



ФУНКЦІОНАЛЬНЕ ПРОГРАМУВАННЯ: ERLANG

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Другий (магістерський)
Галузь знань	12 Інформаційні технології
Спеціальність	122 Комп'ютерні науки
Освітня програма	28521
Статус дисципліни	Вибіркова
Форма навчання	очна
Рік підготовки, семестр	1 курс, 2 семестр
Обсяг дисципліни	5 кредитів ЄКТС / 150 годин: лекції – 36 год., лабораторні роботи – 36 год., самостійна робота – 78 год.
Семестровий контроль/ контрольні заходи	м.к.р., р.г.р., екзамен
Розклад занять	http://rozklad.kpi.ua
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: доцент, к.т.н., доцент Шаповалова Світлана Ігорівна, email: lanashape@gmail.com Лабораторні роботи: доцент, к.т.н., доцент Шаповалова Світлана Ігорівна, email: lanashape@gmail.com
Розміщення курсу	https://campus.kpi.ua

Програма навчальної дисципліни

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

В дисципліні вивчаються концептуальні підходи та методи функціонального програмування. Програмним засобом виконання завдань комп'ютерного практикуму є Erlang/OTP (з вільною ліцензією Erlang Public License). Інсталяцію Erlang/OTP можна отримати з ресурсу [1].

Завдяки оволодінню навичками програмування на мові Erlang, студенти можуть брати участь у проєктах з реалізації паралельних розподілених систем, які здатні обслуговувати мільйони підключень.

Метою дисципліни є формування у студентів здатності застосовувати методи, конструкції та засоби функціонального програмування для розробки високорівневих, розпаралелених застосунків, зокрема призначених для роботи в реальному часі.

Предметом вивчення дисципліни є моделі, методи, інструментальні засоби створення масштабуємих програмних Erlang-застосунків.

Програмні результати навчання

– **знання** технік, конструкцій та методів функціонального програмування, а також реалізації паралельних обчислень на мові Erlang

– **вміння:**

- застосування засобів функціонального програмування;
- послідовного (sequential) та паралельного (concurrent) програмування на мові Erlang.

– **досвід** використання одержаних знань та умінь для:

- програмування на мові Erlang, пристосованої для реалізації паралельних розподілених систем, які здатні обслуговувати мільйони підключень.

2. Пререквізити та постреквізити дисципліни

У структурно-логічній схемі навчання зазначена дисципліна розміщена на 1 курсі підготовки магістрів. Структура викладання побудована таким чином, щоб вивчення дисципліни мало теоретичне, наукове та практичне спрямування.

Вивчення дисципліни спирається на знання, отримані студентами в попередні роки навчання за освітніми програмами бакалавра та магістра в галузі 12 Інформаційні технології.

Після вивчення дисципліни студенти зможуть використати набуті знання та вміння при проектуванні, моделюванні та створенні масштабуємих програмних застосунків, призначених для розподілених обчислень на групі комп'ютерів/процесорів реальному часі.

Матеріал даної дисципліни може бути інструментальною основою для розробки програмного забезпечення та проведення обчислювальних експериментів при підготовці магістерської дисертації.

3. Зміст навчальної дисципліни

Лекційні заняття

Розділ 1. Концепція Erlang

Тема 1.1. Концепція, синтаксис і об'єкти Erlang

Тема 1.2. Представлення клонів Erlang

Тема 1.3. Послідовне програмування Erlang

Тема 1.4. Обробка та перетворення списків

Розділ 2. Функціональне програмування

Тема 2.1. Функції вищого порядку

Тема 2.2. Генератори списків

Розділ 3. Бази даних

Тема 3.1. Зберігання та обробка даних Key-Value

Тема 3.2. Реалізація баз даних

Розділ 4. Паралельне програмування

Тема 4.1. Концепція паралельного програмування

Тема 4.2. Помилки в паралельних програмах

4. Навчальні матеріали та ресурси

Базові

1. Erlang Programming Language / Download Erlang/OTP, News, Articles, Documentation, etc. URL: <http://www.erlang.org/>
2. Шаповалова С.І. Декларативне програмування: Комп'ютерний практикум : навч. посіб. для здобувачів ступеня магістра за освітньою програмою «Комп'ютерний моніторинг та геометричне моделювання процесів і систем». КПІ ім. Ігоря Сікорського, 2019. 62 с.
3. Laurent S. Introducing Erlang. O'Reilly, 2013. 201 p.
4. Heber F. Learn You Some Erlang for Great Good! No Starch Press, 2013. 624 p.
5. Cesarini F., Thompson S. Erlang Programming. O'Reilly, 2009. 498 p.

