



# ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## Робоча програма навчальної дисципліни (Силабус)

### 1. Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>11 Математика та статистика</i>
Спеціальність	<i>113 Прикладна математика</i>
Освітня програма	<i>Наука про дані та математичне моделювання</i>
Статус дисципліни	<i>Вибіркова</i>
Форма навчання	<i>Очна (денна)</i>
Рік підготовки, семестр	<i>2 курс, весняний семестр</i>
Обсяг дисципліни	<i>4 кредити, 120 годин</i>
Семестровий контроль/ контрольні заходи	<i>Залік</i>
Розклад занять	<i>Тижневих годин: 2 год. - лекції, 2 год. – лабораторні заняття</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	Лектор: <i>к.ф.-м.н., Бай Юлія Петрівна, ju.p.bai@gmail.com</i> Лабораторні: <i>Бай Юлія Петрівна, ju.p.bai@gmail.com</i>
Розміщення курсу	-

### 2. Програма навчальної дисципліни

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

Вивчення кредитного модуля «Забезпечення якості ПЗ» дає змогу сформувати в студентів компетенції, потрібні для забезпечення якості програмного забезпечення, у тому числі здійснення аналізу вимог до програмного забезпечення, знання і застосування на практиці основних методів і засобів верифікації ПЗ, знання основних програмних засобів, що застосовуються для автоматизованого тестування ПЗ.

**Метою** кредитного модуля є формування у студентів здатностей:

- розуміти поняття якості ПЗ в усіх його проявах;
- аналізувати вимоги до програмного забезпечення;
- застосовувати методи верифікації (статичний аналіз коду, рецензія, тестування) для перевірки якості ПЗ;
- аналізувати, класифікувати та описувати дефекти, знайдені під час верифікації ПЗ;
- виконувати автоматизоване тестування програм на мові Python, а також тестування веб-застосунків за допомогою Selenium IDE, Selenium WebDriver та Python.

**Основні завдання кредитного модуля.**

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

**Знання:**

- поняття якості ПЗ, метрик якості;
- поняття технічного боргу та способів його зменшення;

- методів верифікації ПЗ: тестування, рецензій, статичного аналізу;
- методів створення тестових випадків та звітів про дефекти;
- способів поліпшення процесу забезпечення якості ПЗ;

#### **уміння:**

- контролювати якість виконання проектних робіт та розроблення ПЗ колективом розробників з урахуванням діючих державних, міжнародних, професійних та корпоративних стандартів;
- проводити контроль відповідності розробленого ПЗ вимогам замовника;
- контролювати правильність роботи ПЗ за допомогою методів верифікації на різних рівнях (модульному, інтеграційному, системному тощо);
- розробляти тести (тестові набори, сценарії та коди), у тому числі автоматизовані;

#### **досвід:**

- аналізу вимог до ПЗ;
- розроблення тестових випадків для заданого ПЗ;
- виконання розроблених тестових випадків;
- формування звітів про знайдені дефекти;
- автоматизації тестування програм на мові Python, а також веб-застосунків за допомогою Selenium IDE, Selenium WebDriver та Python.

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

Дисципліні «Забезпечення якості ПЗ» передують дисципліни «Програмування», «Алгоритми і структури даних», «Дискретна математика», «Математична логіка та теорія алгоритмів» навчального плану підготовки бакалаврів за спеціальністю 113 Прикладна математика.

Дисципліна «Забезпечення якості ПЗ» є необхідною для якісного виконання студентами курсових проектів з дисциплін «Бази даних та інформаційні системи», «Алгоритми і системи комп'ютерної математики» навчального плану підготовки бакалаврів за спеціальністю 113 Прикладна математика, а також успішного дипломного проектування.

