

СИЛАБУС ОСВІТНЬОЇ КОМПОНЕНТИ «ПРИКЛАДНЕ ПРОГРАМУВАННЯ RUST»

Лектор курсу			Гавор Артур Станіславович		Контактна інформація лектора (e-mail), сторінка курсу в Google Classroom	e-mail: a.havor@duikt.edu.ua сторінка курсу в Google Classroom – https://classroom.google.com/c/ODAyMTMwNTU3MzUw?cjc=sfq5fwrp	
Галузь знань					Рівень вищої освіти	бакалавр	
Спеціальність					Семестр		
Освітня програма					Тип дисципліни	Вибіркова	
Обсяг:	Кредитів ECTS	Годин	За видами занять:				
			Лекцій	Семінарських занять	Практичних занять	Лабораторних занять	Самостійна підготовка
	5	150	18	-	18	18	96

АНОТАЦІЯ КУРСУ

Взаємозв'язок у структурно-логічній схемі

Освітні компоненти, які передують вивченню		
Освітні компоненти для яких є базовою		
Мета курсу:	Формування знань та практичних навичок програмування мовою Rust, яка є сучасним інструментом для створення високопродуктивних та безпечних програмних систем. Дисципліна спрямована на забезпечення розуміння концепцій управління пам'яттю, системи типів та механізмів безпеки, які закладені у Rust, а також їх застосування для вирішення прикладних завдань.	

Компетентності відповідно до освітньої програми

Soft- skills / Загальні компетентності (ЗК)	Hard-skills / Спеціальні компетентності (СК)
	ПП 1. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення. ПП 3. Здатність розробляти архітектури, модулі та компоненти програмних систем. ПП 4. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами. ПП 7. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних. ПП 11. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення. ПП 14. Здатність до алгоритмічного та логічного мислення.

Програмні результати навчання (ПР)

ПРН 6. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.

ПРН 9. Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.

ПРН 13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

ПРН 18. Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.

ПРН 20. Знати підходи щодо оцінки та забезпечення якості програмного забезпечення.

ПРН 21. Знати, аналізувати, вибирати, кваліфіковано застосовувати засоби забезпечення інформаційної безпеки (в тому числі кібербезпеки) і цілісності даних відповідно до розв'язуваних прикладних завдань та створюваних програмних систем.

ОРГАНІЗАЦІЯ НАВЧАННЯ			
Тема, опис теми	Вид заняття	Оцінювання за тему	Форми і методи навчання/питання до самостійної роботи
Тема 1. Особливості Rust в порівнянні із іншими мовами програмування. <u>Рекомендовані джерела:</u> 1, 2, 3, 4, 6, 7, 10.			
Заняття 1.1 Теоретичні відмінності від інших мов програмування та основні характеристики Rust. Система запозичень та власності.	Лекція 1 2 год		Пояснювально-ілюстративний, лекція-візуалізація, бліц опитування, розбір прикладів.
Заняття 1.2 Розбір теоретичних питань та понять Rust. Визначення гарантій, що надає компілятор.	Практичне заняття 1 2 год	3 бали	Усне опитування, виконання практичного завдання для закріплення матеріалу.
Заняття 1.3. Дослідження видів систем типів популярних мов програмування. Статична та динамічна типізація. Слабка та сильна типізація.	Лабораторн е заняття 1 2 год	2 бали	Усне опитування, виконання лабораторного завдання для закріплення матеріалу.
Заняття 1.4. Статична типізація Rust, її переваги, та практичне застосування.	Лекція 2 2 год		Пояснювально-ілюстративний, лекція-візуалізація, бліц опитування, розбір прикладів.
Заняття 1.5 Забезпечення коректності ПЗ системою типів Rust.	Практичне заняття 2 2 год	3 бали	Усне опитування, виконання практичного завдання для закріплення матеріалу.
Заняття 1.6. Написання примітивного проекту використовуючи принципи забезпечення коректності поведінки системою типів.	Лабораторн е заняття 2 2 год	2 бали	Усне опитування, виконання лабораторного завдання для закріплення матеріалу.
Заняття 1.7. Абстрагування поведінки. Види поліморфізму в Rust.	Лекція 3 2 год		Пояснювально-ілюстративний, лекція-візуалізація, бліц опитування, розбір прикладів.
Заняття 1.8 Динамічний та статичний поліморфізм у Rust та його застосування для побудови гнучкої архітектури ПЗ.	Практичне заняття 3 2 год	3 бали	Усне опитування, виконання практичного завдання для закріплення матеріалу.

