

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
«ПРИКАРПАТСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАСИЛЯ СТЕФАНІКА»**

Фізико-технічний факультет
Кафедра комп'ютерної інженерії та електроніки

**СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
Програмування (Rust)**

Рівень вищої освіти: перший (бакалаврський)
Освітня програма: Комп'ютерне проектування інтегральних схем
Галузь знань: 17 Електроніка та телекомунікації
Спеціальність: 171 Електроніка

Затверджено на засіданні кафедри
Протокол № 12 від “30” червня 2023 р.

Івано-Франківськ – 2023 рік

ЗМІСТ

1. Загальна інформація
2. Анотація до курсу
3. Мета та цілі курсу
4. Компетентності
5. Результати навчання
6. Організація навчання курсу
7. Система оцінювання курсу
8. Політика курсу
9. Рекомендована література

1. Загальна інформація про викладача і дисципліну

Назва дисципліни	Програмування (Rust)
Рівень вищої освіти	Перший рівень вищої освіти
Викладач	Доцент, кандидат технічних наук Голота Віктор Іванович
Контактний телефон викладача	(0342) 59-60-07
Е-mail викладача	victor.holota@pnu.edu.ua
Формат дисципліни	Односеместровий
Обсяг дисципліни	6 кредитів
Посилання на сайт дистанційного навчання	http://www.d-learn.pu.if.ua
Консультації	Відповідно до графіку індивідуальних консультацій, який розміщений на інформаційному стенді кафедри комп'ютерної інженерії та електроніки

2. Анотація до курсу

Дисципліна “Програмування (Rust)” належить до переліку вибіркових навчальних дисциплін підготовки за освітнім рівнем «бакалавр», що пропонуються в рамках циклу вибіркових дисциплін професійної підготовки студентів за освітньо-професійною програмою “Комп’ютерне проектування інтегральних схем”. Вона забезпечує формування у студентів спеціальних (професійно-орієнтованих) компетенцій. Предметом вивчення навчальної дисципліни є синтаксис та семантика мови програмування Rust, розроблення та налагодження програм на мові Rust, особливості імперативних, процедурних і об’єктно-орієнтованих методів з такими парадигмами, як функційне програмування і модель акторів, а також узагальнене програмування і метапрограмування, у статичних і динамічних стилях.

Силабус навчальної дисципліни “Програмування (Rust)” складений відповідно до освітньо-професійної програми підготовки бакалавра спеціальності 171 “Електроніка”.

3. Мета та цілі курсу

Мета: сформувати у студентів фахові компетенції, теоретичні знання та практичні навички з об’єктно-орієнтованого програмування, тестування програмного коду, реалізації проектів на мові Rust.

Завдання:

- знайомство з сучасним станом, технологіями та перспективами мультипарадигмового програмування на мові Rust;
- вивчення синтаксису та семантики мови Rust;
- вивчення парадигм функційного програмування і моделі акторів;
- вивчення імперативних, процедурних і об’єктно-орієнтованих методів програмування на мові Rust;
- формування практичних навичок узагальненого програмування і метапрограмування;
- формування практичних навичок роботи з файлами і потоками;
- формування практичних навичок з оброблення виняткових ситуацій.
- формування практичних навичок з розроблення і проведення автоматизованих тестів;

У результаті вивчення навчальної дисципліни студент повинен

знати:

- сучасний стан, технології та перспективи розвитку мови Rust;
- синтаксис та семантику мови Rust;
- принципи мультипарадигмового програмування;
- особливості імперативних, процедурних і об'єктно-орієнтованих методів програмування

на мові Rust;

- типи даних і колекції даних мови Rust;
- організацію проектів з використанням пакетів, ящиків (crates) і модулів;
- права власності, посилання та запозичення даних;

вміти:

- розробляти програми відповідно до принципів мультипарадигмового програмування;
- розробляти і налагоджувати класичні алгоритми на мові Rust;
- розробляти і налагоджувати багатопотокові алгоритми на мові Rust;
- керувати проектами з використанням пакетів, ящиків (crates) і модулів;
- обробляти і генерувати виняткові ситуації у програмах;
- використовувати розумні покажчики;
- розробляти автоматизовані тести для програм;
- розробляти програми з консольним інтерфейсом;

4. Компетентності

Загальні

- Здатність застосовувати знання у практичних ситуаціях.

