



DevOps

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	122 Комп'ютерні науки
Освітня програма	Цифрові технології в енергетиці
Статус дисципліни	Вибіркова
Форма навчання	очна(денна)/дистанційна/змішана
Рік підготовки, семестр	3 курс, осінній семестр
Обсяг дисципліни	4 кредити ЄКТС (120 годин): лекції – 36 год., лабораторні роботи – 18 год., самостійна робота – 66 год.
Семестровий контроль/ контрольні заходи	модульна контрольна робота, залік
Розклад занять	rozklad.kpi.ua
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: доцент, к.т.н. Демчишин Анатолій Анатолійович, ntuu.kpi@ukr.net Лабораторні: доцент, к.т.н. Демчишин Анатолій Анатолійович, ntuu.kpi@ukr.net
Розміщення курсу	youtube, https://campus.kpi.ua

Програма навчальної дисципліни

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

Актуальність кредитного модуля спирається на невід'ємність візуалізації об'єктів та процесів як складового елемента будь-якої інформаційної системи.

Кредитний модуль мотивує студентів до практичного використання умінь, набутих впродовж попереднього навчання. Зміст модуля побудований таким чином, щоб його вивчення мало значення, спрямоване на розвиток інженерного мислення.

Метою кредитного модуля є формування у студентів таких загальних і фахових компетентностей:

- здатність до абстрактного мислення, аналізу та синтезу (ЗК1);
- здатність застосовувати знання у практичних ситуаціях (ЗК2);
- знати та розуміти предметну область та мати розуміння професійної діяльності (ЗК3);
- здатність генерувати нові ідеї (креативність) (ЗК8);
- здатність працювати в команді (ЗК9);
- здатність діяти на основі етичних міркувань (ЗК13);
- здатність реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт-сервер, включаючи бази даних, знань і сховища даних, виконувати розподілену обробку великих наборів даних на кластерах стандартних серверів для забезпечення обчислювальних потреб користувачів, у тому числі на хмарних сервісах (ФК9).

Програмні результати навчання кредитного модулю (ПРН)

- знати життєвий цикл програмного забезпечення та сучасні методології його розроблення;
- знати парадигми сучасних мов програмування;
- знати принципи побудови, склад, структуру та функції сучасних UNIX-подібних операційних систем;

- володіти принципами та протоколами взаємодії основних мережних служб мережі Інтернет; застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній галузі комп'ютерних наук (ПРН1);
- використовувати довідкову літературу, технічну документацію;
- встановлювати, налаштовувати та обслуговувати системне та прикладне програмне забезпечення в хмарних середовищах;
- правильно застосовувати основні інструментальні програмні засоби DevOps-інженера.

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

Вивчення дисципліни спирається на знання, отримані за програмою попередніх років навчання за спеціальністю 122 «Комп'ютерні науки».

Пререквізити дисципліни. В структурно-логічній схемі навчання дисципліна “DevOps” розташована в п'ятому семестрі, тобто тоді, коли студенти вже прослухали курси "Алгоритмізація та програмування", "Комп'ютерна схемотехніка та архітектура", засвоїли принципи побудови алгоритмів, ознайомилися з архітектурою сучасних комп'ютерів та набули певного досвіду в галузі розроблення програмного забезпечення.

Постреквізити дисципліни. Матеріал цього кредитного модуля може бути використаний під час вивчення таких кредитних модулів, як: "Системи баз даних", "Технології паралельних обчислень в енергетичних комплексах", що подаються в наступних семестрах.

3. Зміст навчальної дисципліни

Розділ 1. Загальні відомості про методологію DevOps

Тема 1.1. DevOps – методологія інтеграції процесів розроблення програмного забезпечення та процесів операцій