Додаткові

6. Logan M., Merritt E., Carlsson R. Erlang and OTP in Action. Manning, 2011. 431 p.
7. Cesarini F., Vinoski S. Designing for Scalability with Erlang/OTP: Implement Robust, Fault-Tolerant Systems. O'Reilly, 2016. 482 p.
8. Loder W. Erlang and Elixir for Imperative Programmers. Berkeley, CA : Apress : Imprint: Apress. 2016. 256 p.

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

№ з/п	Назва теми лекції та перелік основних питань	Кільк. ауд.год.
Розділ 1. Концепція Erlang		
Тема 1.1. Концепція, синтаксис і об'єкти Erlang		
1	<i>Лекція 1 Тема: Концепція, синтаксис і об'єкти Erlang</i> Концепція Erlang. Базові типи Erlang та операції з ними. Операція порівняння	2
2	<i>Лекція 2 Тема: Декларативне подання Erlang</i> Співставлення зі зразком. Змінна. Модульне подання програми. Вбудовані функції. Середовище Erlang	2
Тема 1.2. Представлення клозів Erlang		
3	<i>Лекція 3 Тема: Конструкції клозів Erlang</i> Охоронні вирази. Представлення клозів Erlang. Умовні вирази	2
4	<i>Лекція 4 Тема: Рекурсія</i> Види рекурсії. Конструкції та приклади прямої та хвостової рекурсії	2
Тема 1.3. Послідовне програмування Erlang		
5	<i>Лекція 5. Послідовне програмування Erlang</i> Конструкції та засоби послідовного програмування. Обробка помилок часу виключення	2
Тема 1.4. Обробка та перетворення списків		
6	<i>Лекція 6. Списки</i> Визначення та подання списків. Побудова списків на основі конструкції cons.	2
7	<i>Лекція 7. Обробка списків</i> Операції зі списками. Конкатенація двох списків. Перетворення частин списку.	2
8	<i>Лекція 8. Обробка списку списків</i> Базові операції зі списками списків. Приклади перетворення матриць на основі cons.	2
Розділ 2. Функціональне програмування		
Тема 2.1. Функції вищого порядку		
9	<i>Лекція 9 Тема: Функції вищого порядку</i> Конструкції функціонального програмування. Визначення функції як окремого типу даних fun.	2
10	<i>Лекція 10 Тема: Базові операції на списках на основі функцій вищого порядку</i> Шаблони використання функцій вищого порядку. Приклади реалізації задач.	2
11	<i>Лекція 11 Тема: Згортка</i> Визначення та різновиди згортки. Складні формати акумулятора згортки.	2
Тема 2.2. Генератори списків		
12	<i>Лекція 12 Тема: Генератори списків</i> Визначення та представлення генераторів списків. Конструювання форматів списків, які генеруються.	2
13	<i>Лекція 13 Тема: Обробка списків на основі генераторів</i> Конструкції базових операцій зі списками. Особливості реалізації фільтрації.	2
Розділ 3. Бази даних		
Тема 3.1. Зберігання та обробка даних Key-Value		
14	<i>ЛЕКЦІЯ 14 Тема: Зберігання та обробка даних Key-Value</i>	2

	Представлення колекцій термів. Записи records. Карти maps.	
Тема 3.2. Реалізація баз даних		
15	<i>ЛЕКЦІЯ 15 Тема: ETS таблиці</i> Засоби зберігання та швидкого отримання даних. ETS таблиці.	2
16	<i>ЛЕКЦІЯ 16 Тема: Mnesia</i> Концепція Mnesia. Побудова запитів до бази даних. Політика транзакцій. Функції ведення транзакцій. Збереження складних типів даних в Mnesia. Налаштування таблиць.	2
Розділ 4. Паралельне програмування		
Тема 4.1. Концепція паралельного програмування		
17	<i>Лекція 17 Тема: Концепція паралельного програмування</i> Паралельні примітиви. Базові операції роботи з процесами. Вбудовані функції управління зареєстрованими процесами	2
Тема 4.2. Помилки в паралельних програмах		
18	<i>ЛЕКЦІЯ 18 Тема: Помилки в паралельних програмах</i> Концепції обробки помилок в паралельних програмах. Примітиви для обробки помилок.	2

Лабораторні роботи

№ з/п	Назва	Кільк. ауд.год
1	Задача класифікації	4
2	Розгалужені алгоритми	6
3	Конструювання списків	4
4	Обробка елементів списків	4
5	Виокремлення частин списків	4
6	Обробка списку списків	4
7	Використання генераторів списків	4
8	Фільтрація списку	6

Розрахунково-графічна робота (РГР)

Метою РГР є навчитись використовувати можливості паралельного програмування для оптимізації часу виконання програми.

Завдання: задано квадратну матрицю розмірності N . Написати дві реалізації вирішення індивідуального завдання:

- 1) на основі виконання підзадач в окремих потоках - за N ітерацій;
- 2) на основі послідовного програмування в одному потоці - за $2N$ ітерацій.