## **3. Зміст навчальної дисципліни**

- 1. Якість програмного забезпечення.** Поняття якості. Професійний підхід до якості ПЗ. Види якості ПЗ. Забезпечення якості та контроль якості. Поняття валідації та верифікації ПЗ.
- 2. Моделі якості ПЗ.** Моделі Боема та МакКола якості програмного забезпечення. Метрики якості.
- 3. Стандарти якості ПЗ.** Міжнародні стандарти і моделі якості ПЗ ISO/IEC 9126, ISO/IEC 9001.
- 4. Поліпшення якості ПЗ.** Моделі зрілості розробки програмного забезпечення: CMMI, TMMI. Вартість якості.
- 5. Моделі життєвого циклу ПЗ.** Каскадна та V-подібна моделі. Інкрементна, еволюційна моделі. Спіральна модель. Гнучкі методології розробки ПЗ (Agile): Scrum, Extreme Programming.
- 6. Аналіз вимог до програмного забезпечення.** Властивості якісних вимог.
- 7. Тестування як засіб підвищення якості ПЗ.** Основні визначення. Аксиоми і принципи тестування. Місце тестування в життєвому циклі розробки ПЗ. Ціна виправлення помилок на різних етапах ЖЦ ПЗ.
- 8. Класифікація тестування програмного забезпечення.** Рівні та типи тестування.
- 9. Статичний аналіз програмного коду.** Ревізії (інспекції) коду. Типи ревізій. Складність коду.
- 10. Функціональне тестування ПЗ.** Методи «чорного ящика»: розбиття на класи еквівалентності; аналіз граничних значень; тестування на основі таблиць рішень; тестування станів і переходів; тестування на основі UML-діаграм.
- 11. Структурне тестування ПЗ.** Методи «білого ящика». Рівні покриття програмного коду: покриття операторів; покриття рішень; покриття умов; покриття шляхів. Тестування потоків даних.
- 12. Тестова документація.** Стандарт IEEE-829. Чек-ліст, тест-план, тест-кейси, характеристики якісного тест-кейсу, набори тест-кейсів, матриця відстежуваності.
- 13. Звіти про дефекти.** Класифікація дефектів у програмному забезпеченні. Життєвий цикл дефекту. Властивості якісного звіту про дефект. Інструментальні засоби управління звітами.
- 14. Критерії завершення тестування.** Підсумковий звіт з тестування: мета та складові.
- 15. Основи автоматизації тестування ПЗ.** Переваги і ризики автоматизації. Области застосування. Необхідні знання і навички.
- 16. Особливості створення тест-кейсів в автоматизованому тестуванні.**

17. **Основні підходи в автоматизації тестування.** Часткові рішення, «запис-відтворення», функціональна декомпозиція, тестування під керуванням даними, тестування під керуванням ключовими словами. Огляд інструментів автоматизації тестування.

#### 4. Навчальні матеріали та ресурси

##### Базова література:

3. Інженерія програмного забезпечення. Посібник для студентів вищих навчальних закладів / І. Бородкіна, Г. Бородкін. – Центр навчальної літератури, 2018. – 204 с.
4. Дідковська М.В. Технології розробки та тестування програм: Навчальний посібник / М.В. Дідковська. – ТОВ «Вік-Принт», 2015. – 344 с.
5. Якість програмного забезпечення та тестування: базовий курс / За ред. Крепич С. Я., Співак І. Я. – Тернопіль: ФОП Паляниця В.А., 2020. – 478 с.
6. Тестування програмного забезпечення : Навчальний посібник / Авраменко А. С., Авраменко В. С., Косенюк Г. В. – Черкаси : ЧНУ імені Богдана Хмельницького, 2017. – 284 с.
7. Сидоров М. О. Якість програмного забезпечення та тестування: Підручник / М. О. Сидоров, М. А. Безверха. – К. : НАУ, 2010. – 282 с.

##### Додаткова література:

1. Galin D. Software Quality Assurance: From Theory to Implementation / D. Galin. — Pearson, 2003. — 616p.
2. Myers G. J. The Art of Software Testing, 2nd edition / G. J. Myers. — New York: John Wiley and Sons Inc., 2004. — 254 p.
3. Copeland L. Practitioner's Guide to Software Test Design / L. Copeland. — Artech House, 2013. — 312 p.
4. Software Requirements. 3<sup>rd</sup> edition. / K. Wiegers, J. Beatty. — Microsoft, 2013. — 673 p.
5. Patton R. Software Testing / R. Patton. — Indianapolis: SAMS, 2005. — 408 p.
6. Ammann P. Introduction to Software Testing / P. Ammann, J. Offutt. — Cambridge University Press, 2008. — 346 p.
7. Morgan P. Software Testing: An ISTQB-ISEB Foundation Guide / P. Morgan, A. Samaroo, G. Thompson, P. Williams. — British Computer Society, 2010. — 224 p.
8. Naik K. Software Testing and Quality Assurance: Theory and Practice / K. Naik, P. Tripathy. — Wiley-Spectrum, 2011. — 648 p.