Заняття 1.9. Розширення програми із попередньої лабораторної новим функціоналом та застосування динамічного поліморфізму.	Лабораторн е заняття 3 2 год	2 бали	Усне опитування, виконання лабораторного завдання для закріплення матеріалу.
Самостійна робота			
Порівняння обробки помилок типом Result в Rust та виключеннями у Java/Js/C#/Python/C++ (на вибір).	10 год.	1 бал	Дослідити та порівняти підходи порівняння помилок шляхом типу Result у Rust та шляхом виключень (exceptions) у Java/Js/C#/Python/C++ (на вибір).
Переваги та недоліки типізацій популярних мов програмування.	10 год.	2 бал	Дослідити та порівняти слабку та сильну, статичну та динамічну типізації у популярних мовах програмування. Оцінити переваги на недоліки різних типізацій.
Порівняння способів поліморфізму із іншими популярними мовами програмування (на вибір).	12 год.	2 бал	Дослідити та порівняти способи реалізації поліморфізму із іншою мовою програмування (на вибір). Порівняння реалізації узагальнених типів із іншою мовою програмування (на вибір).
Тема 2. Екосистема Rust. Рекомендовані джерела: 1, 2, 3, 6, 8, 9.			
Заняття 2.1 Загальні підходи до обробки помилок в Rust.	Лекція 4 2 год		Пояснювально-ілюстративний, лекція-візуалізація, бліц опитування, розбір прикладів.
Заняття 2.2 Написання надійного ПЗ із стійкою обробкою помилок.	Практичне заняття 4 2 год	3 бали	Усне опитування, виконання практичного завдання для закріплення матеріалу.
Заняття 2.3 Додавання обробки помилок до проєкту із попередньої лабораторної роботи.	Лабораторн е заняття 4 2 год	2 бали	Усне опитування, виконання лабораторного завдання для закріплення матеріалу.
Заняття 2.4 Методи логування та конфігурації ПЗ в Rust.	Лекція 5 2 год		Пояснювально-ілюстративний, лекція-візуалізація, бліц опитування, розбір прикладів.
Заняття 2.5 Реалізація гнучкого логування та конфігурування ПЗ за допомогою Rust.	Практичне заняття 5 2 год	3 бали	Усне опитування, виконання практичного завдання для закріплення матеріалу.
Заняття 2.6 Імплементація логування та конфігурування проєкту із попередньої лабораторної роботи.	Лабораторн е заняття 5 2 год	2 бали	Усне опитування, виконання лабораторного завдання для закріплення матеріалу.

Заняття 2.7 Способи та види тестування ПЗ. Основні підходи до тестування в екосистемі Rust.	Лекція 6 2 год		Пояснювально-ілюстративний, лекція-візуалізація, бліц опитування, розбір прикладів.
Заняття 2.8 Написання юніт, інтеграційних, та E2E тестів на мові програмування Rust.	Практичне заняття 6 2 год	3 бали	Усне опитування, виконання практичного завдання для закріплення матеріалу.
Заняття 2.9 Написання E2E тестів до CLI застосунків.	Лабораторн е заняття 6 2 год	2 бали	Усне опитування, виконання лабораторного завдання для закріплення матеріалу.
Заняття 2.10 Інструменти для полегшення підтримки Rust коду: форматувальник, лінтер, оформлення програмної документації.	Лекція 7 2 год		Пояснювально-ілюстративний, лекція-візуалізація, бліц опитування, розбір прикладів.
Заняття 2.11 Форматування коду, застосування лінтів, та написання програмної документації.	Практичне заняття 7 2 год	3 бали	Усне опитування, виконання практичного завдання для закріплення матеріалу.
Заняття 2.12 Покращення та рефакторинг проєкту із попередньої лабораторної роботи інструментами екосистеми Rust.	Лабораторн е заняття 7 2 год	2 бали	Усне опитування, виконання лабораторного завдання для закріплення матеріалу.
Самостійна робота			
Порівняння способів та можливості логування та конфігурування у Rust та у Java/Js/C#/Python/C++ (на вибір).	10 год.	1 бал	Дослідити та порівняти підходи логування та можливості конфігурування у Rust та у Java/Js/C#/Python/C++ (на вибір).
Переваги інструменту rustdoc для написання програмної документації із альтернативним інструментом у Java/Js/C#/Python/C++ (на вибір).	10 год.	2 бал	Дослідити та порівняти інструменти для написання програмної документації у Rust та у Java/Js/C#/Python/C++ (на вибір). Переваги та недоліки підходу закладеного у rustdoc.
Переваги інструментів clippy та rustfmt із альтернативними інструментами у Java/Js/C#/Python/C++ (на вибір).	12 год.	2 бал	Дослідити можливості та порівняти інструменти для підтримки та рефакторингу коду у Rust та у Java/Js/C#/Python/C++ (на вибір). Переваги та недоліки clippy та rustfmt.
Тема 3. Основи метапрограмування в Rust. Знати: поняття метапрограмування, які проблеми вирішує, та варіанти застосування, види та відмінності Rust макросів. Вміти: використовувати процедурні та декларативні макроси для вирішення основних задач метапрограмування. Рекомендовані джерела: 1, 3, 5.			
Заняття 3.1 Метапрограмування, його застосування та реалізація у мові Rust.	Лекція 8 2 год		Пояснювально-ілюстративний, лекція-візуалізація, бліц опитування, розбір прикладів.
Заняття 3.2 Написання примітивних процедурних та декларативних макросів у Rust.	Практичне заняття 8 2 год	3 бали	Усне опитування, виконання практичного завдання для закріплення матеріалу.
Заняття 3.3 Дослідження можливостей декларативних макросів.	Лабораторн е заняття 8 2 год	2 бали	Усне опитування, виконання лабораторного завдання для закріплення матеріалу.