Тема 1.2. Гіпервізор другого типу Oracle VM VirtualBox

Розділ 2. Інфраструктура ОС Linux

Тема 2.1. Команди Linux

Тема 2.2. Команди адміністрування Linux

Тема 2.3. Утиліта автоматичної збірки програм make

Розділ 3. Розподілена система керування версіями файлів GIT

Тема 3.1. Команди для роботи із станами файлів

Тема 3.2. Команди роботи з гілками

Тема 3.3. CI/CD платформа GitHub Actions

Тема 3.4. Мова розмітки YAML

Розділ 4. Збірка програмних систем та створення дистрибутивів

Тема 4.1. Система збірки проєктів GNU Autotools

Тема 4.2. Створення tarball дистрибутиву вихідних кодів програмної системи

Тема 4.3. Створення дистрибутиву програмної системи у вигляді Debian пакету

Тема 4.4. Інженерія програмного застосунку HTTP серверу

Розділ 5. Система контейнеризації Docker

Тема 5.1. Створення образів та запуск контейнерів

Тема 5.2. Багатоетапна побудова образів

Розділ 6. Горизонтальне масштабування

Тема 6.1. Налаштування зворотного проксі серверу Nginx в якості балансувальника навантаження

Тема 6.2. Коректне завершення дочірніх процесів HTTP серверу

Тема 6.3. Безперервне розгортання

4. Навчальні матеріали та ресурси

Основна література

1. Kim G., Humble J., Debois P., Willis J., Forsgren N. The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations. 2nd ed. Portland, OR: IT Revolution Press. 2021. 480p.
2. Kernighan, B. W., and R. Pike. The UNIX Programming Environment. Englewood Cliffs, NJ: Prentice-Hall. 1984. 357p.
3. Nemeth E., Snyder G., Hein T., Whaley B., Mackin D. UNIX and Linux System Administration Handbook. 5th ed. Boston, MA: Addison-Wesley Professional. 2017. 1232p.
4. Wittig A., Wittig M. Amazon Web Services in Action: An in-depth guide to AWS. 3rd ed. Shelter Island, NY: Manning Publications. 2023. 552p.
5. Chacon S., Straub B. Pro Git. 2nd ed. New York, NY: Apress. 2014. 440p.

Додаткова література

1. Mouat A. Using Docker: Developing and Deploying Software with Containers. Sebastopol, CA: O'Reilly Media. 2016. 351p.
2. Domingus D., Arundel J. Cloud Native DevOps with Kubernetes: Building, Deploying, and Scaling Modern Applications in the Cloud. 2nd ed. Sebastopol, CA: O'Reilly Media. 2022. 344p.
3. Morris K. Infrastructure as Code: Dynamic Systems for the Cloud Age. 2nd ed. Sebastopol, CA: O'Reilly Media. 2021. 430p.
4. Freeman J., Keating J. Mastering Ansible: Automate configuration management and overcome deployment challenges with Ansible. 4th ed. Birmingham: Packt Publishing. 2021. 540p.
5. Tanenbaum A. S., Bos. H. Modern Operating Systems. 5th ed. Upper Saddle River, NJ: Prentice-Hall PTR. 2023. 1156p.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

№ з/п	Назва теми лекції та перелік основних питань	Кіль. ауд. год.
Розділ 1. Загальні відомості про методологію DevOps		
1	<i>ЛЕКЦІЯ 1. Тема 1.1. DevOps – методологія інтеграції процесів розроблення програмного забезпечення та процесів операцій</i> Ретроспектива методологій розроблення програмного забезпечення та процесів операцій. Порівняльний аналіз ролей членів команд розробників та команд операцій	2
2	<i>ЛЕКЦІЯ 2. Тема 1.2. Гіпервізор другого типу Oracle VM VirtualBox</i> Огляд гіпервізорів першого та другого типу. Налаштування конфігурації віртуальної машини: кількість ядер, тип мережевої конфігурації (NAT, Bridged Adapter, Host only Adapter), каталоги спільного використання	2
Розділ 2. Інфраструктура ОС Linux		
3	<i>ЛЕКЦІЯ 3. Тема 2.1 Команди Linux</i> Ієрархія каталогів ОС Linux, команди операцій над файлами, команди текстового редактору vim, команди операцій з пакетами, команди пошуку файлів	2
4	<i>ЛЕКЦІЯ 4. Тема 2.2. Команди адміністрування Linux</i> Команди менеджменту користувачів та груп системи, інтерфейс командного	2