## 8. Навчальний контент

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

#### 5.1. Лекційні заняття

№ з/п	Назва теми лекції та перелік основних питань	Кількість ауд. годин
1	<b>Якість програмного забезпечення</b> Поняття якості. Види якості. Професійний підхід до якості ПЗ. Забезпечення якості та контроль якості. Поняття валідації та верифікації ПЗ.	2
2	<b>Моделі якості ПЗ</b> Метрики якості ПЗ. Модель якості Боєма. Модель якості МакКола. Використання моделей якості.	2
3	<b>Стандарти якості.</b> Моделі якості і стандарти програмного забезпечення ISO/IEC 9126, ISO 9000:2000	2
4	<b>Поліпшення якості ПЗ.</b> Моделі зрілості створення програмного забезпечення CMMI, TMMI. Вартість якості.	2
5	<b>Моделі життєвого циклу ПЗ.</b> Каскадна модель. V-подібна модель. Ітераційні моделі: інкрементна та еволюційна моделі. Спіральна модель. Гнучкі методології розробки ПЗ (Agile): Scrum, Extreme Programming.	2

№ з/п	Назва теми лекції та перелік основних питань	Кількість ауд. годин
6	<b>Аналіз вимог до програмного забезпечення.</b> Характеристики якісних вимог до ПЗ. Усунення дефектів у вимогах.	2
7	<b>Тестування як засіб підвищення якості ПЗ.</b> Основні визначення. Аксиоми тестування. Місце тестування в життєвому циклі розробки ПЗ. Ціна помилок. Принципи тестування.	2
8	<b>Класифікація тестування програмного забезпечення.</b> Рівні та типи тестування. Модульне, інтеграційне, системне тестування. Приймальне тестування. Статичне й динамічне тестування. Позитивне й негативне тестування. Тестування вручну і автоматизоване.	2
9	<b>Статичний аналіз ПЗ.</b> Ревізії (інспекції) коду. Типи ревізій. Цикломатична складність коду за МакКейбом. Метрики Холстеда та Чепіна складності коду.	2
10	<b>Функціональне тестування ПЗ.</b> Методи «чорного ящика»: розбиття на класи еквівалентності; аналіз граничних значень; тестування на основі таблиць рішень; тестування станів і переходів.	2
11	<b>Структурне тестування ПЗ.</b> Методи «білого ящика»: покриття операторів; покриття рішень; покриття умов; покриття шляхів. Тестування потоків даних.	2
12	<b>Тестова документація:</b> Чек-ліст, тест-план, тест-кейси, характеристики якісного тест-кейсу, набори тест-кейсів, матриця відстежуваності.	2
13	<b>Звітування про дефекти.</b> Класифікація дефектів у програмному забезпеченні. Життєвий цикл дефекту та звіту про дефект. Інструментальні засоби управління звітами про дефекти. Властивості якісного звіту про дефект.	2
14	<b>Критерії завершення тестування.</b> Підсумковий звіт з тестування, його мета та складові.	2
15	<b>Основи автоматизації тестування ПЗ.</b> Переваги і ризики автоматизації. Области застосування. Необхідні знання і навички.	2
16	<b>Особливості створення тест-кейсів в автоматизованому тестуванні.</b>	2
17	<b>Основні підходи в автоматизації тестування.</b> Часткові рішення, «запис-відтворення», функціональна декомпозиція, тестування під керуванням даними, керуванням ключовими словами, керуванням поведінкою. Огляд інструментів автоматизації тестування	2

## 5.2. Лабораторні заняття

№ з/п	Назва та завдання лабораторної роботи	Кількість ауд. годин
1	<b>Виявлення помилок в ПЗ. Створення простих тестових випадків</b> Завдання: навчитись розробляти тестові випадки для тестування заданого ПЗ відповідно до прийнятих стандартів, визначати тестові процедури для проведення тестування.	4
2	<b>Аналіз вимог до програмного забезпечення</b> Завдання: навчитись аналізувати вимоги до програмного забезпечення, знаходити та класифікувати дефекти у вимогах.	4
3	<b>Метрики якості програмного коду. Бібліотека Radon Python</b> Завдання: навчитись визначати метрики програмного коду: метрики розміру програм, цикломатичну складність за МакКейбом, метрики Холстеда, індекс підтримованості коду.	4

