



ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Робоча програма навчальної дисципліни (Силабус)

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 год. (36 год. – лекції, 36 год. – лабораторні заняття, 48 год. – СРС)</i>
Семестровий контроль/ контрольні заходи	<i>Залік, МКР</i>
Розклад занять	<i>Тижневих годин: 2 год. - лекції, 2 год. – лабораторні заняття</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лекції і лабораторні заняття: к.ф.-м.н., Бай Юлія Петрівна, ju.p.bai@gmail.com</i>
Розміщення курсу	<i>-</i>

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 під час 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, тестування веб-сторінок та веб-застосунків за допомогою Selenium IDE, Selenium WebDriver та Python.

Під час вивчення дисципліни студенти набудуть:

I. загальних компетентностей:

- ЗК1. Здатність учитися і оволодівати сучасними знаннями
- ЗК2. Здатність застосовувати знання у практичних ситуаціях
- ЗК3. Здатність генерувати нові ідеї (креативність)
- ЗК5. Здатність проведення досліджень на відповідному рівні
- ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел
- ЗК8. Знання та розуміння предметної області та розуміння професійної діяльності
- ЗК12. Визначеність і наполегливість щодо поставлених завдань і взятих обов'язків

II. фахових компетентностей:

- ФК4. Здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію
- ФК8. Здатність використовувати сучасні технології програмування та тестування програмного забезпечення
- ФК9. Здатність до проведення математичного і комп'ютерного моделювання, аналізу та обробки даних, обчислювального експерименту, розв'язання формалізованих задач за допомогою спеціалізованих програмних засобів
- ФК13. Здатність зрозуміти постановку завдання, сформульовану мовою певної предметної галузі, здійснювати пошук та збір необхідних вихідних даних
- ФК17. Здатність розробляти програмне забезпечення мовою Python, у тому числі з використанням бібліотек NumPy, Pandas, Matplotlib, Scikit-learn

Програмними результатами навчання є:

- РН4. Виконувати математичний опис, аналіз та синтез дискретних об'єктів та систем, використовуючи поняття й методи дискретної математики та теорії алгоритмів
- РН7. Вміти проводити практичні дослідження та знаходити розв'язок некоректних задач
- РН11. Вміти застосовувати сучасні технології програмування та розроблення програмного забезпечення, програмної реалізації чисельних і символічних алгоритмів
- РН14. Виявляти здатність до самонавчання та продовження професійного розвитку
- РН15. Уміти організувати власну діяльність та одержувати результат у рамках обмеженого часу
- РН17. Уміти здійснювати збір, опрацювання, аналіз, систематизацію науково-технічної інформації, уникаючи при цьому академічної недоброчесності.

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

Вивчення кредитного модуля «Забезпечення якості ПЗ» базується на знаннях і навичках, які студенти одержали при вивченні наступних дисциплін: «Програмування», «Програмування на мові Python», «Дискретна математика», «Алгоритми і структури даних», «Математична логіка та теорія алгоритмів» навчального плану підготовки бакалаврів за спеціальністю 113 Прикладна математика.

Кредитний модуль «Забезпечення якості ПЗ» сприяє більш ефективному вивченню кредитних модулів «Чисельні методи», «Вступ до баз даних та інформаційних систем», «Операційні системи», «Методи оптимізації», «Бази даних», «Інформаційні системи», «Алгоритми та системи комп'ютерної математики» навчального плану підготовки бакалаврів за спеціальністю 113 Прикладна математика, а також сприяє більш якісному виконанню 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.** Метрики якості ПЗ. Модель якості МакКола. Модель якості Боема. Моделі FURPS / FURPS+. Інші моделі якості ПЗ. Застосування на практиці моделей якості ПЗ.

3. **Стандарти якості ПЗ.** Моделі якості і стандарти якості програмного забезпечення ISO/IEC 9126, ISO/IEC 25010:2011. Стандарт IEEE-829 тестування та тестової документації.

4. **Поліпшення якості ПЗ.** Стандарт управління якістю ISO 9001. Моделі зрілості можливостей створення програмного забезпечення CMM. Моделі CMMI, TMMI. Стандарт ISO 15504 визначення можливостей та поліпшення розробки ПЗ. Вартість якості ПЗ.

