



**ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

Кафедра кібербезпеки та програмного забезпечення



**СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ПРОГРАМУВАННЯ МОВОЮ RUST**

першого рівня вищої освіти

м. Кропивницький

1. Загальна інформація

Назва дисципліни	Програмування мовою Rust
Викладач	Лектор – Минайленко Роман Миколайович, кандидат технічних наук, доцент, доцент кафедри кібербезпеки та програмного забезпечення
Контактний телефон	службовий: (0522)390-449 – робочі дні з 8 ³⁰ до 14 ²⁰
E-mail:	aron70@ukr.net
Консультації	<i>Очні консультації</i> відповідно до затвердженого графіку консультацій <i>Онлайн консультації</i> засобами електронної пошти, месенджерів (Facebook-Messenger / Viber / Telegram) у робочі дні

2. Анотація дисципліни

Курс «Програмування мовою Rust» спрямована на формування у здобувачів вищої освіти сучасних компетентностей з розроблення програмного забезпечення на мові Rust — високопродуктивній системній мові програмування, орієнтованій на безпечну роботу з пам'яттю та надійність програм. У межах курсу розглядаються базові та поглиблені можливості мови Rust, принципи побудови програм, організація проєктів у Cargo, модульність, тестування та використання бібліотек (crates). Дисципліна охоплює ключові особливості Rust: систему власності (ownership), запозичення (borrowing), життєві цикли (lifetimes), роботу з типами даних, узагальнення (generics), трейт-орієнтоване програмування, обробку помилок та засоби конкурентного/паралельного програмування. Практична складова курсу передбачає розроблення прикладних програм і невеликих проєктів, що демонструють використання Rust для створення ефективних і безпечних системних та прикладних рішень.

3. Мета і завдання дисципліни

Метою викладання дисципліни «Програмування мовою Rust» є формування у здобувачів вищої освіти теоретичних знань і практичних навичок програмування мовою Rust, необхідних для розроблення надійного та продуктивного програмного забезпечення із застосуванням сучасних інструментів, підходів до модульної розробки, тестування та безпечної роботи з пам'яттю.

Основними **завданнями** вивчення дисципліни є опанування здобувачами синтаксису та базових конструкцій мови Rust, а також формування практичних навичок створення програм і керування проєктами з використанням Cargo. Курс передбачає засвоєння принципів безпечної роботи з пам'яттю, зокрема механізмів ownership/borrowing і життєвих циклів, а також набуття умінь реалізовувати модульні програми із застосуванням структур даних, узагальнень і трейтів. Окремим завданням дисципліни є формування навичок обробки помилок, тестування програмного коду, використання стандартної бібліотеки й зовнішніх crate-бібліотек, а також застосування підходів конкурентного програмування під час розв'язання прикладних задач.

4. Результати навчання

У результаті вивчення дисципліни здобувач вищої освіти повинен:

- Пояснювати ключові принципи мови Rust та її переваги для розроблення безпечного й продуктивного програмного забезпечення.
- Розробляти програми мовою Rust із використанням базових конструкцій (типи даних, оператори, умовні переходи, цикли, функції).
- Використовувати механізми ownership, borrowing та lifetimes для організації безпечної роботи з пам'яттю без витоків та data race.
- Проектувати модульну структуру програми, працювати з пакетами та залежностями за допомогою Cargo, підключати зовнішні бібліотеки (crates).
- Застосовувати структури, переліки, шаблони (pattern matching) і стандартні колекції для реалізації прикладних задач.
- Реалізовувати обробку помилок та виняткових ситуацій із використанням Result/Option, створювати надійний код та виконувати його тестування.
- Використовувати підходи конкурентного програмування (потoki, синхронізація, канали повідомлень) та оцінювати ефективність реалізованих рішень.

5. Обсяг дисципліни

Ознака дисципліни	
Кількість кредитів / годин	4/120
Нормативна / вибіркова	вибіркова
Вид підсумкового контролю	залік

6. Політика дисципліни

Академічна доброчесність:

Очікується, що студенти будуть дотримуватися принципів академічної доброчесності, усвідомлювати наслідки її порушення. Детальніше за посиланням URL : <http://www.kntu.kr.ua/doc/dobro.pdf>

Відвідування занять

Відвідування занять є важливою складовою навчання. Очікується, що всі студенти відвідають лекції і лабораторні заняття курсу.

Пропущені заняття повинні бути відпрацьовані не пізніше, ніж за тиждень до залікової сесії.