№ з/п	Назва та завдання лабораторної роботи	Кількість ауд. годин
4	<b>Тестування «чорного ящика»</b> Завдання: навчитись розробляти тест-кейси на основі розбиття на класи еквівалентності, аналізу граничних значень, аналізу таблиць рішень, діаграм станів і переходів.	4
5	<b>Тестування «білого ящика»</b> Завдання: навчитись розробляти тест-кейси для покриття виразів, покриття гілок (рішень), покриття умов, покриття шляхів, а також визначати рівень покриття тест-кейсами заданого коду.	4
6	<b>Документування тест-кейсів. Звітування про дефекти</b> Завдання: навчитись створювати якісні тест-кейси та звіти про дефекти, а також документувати їх відповідно до прийнятих стандартів.	4
7	<b>Автоматизоване тестування програм на мові Python</b> Завдання: опанувати бібліотеки Python unittest, subprocess; навчитись розробляти і виконувати автоматизацію тестування на мові Python.	6
8	<b>Автоматизоване тестування веб-застосунків за допомогою Selenium IDE, Selenium WebDriver та Python</b> Завдання: опанувати Selenium IDE, Selenium WebDriver; навчитись виконувати автоматизацію тестування веб-орієнтованих аплікацій за допомогою Selenium IDE, Selenium WebDriver та Python	6

## 6. Самостійна робота студента

№ з/п	Назва теми та завдання на самостійну роботу студента	Кількість ауд. годин
1	<b>Якість програмного забезпечення.</b> Проаналізувати місце забезпечення якості ПЗ в процесі розроблення ПЗ	3
2	<b>Моделі якості ПЗ.</b> Детальніше ознайомитися з питаннями використання моделей якості (модель «дім якості»)	3
3	<b>Стандарти якості.</b> Опрацювати стандарти якості ISO/IEC 9126, ISO 9000:2000	3
4	<b>Поліпшення якості ПЗ.</b> Розглянути модель TPI Next	3
5	<b>Моделі життєвого циклу ПЗ.</b> Опрацювати традиційні моделі ЖЦ ПЗ та гнучкі методології розробки	3
6	<b>Аналіз вимог до програмного забезпечення.</b> Опрацювати критерії якісних вимог та можливі шляхи усунення дефектів у вимогах до програмного забезпечення	3
7	<b>Тестування як засіб підвищення якості ПЗ.</b> Ознайомитися з основними принципами психології тестувальника. Підготуватися до МКР-1.	3
8	<b>Класифікація тестування програмного забезпечення.</b> Опрацювати детальну класифікацію тестування. Опрацювати типи приймального тестування.	3
9	<b>Статичний аналіз ПЗ.</b> Розглянути чинники успішних ревізій.	3
10	<b>Функціональне тестування ПЗ.</b> Розібратися з тестуванням сценаріїв використання	3
11	<b>Структурне тестування ПЗ.</b> Детальніше розібратися з методом MC/DC	2
12	<b>Тестова документація.</b> Розібратися з перевагами створення матриці відстежуваності. Опрацювати різні варіанти запису тест-кейсів	2



№ з/п	Назва теми та завдання на самостійну роботу студента	Кількість ауд. годин
13	<b>Звітування про дефекти.</b> Розглянути життєвий цикл дефекту в різних баг-трекінгових системах.	2
14	<b>Критерії завершення тестування.</b> Опрацювати види активностей тестувальників на етапі завершення тестування	2
15	<b>Основи автоматизації тестування ПЗ.</b> Розібратися з перевагами й вадами використання засобів автоматизації виконання тестування. Опрацювати бібліотеку unittest Python	3
16	<b>Особливості створення тест-кейсів в автоматизованому тестуванні.</b> Розібратися з помилками, які можуть виникати при підготовці тест-кейсів для автоматизованого тестування та правильними варіантами запису тест-кейсів	3
17	<b>Основні підходи в автоматизації тестування.</b> Опрацювати ідеологію тестування під керуванням ключовими словами на прикладі Selenium IDE. Підготуватися до МКР-2.	4