5. **Моделі життєвого циклу ПЗ.** Каскадна модель. V-подібна модель. Ітераційні моделі: інкрементна та еволюційна моделі. Спіральна модель. Гнучкі методології розробки ПЗ. Agile-маніфест. Методології Scrum, Extreme Programming.

6. **Аналіз вимог до програмного забезпечення.** Збір (виявлення) вимог. Аналіз вимог та їх узгодження: характеристики якісних вимог до ПЗ, пошук та усунення дефектів у вимогах. Специфікація вимог. Моделювання системи. Перевірка (валідація) вимог. Управління вимогами.

7. **Тестування як засіб підвищення якості ПЗ.** Основні визначення. Аксиоми тестування. Місце тестування в життєвому циклі розробки ПЗ. Ціна виправлення помилок на різних етапах ЖЦ ПЗ. Принципи тестування.

8. **Класифікація тестування програмного забезпечення.** Рівні та типи тестування. Статичне й динамічне тестування. Димове тестування та тестування критичного шляху. Позитивне й негативне тестування. Тестування вручну і автоматизоване тестування. Функціональне та нефункціональне тестування. Модульне, інтеграційне, системне тестування. Приймальне тестування.

9. **Статичний аналіз програмного коду.** Ревізії (інспекції) коду. Типи ревізій: розбір, інспектування, технічний аналіз, експертна оцінка. Метрики програмного коду: метрики розмірності, цикломатична складність коду за МакКейбом, метрики Холстеда та Чепіна складності коду, індекс підтримуваності коду. Автоматичне визначення деяких метрик програмного 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 пар/трибок значень; тестування на основі UML-діаграм.

11. **Структурне тестування ПЗ.** Методи «білого ящика»: покриття операторів; покриття гілок (рішень, альтернатив); покриття умов; комбінаторне покриття гілок / умов; покриття шляхів. Тестування потоків даних.

12. **Тестова документація.** Стандарт тестової документації IEEE-829. Тест-план, чек-ліст, специфікація вимог, матриця відповідності вимог, тест-кейси, характеристики якісного тест-кейсу, набори тест-кейсів.

13. **Звіти про дефекти.** Класифікація дефектів у програмному забезпеченні. Життєвий цикл дефекту та звіту про дефект. Інструментальні засоби управління звітами про дефекти. Властивості якісного звіту про дефект. Пріоритет та серйозність дефекту. Підсумковий звіт з тестування: мета та складові.

14. **Критерії завершення тестування.** Ступінь задоволення вимог. Рівень покриття тестами програмного коду. Частота виявлення помилок. «Підсів» помилок. Обмеження часу та бюджету проекту.

15. **Основи автоматизації тестування ПЗ.** Переваги і ризики автоматизації тестування. Необхідні знання і навички. Типи тест-кейсів, що підлягають автоматизації. Тести, що не підходять для автоматизації. Кроки реалізації автоматизованого тестування. Огляд інструментів автоматизації тестування.

16. **Основні підходи в автоматизації тестування.** Тестування на основі коду. Тестування графічного інтерфейсу користувача. Лінійні сценарії, «запис-відтворення» тестів. Функціональна декомпозиція. Тестування під керуванням даними, керуванням ключовими словами. Тестування під керуванням поведінкою.

17. **Особливості створення тест-кейсів в автоматизованому тестуванні.** Перевірка тверджень. Перевірка тверджень. Розумні локатори елементів. Очікування та очікування за умови. Керуючі структури. Повторювані кроки. Підтримка кросбраузерності. Додавання програмного коду. Створення скріншотів. Звітування. Безперервна інтеграція.

4. Навчальні матеріали та ресурси

Базова література:

1. Інженерія програмного забезпечення. Посібник для студентів вищих навчальних закладів / І. Бородкіна, Г. Бородкін. – Центр навчальної літератури, 2018. – 204 с.

2. Дідковська М.В. Технології розробки та тестування програм: Навчальний посібник / М.В. Дідковська. – ТОВ «Вік-Принт», 2015. – 344 с.