	рядка <code>bash</code> , команди навколишнього середовища <code>bash</code> , команди конфігурації мережевих інтерфейсів ядра, класи приватних мереж	
5	<i>ЛЕКЦІЯ 5. Тема 2.3. Утиліта автоматичної збірки програм <code>make</code></i> Компіляція та лінковка програмних застосунків на мові програмування C++, ключі налаштування рівнів оптимізації та попереджень компілятора <code>g++</code> , предметно-орієнтована мова програмування утиліти <code>make</code>	2
Розділ 3. Розподілена система керування версіями файлів GIT		
6	<i>ЛЕКЦІЯ 6. Тема 3.1. Команди для роботи із станами файлів</i> Порівняння централізованих та децентралізованих систем контролю версій, стани файлів у системі контролю версій <code>git</code> , налаштування зовнішніх утиліт порівняння та злиття файлів, рівні відкату версій, команди роботи зі стешем, команди роботи з історією, синтаксис файлу <code>.gitignore</code>	2
7	<i>ЛЕКЦІЯ 7. Тема 3.2. Команди роботи з гілками</i> Сценарії повсякденної роботи з гілками учасника команди <code>devops</code>	2
8	<i>ЛЕКЦІЯ 8. Тема 3.3. CI/CD платформа <code>GitHub Actions</code></i> Автоматизація процесів безперервної інтеграції та безперервного розгортання із застосуванням серверів <code>Microsoft Azure</code>	2
9	<i>ЛЕКЦІЯ 9. Тема 3.4. Мова розмітки <code>YAML</code></i> Лексеми мови розмітки <code>YAML</code> послідовностей виконуваних дій <code>GitHub Actions</code>	2
Розділ 4. Збірка програмних систем та створення дистрибутивів		
10	<i>ЛЕКЦІЯ 10. Тема 4.1. Система збірки проєктів <code>GNU Autotools</code></i> Компоненти пакету <code>Autotools</code> , предметно-орієнтована мова програмування утиліти <code>autoconf</code> , предметно-орієнтована мова програмування утиліти <code>automake</code> , стандартні цілі <code>automake</code> . Приклад файлів конфігурації збірки проєктів <code>configure.ac</code> і <code>Makefile.am</code>	2
11	<i>ЛЕКЦІЯ 11. Тема 4.2. Створення <code>tarball</code> дистрибутиву вихідних кодів програмної системи</i> Інженерія <code>tarball</code> дистрибутиву вихідних кодів програмної системи, налаштування послідовності виконуваних дій <code>GitHub Actions</code> для безперервної інтеграції та автоматизованого тестування	2
12	<i>ЛЕКЦІЯ 12. Тема 4.3. Створення дистрибутиву програмної системи у вигляді <code>Debian</code> пакету.</i> Обов'язкові та додаткові поля опису <code>Debian</code> пакету. Команди менеджменту <code>Debian</code> пакетую Приклад створення <code>Debian</code> пакету	2
13	<i>ЛЕКЦІЯ 13. Тема 4.4. Інженерія програмного застосунку <code>HTTP</code> серверу</i> Інструмент <code>curl</code> передачі даних з використанням <code>URL</code> , методи протоколу <code>HTTP1.1</code> , заголовки протоколу <code>HTTP1.1</code> , послідовність команд встановлення з'єднання з боку сервера (системні виклики: <code>socket</code> , <code>bind</code> , <code>listen</code> , <code>accept</code>), приклад мульти-клієнтського серверу протоколу <code>HTTP1.1</code> на мові програмування C++, системний виклик <code>fork</code>	2
Розділ 5. Система контейнеризації <code>Docker</code>		
14	<i>ЛЕКЦІЯ 14. Тема 5.1. Створення образів та запуск контейнерів</i> Встановлення демону <code>Docker</code> . Предметно-орієнтована мова програмування опису <code>Docker</code> образів. Запуск та приєднання до контейнеру, виконання команд у контейнері. Хмарний сервіс зберігання та обміну <code>Docker</code> -образів <code>Docker Hub</code>	2
15	<i>ЛЕКЦІЯ 15. Тема 5.2. Багатостадійна побудова <code>Docker</code>-образів</i>	2