## 9. Політика та контроль

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

- **Відвідування лекцій та лабораторних занять.** Відсутність на лекціях та лабораторних заняттях без поважної причини не допускається.
- **Правила поведінки на заняттях.** На лекційних та лабораторних заняттях студенти мають вимкнути мобільні телефони або увімкнути їх на беззвучний режим.
- **Правила захисту лабораторних робіт.** Усі лабораторні роботи оформляються і здаються студентами у вигляді звітів – doc або pdf-файлів. Захист лабораторних робіт відбувається у вигляді усної співбесіди зі студентом за результатами оформленого звіту, також студенту можуть бути поставлені запитання зі списку контрольних питань, що міститься в кінці кожної лабораторної роботи.
- **Правила призначення заохочувальних та штрафних балів.** За активну участь на лекціях та лабораторних заняттях передбачаються заохочувальні бали в кількості до 4-х балів за семестр. Штрафні бали призначаються за несвоєчасне виконання лабораторних робіт (див. нижче PCO)
- **Політика дедлайнів та перескладань.** Залік проводиться на останньому лекційному занятті в семестрі. У випадку неотримання студентом заліку, він має здавати його на додатковій сесії, згідно з розкладом перескладань.
- **Політика щодо академічної доброчесності.** Згідно з Кодексом честі студента КПІ, при виконанні лабораторних робіт **забороняється** користуватися чужими виконаними лабораторними роботами та їх фрагментами. У випадку виявлення плагіату лабораторна робота може бути оцінена від 0 до 1/3 сумарної кількості запланованих за неї балів.

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

Рейтинг студента з кредитного модуля складається з балів, які він отримує:

- 1) за виконання та захист лабораторних робіт;
- 2) за виконання модульної контрольної роботи;
- 3) за виконання залікової контрольної роботи.

Алгоритм процесу оцінювання успішності студентів наведено на рис. 1.

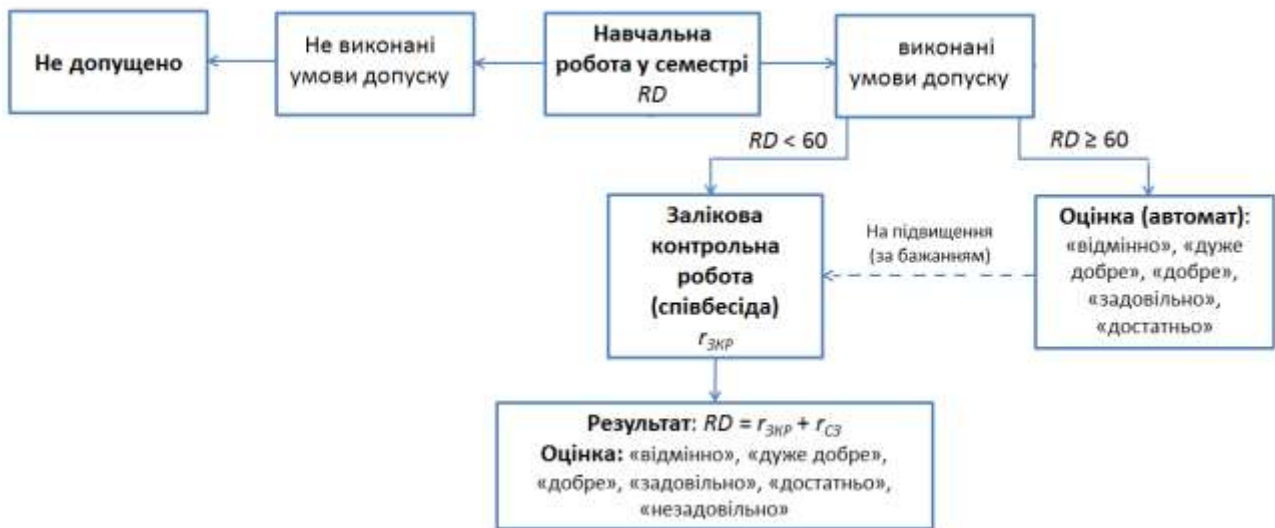


Рис. 1. Схема функціонування PCS

## СИСТЕМА РЕЙТИНГОВИХ БАЛІВ

### 8.1. Бали за виконання та захист лабораторних робіт

Протягом семестру студенти виконують 8 лабораторних робіт:

- 1) Виявлення помилок в ПЗ. Створення простих тестових випадків.
- 2) Аналіз вимог до програмного забезпечення.
- 3) Метрики якості програмного коду.
- 4) Тестування «чорного ящика».
- 5) Тестування «білого ящика».
- 6) Документування тест-кейсів. Звітування про дефекти.
- 7) Автоматизоване тестування програм на мові Python.
- 8) Автоматизоване тестування веб-застосунків (web applications) за допомогою Selenium IDE, Selenium WebDriver та Python.