3. Якість програмного забезпечення та тестування: базовий курс / За ред. Крепич С. Я., Співак І. Я. – Тернопіль : ФОП Паляниця В.А., 2020. – 478 с.

4. Тестування програмного забезпечення : Навчальний посібник / Авраменко А. С., Авраменко В. С., Косенюк Г. В. – Черкаси : ЧНУ імені Богдана Хмельницького, 2017. – 284 с.

5. 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. — 616 p.

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. 3rd 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.

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	Якість програмного забезпечення Поняття якості. Види якості. Професійний підхід до якості ПЗ. Забезпечення якості та контроль якості. Поняття валідації та верифікації ПЗ.	2
2	Моделі якості ПЗ Метрики якості ПЗ. Модель якості МакКола. Модель якості Боема. Моделі FURPS / FURPS+. Інші моделі якості ПЗ. Застосування на практиці моделей якості ПЗ.	2

№ з/п	Назва теми лекції та перелік основних питань	Кількість ауд. годин
3	Стандарти якості. Моделі якості і стандарти якості програмного забезпечення ISO/IEC 9126, ISO/IEC 25010:2011. Стандарт IEEE-829 тестування та тестової документації.	2
4	Поліпшення якості ПЗ. Стандарт управління якістю ISO 9001. Модель зрілості можливостей створення програмного забезпечення CMM. Моделі CMMI, TMMI. Стандарт ISO 15504 визначення можливостей та поліпшення розробки ПЗ. Вартість якості ПЗ.	2
5	Моделі життєвого циклу ПЗ. Каскадна модель. V-подібна модель. Ітераційні моделі: інкрементна та еволюційна моделі. Спіральна модель. Гнучкі методології розробки ПЗ. Agile-маніфест. Методології Scrum, Extreme Programming.	2
6	Аналіз вимог до програмного забезпечення. Збір (виявлення) вимог. Аналіз вимог та їх узгодження: характеристики якісних вимог до ПЗ, пошук та усунення дефектів у вимогах. Специфікація вимог. Моделювання системи. Перевірка (валідація) вимог. Управління вимогами.	2
7	Тестування як засіб підвищення якості ПЗ. Основні визначення. Аксиоми тестування. Місце тестування в життєвому циклі розробки ПЗ. Ціна виправлення помилок на різних етапах ЖЦ ПЗ. Принципи тестування.	2
8	Класифікація тестування програмного забезпечення. Рівні та типи тестування. Статичне й динамічне тестування. Димове тестування та тестування критичного шляху. Позитивне й негативне тестування. Тестування вручну і автоматизоване тестування. Функціональне та нефункціональне тестування. Модульне, інтеграційне, системне тестування. Приймальне тестування.	2
9	Статичний аналіз ПЗ. Ревізії (інспекції) коду. Типи ревізій: розбір, інспектування, технічний аналіз, експертна оцінка. Метрики програмного коду: метрики розмірності, цикломатична складність коду за МакКейбом, метрики Холстеда та Чепіна складності коду, індекс підтримованості. Автоматичне визначення деяких метрик програмного коду.	2
10	Функціональне тестування ПЗ. Методи «чорного ящика»: розбиття на класи еквівалентності; аналіз граничних значень; тестування на основі таблиць рішень; тестування станів і переходів; тестування пар/тріоек значень; тестування на основі UML-діаграм.	2
11	Структурне тестування ПЗ. Методи «білого ящика»: покриття операторів; покриття гілок (рішень, альтернатив); покриття умов; комбінаторне покриття гілок / умов; покриття шляхів. Тестування потоків даних.	2
12	Тестова документація. Стандарт тестової документації IEEE-829. Тест-план, чек-ліст, специфікація вимог, матриця відповідності вимог, тест-кейси, характеристики якісного тест-кейсу, набори тест-кейсів.	2
13	Звітування про дефекти. Класифікація дефектів у програмному забезпеченні. Життєвий цикл дефекту та звіту про дефект. Інструментальні засоби управління звітами про дефекти. Властивості якісного звіту про дефект. Пріоритет та серйозність дефекту. Підсумковий звіт з тестування, його мета та складові.	2