Заняття 3.4 Процедурні макроси в Rust. Патерни та антипатерни.	Лекція 9 2 год		Пояснювально-ілюстративний, лекція-візуалізація, бліц опитування, розбір прикладів.
Заняття 3.5 Написання процедурних макросів.	Практичне заняття 9 2 год	3 бали	Усне опитування, виконання практичного завдання для закріплення матеріалу.
Заняття 3.6 Дослідження можливостей процедурних макросів.	Лабораторн е заняття 9 2 год	2 бали	Усне опитування, виконання лабораторного завдання для закріплення матеріалу.
Самостійна робота			
Порівняння підходів до метапрограмування у Rust та у Java/Js/C#/Python/C++ (на вибір).	10 год.	1 бал	Дослідити та порівняти підходи та можливості метапрограмування у Rust та у Java/Js/C#/Python/C++ (на вибір).
Безпека процедурних макросів у Rust.	10 год.	2 бал	Дослідження небезпеки виконання процедурних макросів. Способи захисту від зловмисних процедурних макросів.
Проблеми макросів у Rust.	12 год.	2 бал	Дослідити найпоширеніші проблеми макросів у Rust: читабельність коду, підтримуваність, зворотня сумісність.
МАТЕРІАЛЬНО-ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ ДИСЦИПЛІНИ			
• Мультимедійний проектор. • Комп'ютерний клас для проведення лабораторних та лекційних занять. • Програмне забезпечення: Visual Studio Code із розширенням rust-analyzer або JetBrains RustRover (додаткові розширення не потрібні).			
ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ДИСЦИПЛІНИ			
1. Martin Geisler. Comprehensive Rust. URL: https://google.github.io/comprehensive-rust/comprehensive-rust.pdf. 2. The Rust Programming Language: [Електронний ресурс]. – Режим доступу: https://doc.rust-lang.org/book/. 3. Rust By Example: [Електронний ресурс]. – Режим доступу: https://doc.rust-lang.org/stable/rust-by-example/. 4. Rustlings: [Електронний ресурс]. – Режим доступу: https://rustlings.rust-lang.org/. 5. The Little Book of Rust Macros: [Електронний ресурс]. – Режим доступу: https://lukaswirth.dev/tlborm/. 6. The rustdoc book: [Електронний ресурс]. – Режим доступу: https://doc.rust-lang.org/rustdoc/. 7. Rust Design Patterns: [Електронний ресурс]. – Режим доступу: https://rust-unofficial.github.io/patterns/. 8. Effective Rust: 35 Specific Ways to Improve Your Rust Code: [Електронний ресурс]. – Режим доступу: https://www.lurklurk.org/effective-rust/. 9. The Rust Performance Book: [Електронний ресурс]. – Режим доступу: https://nnethercote.github.io/perf-book/title-page.html. 10. Welcome to Rust-101: [Електронний ресурс]. – Режим доступу: https://www.ralfj.de/projects/rust-101/main.html.			
ПОЛІТИКА КУРСУ («ПРАВИЛА ГРИ»)			
• Середовище в аудиторії є дружнім, творчим, відкритим до конструктивної критики. • Освоєння дисципліни передбачає обов'язкове відвідування лекцій і лабораторних занять, а також самостійну роботу. • Самостійна робота включає в себе теоретичне вивчення питань, що стосуються тем лекційних занять, які не ввійшли в теоретичний курс, або ж були розглянуті коротко, їх поглиблена проробка за рекомендованою літературою. • Усі завдання, передбачені програмою, мають бути виконані у встановлений термін. • Якщо студент відсутній з поважної причини, він презентує виконані завдання під час самостійної підготовки та консультації викладача. • Під час роботи над завданнями не допустимо порушення академічної доброчесності: при використанні Інтернет ресурсів та інших джерел інформації студент повинен вказати джерело, використане в ході виконання завдання. У разі виявлення факту плагіату студент отримує за завдання 0 балів. • Код роботи студента має зберігатися у закритому GitHub репозиторії студента, доступ до якого має лише викладач та сам студент.			

