https://trello.com/en-US>

## Навчальний контент

### 5. Методика опанування навчальної дисципліни (освітнього компонента)

Навчальна дисципліна охоплює 36 годин лекцій та 18 годин практичних занять, а також виконання модульної контрольної роботи, яка складається з тестових питань.

Практичні заняття з дисципліни проводяться з метою закріплення теоретичних положень навчальної дисципліни і набуття студентами практичних навичок тестування програмного забезпечення.

№ п/п	Теми	Кількість годин, відведених на:			Термін виконання
		лекції	практ. заняття	сам. робота.	
1	2	3	4	5	
P1	<b>ПРИНЦИПИ ТА МЕТОДОЛОГІЇ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</b>				
T1.1	<i>Розробка і тестування програм</i>	2	-	4	1-й тиждень
T1.2	<i>Принципи розробки програмного забезпечення</i>	2	-	4	2-й тиждень
T1.3	<i>Моделі розробки програмного забезпечення</i>	2	-	6	3-й тиждень
T1.4	<i>Методології гнучкої розробки програмного забезпечення</i>	4	-	4	5-й тиждень
T1.5	<i>Agile та SCRUM</i>	4	-	6	7-й тиждень
P2	<b>ОСНОВИ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</b>				
T2.1	<i>Основні принципи тестування ПЗ</i>	2	-	4	8-й тиждень
T2.2	<i>Формування вимог тестування</i>	2	2	4	9-й тиждень
T2.3	<i>Класифікація тестування</i>	6	4	8	12-й тиждень
T2.4	<i>Тестовий випадок</i>	6	6	10	15-й тиждень
T2.5	<i>Дефекти</i>	4	2	8	17-й тиждень
T2.6	<i>Планування та звітність</i>	2	4	8	18-й тиждень
	<b>Всього</b>	<b>36</b>	<b>18</b>	<b>66</b>	

Виходячи з розподілу часу на вивчення дисципліни, рекомендується дев'ять практичних робіт протягом семестру:

№ п/п	Тема
ПР1	Аналіз та тестування вимог
ПР2	Чек-лист та тест кейс
ПР3	Реалізація логіки створення ефективних перевірок на прикладі чек-листа
ПР4	Типові помилки при розробці чек-листів, тест-кейсів та наборів тест-кейсів
ПР5	Дефекти
ПР6	Тест-план та звіт про результати тестування
ПР7	Оцінка трудовитрат
ПР8	Використання різних технік тестування
ПР9	Автоматизація тестування

## 6. Самостійна робота студента

Самостійна робота студента охоплює такі складники як підготування до поточних фронтальних опитувань за тематикою лекцій, підготування до практичних занять, підготування до модульної контрольної роботи.

## Політика та контроль

### 7. Політика навчальної дисципліни (освітнього компонента)

**Відвідування занять.** Відсутність на аудиторному занятті не передбачає нарахування штрафних балів, оскільки фінальний рейтинговий бал студента формується виключно на основі оцінювання результатів навчання. Разом з тим, обговорення результатів виконання тематичних завдань, а також виконання та представлення практичних робіт оцінюватимуться під час аудиторних занять.

**Пропущені контрольні заходи оцінювання.** Кожен студент має право відпрацювати пропущені з поважної причини (лікарняний, мобільність тощо) заняття за рахунок самостійної роботи. Детальніше за посиланням: <https://kpi.ua/files/n3277.pdf>.

**Процедура оскарження результатів контрольних заходів оцінювання.** Студент може підняти будь-яке питання, яке стосується процедури контрольних заходів та очікувати, що воно буде розглянуто згідно із наперед визначеними процедурами. Студенти мають право аргументовано оскаржити результати контрольних заходів, пояснивши з яким критерієм не погоджуються відповідно до оціночного.

**Календарний контроль** проводиться з метою підвищення якості навчання студентів та моніторингу виконання студентом вимог силабусу.

Критерій		Перший календарний контроль	Другий календарний контроль
Термін календарного контролю		Тиждень 8	Тиждень 14
Умови отримання позитивної оцінки	Поточний рейтинг	≥ 10 балів	≥ 30 балів

**Академічна доброчесність.** Політика та принципи академічної доброчесності визначені у розділі 3 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.

**Норми етичної поведінки.** Норми етичної поведінки студентів і працівників визначені у розділі 2 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.

**Інклюзивне навчання.** Засвоєння знань та умінь в ході вивчення дисципліни «Розробка і тестування програм» може бути доступним для більшості осіб з особливими освітніми потребами, окрім здобувачів з серйозними вадами зору, які не дозволяють виконувати завдання за допомогою персональних комп'ютерів, ноутбуків та/або інших технічних засобів.

**Навчання українською мовою.** У ході виконання завдань студентам може бути рекомендовано звернутися до англомовних джерел.