№ з/п	Назва теми лекції та перелік основних питань	Кількість ауд. годин
14	Критерії завершення тестування. Ступінь задоволення вимог. Рівень покриття тестами програмного коду. Частота виявлення помилок. «Підсів» помилок. Обмеження часу та бюджету проекту.	2
15	Основи автоматизації тестування ПЗ. Переваги і ризики автоматизації тестування. Необхідні знання і навички. Типи тест-кейсів, що підлягають автоматизації. Тести, що не підходять для автоматизації. Кроки реалізації автоматизованого тестування. Огляд інструментів автоматизації тестування.	2
16	Основні підходи в автоматизації тестування. Тестування на основі коду. Тестування графічного інтерфейсу користувача. Лінійні сценарії, «запис-відтворення» тестів. Функціональна декомпозиція. Тестування під керуванням даними, керуванням ключовими словами. Тестування під керуванням поведінкою.	2
17	Особливості створення тест-кейсів в автоматизованому тестуванні. Перевірка тверджень. Розумні локатори елементів. Очікування та очікування за умови. Керуючі структури. Повторювані кроки. Підтримка кросбраузерності. Додавання програмного коду. Створення скріншотів. Звітування. Безперервна інтеграція.	2

5.2. Лабораторні заняття

№ з/п	Назва та завдання лабораторної роботи	Кількість ауд. годин
1	Виявлення помилок в ПЗ. Створення простих тестових випадків Завдання: навчитись розробляти тестові випадки для тестування заданого ПЗ відповідно до прийнятих стандартів, визначати тестові процедури для проведення тестування.	4
2	Аналіз вимог до програмного забезпечення Завдання: навчитись аналізувати вимоги до програмного забезпечення, знаходити та класифікувати дефекти у вимогах.	4
3	Статичний аналіз коду. Метрики якості програмного коду. Бібліотека Radon Python Завдання: навчитись виконувати статичний аналіз коду, визначати метрики програмного коду: метрики розміру програм, цикломатичну складність за МакКейбом, метрики Холстеда, індекс підтримуваності коду.	4
4	Тестування «чорного ящика» Завдання: навчитись розробляти тест-кейси на основі розбиття на класи еквівалентності, аналізу граничних значень, аналізу таблиць рішень, діаграм станів і переходів.	4
5	Тестування «білого ящика» 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	Документування тест-кейсів. Звітування про дефекти Завдання: навчитись створювати якісні тест-кейси та звіти про дефекти, а також документувати їх відповідно до прийнятих стандартів.	4
7	Автоматизоване тестування програм на мові Python Завдання: опанувати бібліотеки Python unittest, subprocess; навчитись розробляти і виконувати автоматизацію тестування на мові Python.	6
8	Автоматизоване тестування веб-сторінок та веб-застосунків за допомогою Selenium IDE, Selenium WebDriver та Python 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	Якість програмного забезпечення. Проаналізувати місце забезпечення якості ПЗ в процесі розроблення ПЗ	3
2	Моделі якості ПЗ. Детальніше ознайомитися з питаннями використання моделей якості МакКола, Боема, FURPS.	3
3	Стандарти якості. Опрацювати стандарти якості програмного забезпечення ISO/IEC 9126, ISO/IEC 25010:2011, стандарт тестування IEEE-829.	3
4	Поліпшення якості ПЗ. Опрацювати стандарт управління якістю ISO 9001, моделі CMMI, TMMI, стандарт ISO 15504, розглянути модель TPI Next.	3
5	Моделі життєвого циклу ПЗ. Опрацювати традиційні моделі ЖЦ ПЗ та гнучкі методології розробки. Розібрати Agile-маніфест розробки ПЗ.	3
6	Аналіз вимог до програмного забезпечення. Опрацювати критерії якісних вимог та можливі шляхи усунення дефектів у вимогах до програмного забезпечення.	3
7	Тестування як засіб підвищення якості ПЗ. Опрацювати принципи тестування. Ознайомитися з основними принципами психології тестувальника. Підготуватися до МКР-1.	3
8	Класифікація тестування програмного забезпечення. Опрацювати детальну класифікацію тестування. Опрацювати типи приймального тестування.	3
9	Статичний аналіз ПЗ. Опрацювати типи ревізій програмного коду, розглянути чинники успішних ревізій.	3
10	Функціональне тестування ПЗ. Розібратися з тестуванням сценаріїв використання. Опрацювати тестуванням пар/трійок значень.	3
11	Структурне тестування ПЗ. Детальніше розібратися з тестуванням гілок, умов, комбінаторним тестуванням гілок / умов.	2
12	Тестова документація. Розібратися з перевагами створення матриці відповідності. Опрацювати різні варіанти запису тест-кейсів.	2
13	Звітування про дефекти. Розглянути життєвий цикл дефекту в різних баг-трекінгових системах.	3
14	Критерії завершення тестування. Опрацювати види активностей тестувальників на етапі завершення тестування	2
15	Основи автоматизації тестування ПЗ. Розібратися з перевагами й вадами використання засобів автоматизації виконання тестування. Опрацювати бібліотеку unittest Python	3
16	Основні підходи в автоматизації тестування. Опрацювати ідеологію тестування під керуванням ключовими словами на прикладі Selenium IDE.	3
17	Особливості створення тест-кейсів в автоматизованому тестуванні. Розібратися з помилками, які можуть виникати при підготовці тест-кейсів для автоматизованого тестування та правильними варіантами запису тест-кейсів. Підготуватися до МКР-2.	3