Спеціальні (фахові)

- Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.
- Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.
- *Здатність застосовувати принципи мультипарадигмового програмування;*
- *Здатність керувати великими програмними проектами за допомогою пакетів, ящиків і модулів.*
- *Здатність розробляти та налагоджувати одно- та багатопотокові програми.*

5. Результати навчання

- Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовувати методи, що є найбільш придатними для досягнення поставлених цілей.

6. Організація навчання курсу

Обсяг курсу	
Вид заняття	Загальна кількість годин
лекції	30

семінарські заняття / практичні / лабораторні		30			
самостійна робота		120			
Ознаки курсу					
Семестр	Спеціальність	Курс (рік навчання)		Нормативний/ вибірковий	
4	171 Електроніка	2		вибірковий	
Тематика курсу					
Тема	Форма заняття, год.	Література	Кількість годин	Вага оцінки	Термін виконання
Змістовий модуль 1. Основи Rust					
Тема 1. Вступ в Rust. Мультипарадигмове програмування. Змінні та мінливість. Скалярні і складені типи даних.	лекція	1, 4, 5, 9	2	2	Згідно розкладу
Тема 2. Керування потоком. Функції та параметри.	лекція	1, 4, 5, 9	2	2	Згідно розкладу
Тема 3. Володіння і правила володіння. Посилання та запозичення. Зрізи.	лекція	1, 4, 5, 9	2	2	Згідно розкладу
Тема 4. Структурування пов'язаних даних. Визначення та створення екземплярів Struct. Кортежі. Перерахування та зіставлення за зразком. Значення Enum.	лекція	1, 4, 5, 9	2	2	Згідно розкладу
Тема 5. Керування проектами. Пакети, ящики, модулі. Шляхи, область видимості.	лекція	1, 4, 5, 9	2	2	Згідно розкладу
Тема . 6. Загальні колекції. Вектори, стрічки символів String, словникові хеш-карти.	лекція	1, 4, 5, 9	2	2	Згідно розкладу
Тема 7. Оброблення помилок. Відновлювані і невідновлювані помилки.	лекція	1, 4, 5, 9	2	2	Згідно розкладу
Модульний контроль 1			2		
Змістовий модуль 2. Rust і мультипарадигмовість					
Тема 8. Узагальнені типи, ознаки та тривалість життя. Перевірка посилань за допомогою тривалості життя	лекція	1, 4, 5, 9	2	2	Згідно розкладу

Тема 9. Автоматизовані тести. Написання і виконання тестів. Макрокоманда assert!. Контроль виконання тестів. Організація тестів.	лекція	1, 4, 5, 9	2	2	Згідно розкладу
Тема 10. Введення/виведення даних. Аргументи командного рядка. Робота зі змінними середовища.	лекція	1, 4, 5, 9	2	2	Згідно розкладу
Тема 11. Функційне програмування. Ітератори та замкнення. Анонімні функції.	лекція	1, 4, 5, 9	2	2	Згідно розкладу
Тема 12. Розумні вказівники. Використання Vox<T>, Rc<T>, RefCell<T>, Weak<T>.	лекція	1, 4, 5, 9	2	2	Згідно розкладу
Тема 13. Багатопоточність. Потоки. Канали. Передача повідомлень між потоками. Паралелізм і синхронізація, м'ютекси.	лекція	1, 4, 5, 9	2	2	Згідно розкладу
Тема 14. Об'єктно-орієнтоване програмування. Використання об'єктів. Об'єкти з рисами.	лекція	1, 4, 5, 9	2	2	Згідно розкладу
Тема 15. Зіставлення із зразком. Синтаксис. Можливості застосування.	лекція	1, 4, 5, 9	2		Згідно розкладу
Модульний контроль 2			2		
Практичний модуль					
1. Типи даних, операції і оператори.	лаб. робота	1, 4, 5, 9	2	2	Згідно розкладу
2. Потік керування, функції.	лаб. робота	1, 4, 5, 9	2	2	Згідно розкладу
3. Володіння даними.	лаб. робота	1, 4, 5, 9	2	2	Згідно розкладу
4. Структури даних.	лаб. робота	1, 4, 5, 9	2	4	Згідно розкладу
5. Шаблони.	лаб. робота	1, 4, 5, 9	2	4	Згідно розкладу
6. Керування прекомом. Cargo, Crates.io.	лаб. робота	1, 4, 5, 8, 9	2	4	Згідно розкладу
7. Загальні колекції.	лаб. робота	1, 4, 5, 8, 9	2	4	Згідно розкладу
8. Оброблення помилок.	лаб.	1, 4, 5, 8, 9	2	4	Згідно