Провести обчислювальні експерименти з перетворення згенерованих матриць, розмір N яких знаходиться в діапазоні від 10 до 10^5 , для визначення реалізації, оптимальної за часом виконання.

6. Самостійна робота студента

Самостійна робота студента (78 годин) передбачає підготовку до аудиторних занять та контрольних заходів, проведення досліджень та створення програмного забезпечення для виконання завдань РГР та лабораторних робіт.

Розподіл годин самостійної роботи: підготовка до екзамену – 30 годин; підготовка до модульної контрольної роботи – 6 годин; підготовка до розрахунково-графічної роботи – 10 годин; підготовка до лабораторної роботи – 4 години.

7. Політика навчальної дисципліни (освітнього компонента)

- Відвідування лекцій, а також відсутність на них, не оцінюється. Відвідування лабораторних занять є обов'язковою складовою вивчення матеріалу;
- При захисті робіт комп'ютерного практикуму студент має продемонструвати розроблений програмний код та результати його виконання на тестах, як заздалегідь підготованих, так і запропонованих викладачем. У випадку дистанційної форми навчання захист відбувається на відповідній конференції шляхом демонстрації екрана.
- Політика та принципи академічної доброчесності визначені у розділі 3 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.
- Норми етичної поведінки Норми етичної поведінки студентів і працівників визначені у розділі 2 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.

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

1. Рейтинг студента з кредитного модуля складається з балів, що він отримує за:

- виконання та захисту 8 лабораторних робіт;
- виконання 1 модульної контрольної роботи;
- виконання 1 розрахунково-графічної роботи;
- відповідь на екзамені.

2. Критерії нарахування балів:

2.1. Кожна з лабораторних робіт оцінюється із 5 балів і зараховується за наявності коректних рішень. Бали нараховуються таким чином:

- 1) оптимальність алгоритму (3 балів);
- 2) оптимальність використаних структур подання інформації (2 балів).

Штрафні бали призначаються за:

- 1) несвоєчасне (пізніше ніж на тиждень) виконання завдання комп'ютерного практикуму – 1 бали;
- 2) ненадану або невірну відповідь на запитання – 1 бали.

2.2 Розрахунково-графічна робота оцінюється із 15 балів таким чином:

- 1) оптимальність алгоритму реалізації послідовного програмування (5 балів);
- 2) оптимальність алгоритму реалізації паралельного програмування (5 балів);
- 3) проведення обчислювальних експериментів і представлення їх результатів (5 балів).

2.3 Модульна контрольна робота оцінюється із 5 балів таким чином:

- 4) коректність отриманих рішень за всіма заданими підходами (3 балів);
- 5) оптимальність алгоритмів (2 балів).

Штрафні бали призначаються за:

- 1) Відсутність реалізації одного з заданих методів – 5 балів.

2.3. Відповідь на екзамені оцінюється із 40 балів. Екзаменаційний білет складається з двох запитань, а також практичного завдання.

Кожна відповідь на запитання оцінюється з 10 балів за такими критеріями::

- «відмінно», повна відповідь (не менше 90% потрібної інформації) – 10-9 балів;
- «добре», достатньо повна відповідь (не менше 75% потрібної інформації, або незначні неточності) – 8-7 балів;
- «задовільно», неповна відповідь (не менше 60% потрібної інформації та деякі помилки) – 6 балів;
- «незадовільно», відповідь не відповідає умовам на «задовільно» – 0 балів.

Нарахування балів за виконання практичного завдання оцінюється із 20 балів таким чином:

1) коректність отриманих рішень за всіма заданими підходами (15 балів);

2) оптимальність алгоритмів (5 балів).

4. До екзамену допускаються студенти, які отримали в семестрі не менше 40 балів.

5. Сума стартових балів і балів, отриманих на екзамені, визначає оцінку згідно з таблицею:

<i>Кількість балів</i>	<i>Оцінка</i>
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

9. Додаткова інформація з дисципліни (освітнього компонента)

В дисципліні представлено базовий курс функціонального програмування в Erlang. Стартові бали за окремі теми розділу 1 та 2 можуть бути зараховані за наявності сертифікату як курсу для початківців (наприклад, курс “Erlang Programming for Beginners”, представлений Udemy), так і фахових розробників.

Робочу програму навчальної дисципліни (силабус):

Складено доцентом, к.т.н., доц. Шаповаловою Світланою Ігорівною

Ухвалено кафедрою ЦТЕ (протокол № 20 від 30.05.24)

Погоджено Методичною радою НН ІАТЕ КПІ ім. Ігоря Сікорського (протокол № 9 від 31.05.24)

Погоджено науково-методичною комісією КПІ ім. Ігоря Сікорського зі спеціальності 122
(протокол № 11 від 28.06.24)