Максимальна кількість балів за кожну лабораторну роботу: **7 балів**.

Бали нараховуються за:

- якість реалізації роботи: 0-6 балів;
- своєчасність виконання: 0-1 бали.

**Критерії оцінювання якості реалізації роботи:**

- 6 балів — робота виконана якісно, в повному обсязі;
- 4-5 балів — робота виконана якісно, в повному обсязі, але має незначні помилки;
- 2-3 бали — робота виконана в повному обсязі, але містить значні помилки;
- 0-1 балів — робота виконана не в повному обсязі і має суттєві помилки.

**Критерії оцінювання своєчасності виконання лабораторної роботи:**

- 1 бал — роботу здано вчасно, в зазначений термін;
- 0 балів — роботу здано пізніше, ніж через 1 тиждень від зазначеного терміну.

**Максимальна кількість балів за виконання та захист лабораторних робіт:**

7 балів × 8 лабораторних робіт = **56 балів**.

### 8.2. Бали за виконання модульної контрольної роботи

Модульна контрольна робота поділяється на дві 45-хвилинні контрольні роботи, кожна з яких містить 22 запитання тестового характеру.

Максимальна кількість балів за відповідь на кожне питання: **1 бал**.

**Критерії оцінювання:**

1 бал — відповідь правильна;

0 балів — відповідь неправильна.

**Максимальна кількість балів за модульний контроль:**

1 бал × 22 питання × 2 КР = **44 бали**.

### 8.3. Бали за виконання залікової контрольної роботи

Залікова контрольна робота містить 11 завдань. Ваговий бал: **4 бали** за кожне завдання.

Максимальна кількість балів за залікову роботу:

4 бали × 11 завдань = **44 бали**.

*Критерії оцінювання:*

4 бали – повна та правильна відповідь;

3 бали – повна та в цілому правильна відповідь з незначними помилками;

1-2 бали – відповідь неповна або містить суттєві помилки;

0 балів – відповідь неправильна або немає відповіді.

### 8.4. Розрахунок шкали (R) рейтингу

Рейтингова шкала з дисципліни складає **R = 100 балів**.

Сума вагових балів протягом семестру складає:

$$R_c = 56 + 44 = 100 \text{ балів.}$$

Необхідною умовою допуску студента до заліку є виконання всіх лабораторних робіт. Студенти, які допущені до заліку, отримують оцінку з дисципліни **R<sub>c</sub>** автоматично.

Якщо студент не згоден із попередньою оцінкою, яку він отримав «автоматом», то він пише залікову контрольну роботу. Остаточний результат **R<sub>c</sub>** складається з балів за виконання залікової контрольної роботи та балів за виконання лабораторних робіт і виставляється в залікову відомість. Оцінка (**ECTS** та традиційна) виставляється відповідно до значення **R<sub>c</sub>**, згідно з таблицею.

### 8.5. Поточна атестація

На першій атестації, 8-й тиждень, студент отримує «атестовано», якщо його поточний рейтинг складає не менше, ніж **25 балів**, тобто 50% від максимальної кількості балів, яку може отримати студент до першої атестації.

На другій атестації, 14-й тиждень, студент отримує «атестовано», якщо його поточний рейтинг складає не менш, ніж **50 балів**, тобто 50% від максимальної кількості балів, яку може отримати студент до другої атестації.

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
< 60	Незадовільно
Не виконані умови допуску	Не допущено

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

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



**Складено** к.ф.-м.н., старшим викладачем кафедри ПМА Бай Ю.П.

**Ухвалено** кафедрою прикладної математики (протокол № 7 від 09.02.2022 )

**Погоджено** Методичною комісією факультету прикладної математики (протокол № 6 від 25.03.2022 )