	робота				розкладу
9. Тестування програм	лаб. робота	1, 4, 5, 8, 9	2	4	Згідно розкладу
10. Введення/виведення даних.	лаб. робота	1, 4, 5, 8, 9	4	4	Згідно розкладу
11. Функційне програмування.	лаб. робота	1, 4, 5, 9	4	4	Згідно розкладу
12. Розумні покажчики.	лаб. робота	1, 4, 5, 8, 9	4	4	Згідно розкладу
13. Потоки і паралелізм.	лаб. робота	1, 4, 5, 9	4	4	Згідно розкладу
14. Об'єктно-орієнтоване програмування.	лаб. робота	1, 4, 5, 8, 9	4	4	Згідно розкладу
15. Шаблони					
Самостійна робота					
Тема 1. Середовища розроблення Rust програм.	самостій на робота	1, 5, 9	8	2	Впродовж семестру
Тема 2. Стандартна бібліотека.	самостій на робота	1, 5, 9	8	2	Впродовж семестру
Тема 3. Менеджер пакетів Rust.	самостій на робота	1, 5, 9	10	2	Впродовж семестру
Тема 4. Створення документації Rust.	самостій на робота	1, 5, 9	10	2	Впродовж семестру
Тема 5. Компілятор Rust.	самостій на робота	1, 5, 9	10	2	Впродовж семестру
Тема 6. Програми командного рядка.	самостій на робота	1, 4, 9	10	2	Впродовж семестру
Тема 7. Rust і WebAssembly	самостій на робота	2, 3, 8, 9	10	2	Впродовж семестру
Тема 8. Rust і вбудовані системи.	самостій на робота	3, 9, 12	10	2	Впродовж семестру
Тема 9. Rust і бібліотеки GUI	самостій на робота	1, 9, 11	10	2	Впродовж семестру
Тема 10. Rust і інтерфейси до баз даних	самостій на робота	1, 9, 10	10	2	Впродовж семестру

Тема 11. Rust і мікроконтролери.	самостійна робота	1, 9, 13	10	2	Впродовж семестру
Тема 12. Рефакторинг і масштабування Rust програм.	самостійна робота	1, 5, 9	10	2	Впродовж семестру
Контроль самостійної роботи	самостійна робота		2		Згідно розкладу
Підсумковий контроль (залік)				100	

7. Система оцінювання курсу

Загальна система оцінювання курсу			
<p><i>Поточний контроль</i> здійснюється під час проведення лекційних, лабораторних, індивідуальних занять і має на меті перевірку знань студентів з окремих тем навчальної дисципліни та рівня їх підготовленості до виконання конкретної роботи. Оцінки у національній шкалі («відмінно» - 5, «добре» - 4, «задовільно» - 3, «незадовільно» - 2), отримані студентами, виставляються у журналах обліку відвідування та успішності академічної групи.</p> <p><i>Модульний контроль (сума балів за окремих змістовий модуль)</i> проводиться (виставляється) на підставі оцінювання результатів знань студентів після вивчення матеріалу з логічно завершеної частини дисципліни – змістового модуля.</p> <p>Завданням модульного контролю є перевірка розуміння та засвоєння певного матеріалу (теми), вироблення навичок проведення розрахункових робіт, вміння вирішувати конкретні ситуативні задачі, самостійно опрацьовувати тексти, здатності осмислювати зміст даної частини дисципліни, вміння публічно чи письмово подати певний матеріал.</p> <p><i>Семестровий (підсумковий) контроль</i> проводиться у формі заліку заліку.</p> <p><i>Залік</i> – форма підсумкового контролю, що полягає в оцінюванні засвоєння здобувачем навчального матеріалу з певної дисципліни, і складається із зданих залікових змістових модулів, виконаних тестових завдань, ситуаційних робіт, опрацювання завдань робочих зошитів, тематичних рефератів, лабораторних робіт тощо, передбачених навчальною програмою.</p>			
Шкала оцінювання: національна та ECTS			
Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90-100	A	відмінно	зараховано
80-89	B	добре	
70-79	C		
60-69	D		
50-59	E		
26-49	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного