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.

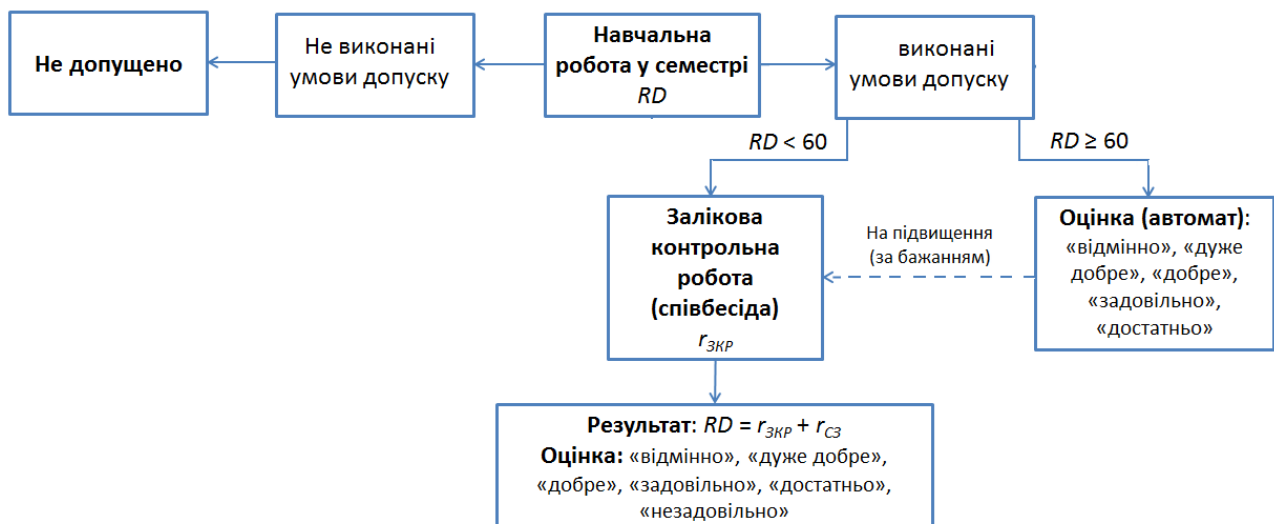


Рис. 1. Схема функціонування PCO

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) Статичний аналіз коду. Метрики 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-сторінок та веб-застосунків 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;
- вчасність 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 вчасності 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 запитання: **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 усіх лабораторних робіт. Студенти, які допущені до заліку, отримують оцінку з дисципліни R_c автоматично.

Якщо студент не згоден із попередньою оцінкою, яку він отримав «автоматом», то він пише залікову контрольну роботу. Остаточний результат R_c 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_c , 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

Робочу програму навчальної дисципліни (силабус):

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

Ухвалено кафедрою прикладної математики (протокол № 10 від 02.01.2024)

Погоджено Методичною комісією факультету прикладної математики (протокол № 6 від 26.01.2024)