- Роботи здаються у вигляді Pull Requests у приватний репозиторій. Критерієм прийняття роботи є approve на платформі GitHub. Якщо посилання завантажено у Classroom, а код відсутній – робота буде оцінена в 0 балів.

КРИТЕРІЇ ТА МЕТОДИ ОЦІНЮВАННЯ

Умовою допуску до підсумкового контролю є виконання всіх практичних робіт і виконання самостійних завдань, які передбачені структурою освітньої компоненти Прикладне програмування Rust.

Якщо студента не допущено до складання заліку, як такого, що не виконав індивідуальний план, йому надається час до перескладання для виконання всіх вимог допуску. Студент має право на два перескладання. При повторному перескладанні екзамену його у студента може приймати комісія, яка створюється директором ННІТ. Оцінка комісії є остаточною. У випадку отримання студентом 0 балів (неприйнятно), що тягне відрахування за невиконання навчального плану.

Оцінювання студентів здійснюється за накопичувальною 100-бальною системою і складається із двох основних оцінкових блоків і розподіляється в певних пропорціях 60 (бали напрацьовані під час вивчення дисципліни – Поточний контроль), 40 (підсумкове оцінювання - Екзамен):

Форми контролю	Види навчальної роботи	Оцінювання
ПОТОЧНИЙ КОНТРОЛЬ	● Виконання практичних робіт	45 балів
	● Самостійна робота	15 балів
ПІДСУМКОВЕ ОЦІНЮВАННЯ Екзамен	Екзамен проходить в письмовій формі.	40 балів

Додаткова оцінка

Види навчальної роботи	Оцінювання
Проходження Rustcamp від Української Rust спільноти (https://github.com/rust-lang-ua/rustcamp): Якщо студент уже проходив цей курс і по ньому були нараховані додаткові бали, то другий раз бали не зараховуються.	
- Завершення базової частини Rustcamp. Студент повинен мати сертифікат про закінчення курсу та успішно складений екзаменаційний тест.	5 балів
- Завершення базової частини Rustcamp + web спеціалізація. Студент повинен мати сертифікат про закінчення курсу та успішно складений екзаменаційний тест.	10 балів
Проект написаний мовою програмування Rust:	
- Проект включає роботу із файлами/мережею/базою даних, асинхронне виконання, налаштоване логування, гнучка обробка помилок та містить проектну документацію.	5 балів
- Проект включає роботу із файлами/мережею/базою даних, асинхронне виконання, налаштоване логування, гнучка обробка помилок та містить проектну документацію. Налаштоване середовище CI/CD, більшість логіки покрита тестами, основна поведінка винесена в окремий крейт, використання патернів Rust.	10 балів

Максимальна кількість додаткових балів, які можуть бути зараховані здобувачу освіти - 10 балів.

ПІДСУМКОВА ОЦІНКА ЗА ДИСЦИПЛІНУ

бали	Критерії оцінювання	Рівень компетентності	Оцінка /запис в екзаменаційній відомості
90-100	Студент демонструє повні й міцні знання навчального матеріалу в обсязі, що відповідає робочій програмі дисципліни, правильно й обґрунтовано приймає необхідні рішення в різних нестандартних ситуаціях. Вміє реалізувати теоретичні положення дисципліни в практичних розрахунках, аналізувати та співставляти дані об'єктів діяльності фахівця на основі	Високий Повністю забезпечує вимоги до знань, умінь і навичок, що викладені в робочій програмі дисципліни. Власні пропозиції студента в оцінках і вирішенні	Відмінно / Зараховано (А)