			складання
0-25	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

8. Політика курсу

Студент зобов'язаний відвідувати заняття відповідно до встановленого розкладу, не запізнюватися, мати відповідний зовнішній вигляд. У разі відсутності через хворобу надається відповідна довідка.

Пропущена лекція відпрацьовується студентом самостійно і оформляється як короткий конспект за темою заняття.

Пропущена лабораторна робота виконується студентом самостійно вдома або в комп'ютерному класі, результати оцінюються викладачем.

У випадку, коли студент приймав участь у програмі мобільності, можливе врахування отриманих оцінок в іншому навчальному закладі за умови відповідності навчальних планів.

Політика академічної поведінки і етики

Студент повинен бути толерантним і поважати думку інших.

Заперечення повинні формулюватися тільки в коректній формі.

Академічна доброчесність здобувачами вищої освіти передбачає:

- самостійне виконання навчальних завдань, завдань поточного та підсумкового контролю результатів навчання;
- посилення на джерела інформації у разі використання ідей, розробок, тверджень, відомостей;
- дотримання норм законодавства про авторське право і суміжні права;
- надання достовірної інформації про результати власної навчальної (наукової, творчої) діяльності, використанні методики досліджень і джерелі інформації.

Не допускається підказування та списування під час здачі будь-яких робіт поточного, семестрового, підсумкового контролю.

Не допускається користування телефонами та будь-якими іншими електронними засобами під час здачі будь-яких робіт поточного, семестрового, чи підсумкового контролю.

9 Рекомендована література

1. Основна

1. Steve Klabnik, Carol Nichols. The Rust Programming Language, 2nd Edition. No Starch Press: 2023. 560 p.
2. Maxwell Flitton. Rust Web Programming: A hands-on guide to developing, packaging, and deploying fully functional Rust web applications 2nd ed. Packt Publishing: 2023. 666 p.
3. Shing Lyu, Andrew Rzeznik. Practical Rust Projects: Build Serverless, AI, Machine Learning, Embedded, Game, and Web Applications 2nd ed. Apress, 2023. 410 p.
4. Ken Youens-Clark. Command-Line Rust: A Project-Based Primer for Writing Rust CLIs. O'Reilly Media, 2022. 396 p.
5. Jon Gjengset. Rust for Rustaceans: Idiomatic Programming for Experienced Developers. No Starch Press, 2021. 258 p.

Додаткова

6. McNamara, T. S. Rust in Action. Shelter Island, NY: Manning Publications, 2021. 456 p.

7. Jim Blandy, Jason Orendorff, Leonora Tindall. Programming Rust: Fast, Safe Systems Development. 2nd Edition. O'Reilly Media, 2021. 735 p.

8. Carlo Milanesi. Creative Projects for Rust Programmers Build exciting projects on domains such as web apps, WebAssembly, games, and parsing. Packt Publishing: 2020. 404 p.

Інтернет ресурси

9. Rust. Документація. Режим доступу: <https://doc.rust-lang.org/book/>

10. Rust. Інтерфейс до баз даних. Режим доступу: <https://crates.io/categories/database/>

11. Rust. Бібліотека GUI. Режим доступу: <https://blog.logrocket.com/state-of-rust-gui-libraries/>

12. Rust. Вбудовувані системи. Режим доступу: <https://www.rust-lang.org/what/embedded>

13. Rust. Мікроконтролери. Режим доступу:
<https://www.infineon.com/cms/en/product/promopages/rust/>

Викладач



Голота В.І.