**Призначення заохочувальних та штрафних балів** Відповідно до Положення про систему оцінювання результатів навчання сума всіх заохочувальних балів не може перевищувати 10% рейтингової шкали оцінювання.

<i>Критерій</i>	<i>Заохочувальні бали</i>		<i>Штрафні бали</i>	
	<i>Ваговий бал</i>	<i>Критерій</i>	<i>Ваговий бал</i>	<i>Критерій</i>
Написання тез, статті (за тематикою навчальної дисципліни)	5-10 балів	-	-	-
Участь у міжнародних, всеукраїнських та/або інших заходах та/або конкурсах (за тематикою навчальної дисципліни)	5-10 балів	-	-	-
Проходження дистанційних курсів на платформі Coursera за тематикою дисципліни	5-10 балів	-	-	-

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

*Семестровий контроль проводиться у вигляді заліку. Для оцінювання результатів навчання застосовується 100-бальна рейтингова система та університетська шкала.*

**Поточний контроль:** фронтальні опитування, практичні роботи, МКР.

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

**Семестровий контроль:** залік

*Якщо семестровий рейтинг більше 60 балів студент може не виходити на залікову контрольну роботу, а отримати оцінку «автоматом».*

### Перелік контрольних заходів

<i>№ п/п</i>	<i>Контрольний захід оцінювання</i>	<i>Ваговий бал</i>	<i>Кількість</i>	<i>Разом</i>
1	Фронтальні опитування	6	6	36
2	Практичні роботи	4	9	36
3	Модульна контрольна робота	28	1	28
	Разом		16	100



## Система оцінювання контрольних заходів

### Компонент

### Система оцінювання

1	<b>Фронтальне опитування</b> проводиться у вигляді тестів і складається з 72 питань за тематикою лекцій. За кожне питання можна отримати 0,5 балів у разі надання правильної відповіді. За правильні відповіді на усі тестові питання протягом семестру можна отримати 36 балів.
2	<b>Практичні роботи</b> оцінюються за допомогою виконання тестів, які можуть містити 6 чи 12 питань (в залежності від об'єму матеріалу, який винесений для опрацювання в певній практичній роботі), за кожне з яких можна отримати 0,5 балів у разі надання правильної відповіді. Загальна кількість питань, які містять тести для засвоєння знань та навичок, що винесені у практичні роботи, складає 72 та за правильні відповіді на всі питання можна отримати протягом семестру 36 балів.
6	<b>Модульна контрольна робота</b> являє собою набір тестових питань для перевірки отриманих протягом семестру навичок. Модульна контрольна робота складається з 28 тестових питань, за кожне з яких можна отримати 1 бал у разі правильної відповіді. За виконання модульної контрольної роботи студент отримує 28 балів максимум. Оцінювання виконується наступним чином: <ul style="list-style-type: none"><li>• «відмінно» – 25 – 28 балів;</li><li>• «добре» – 22 – 24 балів;</li><li>• «задовільно» – 17 - 21 балів;</li><li>• «незадовільно» – &lt;17 балів.</li></ul>

Для отримання заліку з навчальної дисципліни «автоматом» потрібно мати рейтинг не менше 60 балів, а також зараховані 4 фронтальних опитування, зараховані 5 практичних робіт за результатами тестових опитувань і зарахована модульна контрольна робота.

Студенти, які наприкінці семестру мають рейтинг менше 60 балів, а також ті, хто хоче підвищити оцінку, виконують залікову контрольну роботу.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

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

### **9. Додаткова інформація з дисципліни (освітнього компонента)**

Методи та форми навчання включають не лише традиційні університетські лекції та практичні заняття, а також візуалізацію та інформаційно-комунікаційні технології, зокрема електронні презентації для лекційних занять, відео матеріали до лекційних занять та практичних робіт. Комунікація з викладачем будується за допомогою використання інформаційної системи «Електронний кампус», платформи дистанційного навчання «Сікорський», а також такими інструментами комунікації, як веб-ресурс викладача, електронна пошта, месенджер Telegram. Під час навчання та для взаємодії зі студентами

використовуються сучасні інформаційно-комунікаційні та мережеві технології для вирішення навчальних завдань.

**Факультативне навчання.** Для кращого розуміння засад, принципів розробки та тестування програмного забезпечення пропонується проходження онлайн-курсів відповідної тематики.

**Робочу програму навчальної дисципліни (силабус):**

**Складено:**

доцент кафедри штучного інтелекту,

канд. техн. наук, доцент, Бендюг Владислав Іванович

**Ухвалено** кафедрою штучного інтелекту (протокол № 14 від 11.06.2024)

**Погоджено** Методичною комісією ННІПСА (протокол № 10 від 24.06.2024)