	набутих з даної та суміжних дисциплін знань та умінь. Знає сучасні технології та методи розрахунків з даної дисципліни. За час навчання при проведенні практичних занять, при виконанні індивідуальних / контрольних завдань проявив вміння самостійно вирішувати поставлені завдання, активно включатись в дискусії, може відстоювати власну позицію в питаннях та рішеннях, що розглядаються. Зменшення 100-бальної оцінки може бути пов'язане з недостатнім розкриттям питань, що стосується дисципліни, яка вивчається, але виходить за рамки об'єму матеріалу, передбаченого робочою програмою, або студент проявляє невпевненість в тлумаченні теоретичних положень чи складних практичних завдань.	практичних задач підвищує його вміння використовувати знання, які він отримав при вивченні інших дисциплін, а також знання, набуті при самостійному поглибленому вивченні питань, що відносяться до дисципліни, яка вивчається.	
82-89	Студент демонструє гарні знання, добре володіє матеріалом, що відповідає робочій програмі дисципліни, робить на їх основі аналіз можливих ситуацій та вміє застосовувати теоретичні положення при вирішенні практичних задач, але допускає окремі неточності. Вміє самостійно виправляти допущені помилки, кількість яких є незначною. Знає сучасні технології та методи розрахунків з даної дисципліни. За час навчання при проведенні практичних занять, при виконанні індивідуальних / контрольних завдань та поясненні прийнятих рішень, дає вичерпні пояснення.	Достатній Забезпечує студенту самостійне вирішення основних практичних задач в умовах, коли вихідні дані в них змінюються порівняно з прикладами, що розглянуті при вивченні дисципліни	Добре / Зараховано (B)
75-81	Студент в загальному добре володіє матеріалом, знає основні положення матеріалу, що відповідає робочій програмі дисципліни, робить на їх основі аналіз можливих ситуацій та вміє застосовувати при вирішенні типових практичних завдань, але допускає окремі неточності. Вміє пояснити основні положення виконаних завдань та дати правильні відповіді при зміні результату при заданій зміні вихідних параметрів. Помилки у відповідях/ рішеннях/ розрахунках не є системними. Знає характеристики основних положень, що мають визначальне значення при проведенні практичних занять, при виконанні індивідуальних / контрольних завдань та поясненні прийнятих рішень, в межах дисципліни, що вивчається.	Достатній Конкретний рівень, за вивченим матеріалом робочої програми дисципліни. Додаткові питання про можливість використання теоретичних положень для практичного використання викликають утруднення.	Добре / Зараховано (C)
67-74	Студент засвоїв основний теоретичний матеріал, передбачений робочою програмою дисципліни, та розуміє постанову стандартних практичних завдань, має пропозиції щодо напрямку їх вирішень. Розуміє основні положення, що є визначальними в курсі, може вирішувати подібні завдання тим, що розглядалися з викладачем, але допускає значну кількість неточностей і грубих помилок, які може усувати за допомогою викладача.	Середній Забезпечує достатньо надійний рівень відтворення основних положень дисципліни	Задовільно / Зараховано (D)
60-66	Студент має певні знання, передбачені в робочій програмі дисципліни, володіє основними положеннями, що вивчаються на рівні, який визначається як мінімально допустимий. З використанням основних теоретичних положень, студент з труднощами пояснює правила вирішення практичних/розрахункових завдань дисципліни. Виконання практичних / індивідуальних / контрольних завдань значно формалізовано: є відповідність алгоритму, але відсутнє глибоке розуміння роботи та взаємозв'язків з іншими дисциплінами.	Середній Є мінімально допустимим у всіх складових навчальної програми з дисципліни	Задовільно / Зараховано (E)
35-59	Студент може відтворити окремі фрагменти з курсу.	Низький	Незадовільно з

	Незважаючи на те, що програму навчальної дисципліни студент виконав, працював він пасивно, його відповіді під час практичних робіт в більшості є невірними, необґрунтованими. Цілісність розуміння матеріалу з дисципліни у студента відсутні.	Не забезпечує практичної реалізації задач, що формуються при вивченні дисципліни	можливістю повторного складання) / Не зараховано (FX) В залікову книжку не проставляється
0-34	Студент повністю не виконав вимог робочої програми навчальної дисципліни. Його знання на підсумкових етапах навчання є фрагментарними. Студент не допущений до здачі екзамену/заліку.	Незадовільний Студент не підготовлений до самостійного вирішення задач, які окреслює мета та завдання дисципліни	Незадовільно з обов'язковим повторним вивченням / Не допущений (F) В залікову книжку не проставляється

ПОЛІТИКА ДОБРОЧЕСНОСТІ

Здобувач вищої освіти виконуючи самостійну або індивідуальну роботу повинен дотримуватись політики доброчесності. У разі наявності плагіату в будь-яких видах робіт Здобувача він отримує незадовільну оцінку і повинен повторно виконати завдання, які передбачені у Силабусі.