Поведінка на заняттях

Недопустимість: запізнь на заняття, списування та плагіат, несвоєчасне виконання поставленого завдання.

При організації освітнього процесу в Центральнoукраїнському національному технічному університеті студенти, викладачі та адміністрація діють відповідно до: Положення про організацію освітнього процесу; Положення про організацію вивчення навчальних дисциплін вільного вибору; Положення про рубіжний контроль успішності і сесійну атестацію студентів ЦНТУ, Кодексу академічної доброчесності ЦНТУ.

7. Програма навчальної дисципліни

Тема 1. Вступ до Rust та екосистеми розробки

Призначення Rust. Переваги безпечної роботи з пам'яттю. Інструменти: rustc, Cargo, rustup.

Тема 2. Основи синтаксису та типи даних

Змінні, область видимості, типи, приведення типів. Оператори. Робота з рядками.

Тема 3. Управляючі конструкції та функції

Умови (if), цикли (loop/while/for). Функції, параметри, повернення значень. Вирази та блоки коду.

Тема 4. Ownership і модель керування пам'яттю

Правила власності. Move/Copy. Робота зі стеком і купою. Drop і звільнення ресурсів.

Тема 5. Borrowing, references та lifetimes

Позичання (immutable/mutable). Правила посилань. Концепція життєвих циклів та їх застосування.

Тема 6. Структури, переліки та pattern matching

Struct/enum. Вбудовані методи impl. Match, if let, while let. Практичне моделювання даних.

Тема 7. Колекції стандартної бібліотеки

Vec, String, HashMap. Ітерації, пошук, модифікація. Робота з ownership у колекціях.

Тема 8. Модулі, пакети та використання crates

Модульність проєктів. Публічність (pub). Управління залежностями. Документація й структура бібліотек.

Тема 9. Узагальнення (generics) і трейти

Generic-типи та функції. Трейти як інтерфейси. Trait bounds. Реалізація поліморфізму.

Тема 10. Обробка помилок та надійність коду

Option/Result. unwrap/expect. Оператор ?. Створення власних помилок. Підхід до безпечного коду.

Тема 11. Введення/виведення та робота з файлами

stdin/stdout. Файлова система. Читання/запис файлів. Серіалізація даних (огляд serde).

Тема 12. Основи паралельного та конкурентного програмування

Потоки (threads). Mutex, Arc. Канали (mpsc). Безпечна конкурентність Rust. Типові сценарії.

8. Система оцінювання та вимоги

Види контролю: поточний, підсумковий.

Методи контролю: спостереження за навчальною діяльністю, усне опитування, письмовий контроль, тестовий контроль.

Форма підсумкового контролю: залік.

Поточний контроль у межах дисципліни здійснюється через оцінювання виконання кожної лабораторної роботи. Під час виставлення балів враховуються коректність розв'язання поставленого завдання, рівень підготовки та аргументованість захисту результатів, а також дотримання визначених строків виконання. Додатково оцінюється ступінь засвоєння теоретичних положень і сформованість практичних умінь.

Підсумковий контроль проводиться у формі заліку та спрямований на перевірку рівня опанування навчального матеріалу і здатності здобувача застосовувати знання під час виконання практичних завдань.

9. Рекомендована література

Базова

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. Asynchronous Programming in Rust: [Електронний ресурс]. – Режим доступу: <https://rust-lang.github.io/async-book/>.

Допоміжна

7. Mara Bos. Rust Atomics and Locks: [Електронний ресурс]. – Режим доступу: <https://marabos.nl/atomics/>.
8. The rustdoc book: [Електронний ресурс]. – Режим доступу: <https://doc.rust-lang.org/rustdoc/>.
9. Rust Design Patterns: [Електронний ресурс]. – Режим доступу: <https://rust-unofficial.github.io/patterns/>.
10. Effective Rust: 35 Specific Ways to Improve Your Rust Code: [Електронний ресурс]. – Режим доступу: <https://www.lurklurk.org/effective-rust/>.

Інформаційні ресурси

11. Онлайн-курси Prometheus. – URL: <https://prometheus.org.ua/>
12. Онлайн-курси Coursera. – URL: <https://www.coursera.org>
13. Академія Cisco. – URL: <https://www.netacad.com>
14. Он-лайн ресурс з інформаційних технологій. – URL: <https://habr.com>
15. Он-лайн ресурс з інформаційних технологій. – URL: <https://dou.ua/>
16. Пошукова система. – URL: <https://www.google.com/>
17. Он-лайн ресурс перегляду відеоуроків. – URL: <https://www.youtube.com>