	Створення оптимізованих образів шляхом розділення середовищ для збірки та виконання, побудова образів для різних архітектур процесорів, налаштування послідовності виконуваних дій GitHub Actions для створення Docker-маніфестів	
Розділ 6. Горизонтальне масштабування		
16	<i>ЛЕКЦІЯ 16. Тема 6.1. Налаштування зворотного проксі серверу Nginx в якості балансувальника навантаження</i> Референсна модель опису взаємодії мережевих систем OSI. Варіанти використання віртуальних приватних мереж. Варіанти використання проксі серверів. Варіанти використання зворотних проксі серверів, встановлення та конфігурація серверу Nginx для балансування навантаження та термінації шифрування	2
17	<i>ЛЕКЦІЯ 17. Тема 6.2. Коректне завершення дочірніх процесів HTTP серверу</i> Стани процесів в ОС Linux, зомбі дочірні процеси, сигнали ОС Linux, приклад обробки сигналів SIGCHLD, SIGINT	2
18	<i>ЛЕКЦІЯ 18. Тема 6.2. Безперервне розгортання</i> Надсилання сигналів до контейнерів. Очікування завершення контейнерів. Оновлення контейнерів до найновішої версії. Моніторинг статистики використання ресурсів контейнером. Приклад bash-скрипту безперервного розгортання контейнеру	2

Лабораторні роботи

N	Назва лабораторної роботи	Кільк. ауд. год.
Розділ 2. Інфраструктура ОС Linux		
1	Створення bash-скриптів та налаштування прав доступу	4
Розділ 3. Розподілена система керування версіями файлів GIT		
2	Розробка застосунку з використанням сценарію повсякденної роботи з гілками GIT	4
Розділ 4. Збірка програмних систем та створення дистрибутивів		
3	Безперервна підготовка tarball дистрибутиву вихідних кодів програмної системи та Debian пакету	4
4	Розробка HTTP серверу на основі архітектурного стилю REST	4
Розділ 5. Система контейнеризації Docker		
5	Створення Docker-образу з HTTP сервером для різних архітектур процесорів	4

Контрольні роботи

В кінці семестру проводять одну модульну контрольну роботу. Для її проведення виділяють 2 навчальні години за рахунок часу, передбаченого для виконання лабораторних робіт.

Метою модульної контрольної роботи є перевірка засвоєння студентами теоретичних відомостей з налаштування зворотного проксі серверу Nginx в якості балансувальника навантаження та написання bash-скрипту безперервного розгортання, а також перевірка знань та практичних навичок студентів, необхідних DevOps-інженеру в повсякденній діяльності.

6. Самостійна робота студента

Самостійна робота студента (66 годин) передбачає підготовку до аудиторних занять та контрольних заходів, проведення розрахунків та підготовка вхідних даних до роботи.

На підготовку до лекції виділяється 2 години СРС; на підготовку до лабораторної роботи – 4 години; на підготовку до заліку – 6 годин СРС; на підготовку до модульної контрольної – 4 години.

Політика та контроль

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 лабораторних робіт 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. Поточний контроль (виконання робіт в семестрі) оцінюється у 100 балів. Розподіл балів наведено в таблиці:

Роботи	Максимальна кількість балів за виконання однієї роботи	Σ
Лабораторні роботи 1 - 5	8	$\Sigma_{\text{лр}} = 40$
Модульна контрольна робота	10	$\Sigma_{\text{мкр}} = 10$
Відсутність заборгованості за всіма роботами зазначеними вище	x2	$(\Sigma_{\text{лр}} + \Sigma_{\text{мкр}}) \times 2$
		100

Лабораторні завдання оцінюється наступним чином:

- повне, безпомилкове розв'язання завдання – 7-8 балів;
- повне, розв'язання завдання із несуттєвими невідповідностями – 4-6 балів;
- завдання виконане з суттєвими недоліками – 1-3 балів;
- завдання не виконано – 0 балів.

Завдання МКР оцінюється наступним чином:

- повне, безпомилкове розв'язання завдання – 9-10 балів;
- повне, розв'язання завдання із несуттєвими невідповідностями – 5-8 балів;
- завдання виконане з суттєвими недоліками – 1-4 балів;
- завдання не виконано – 0 балів.

2. Календарний контроль проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу:

Критерій	Перший календарний контроль	Другий календарний контроль
Термін календарного контролю	Тиждень 7-8	Тиждень 14-15
Умови отримання позитивної оцінки	відсутність заборгованостей з лабораторних робіт 1-2	відсутність заборгованостей з лабораторних робіт 1-5

3. Умови допуску до заліку: відсутність заборгованостей з лабораторних робіт 1-5 та МКР. Студенти, які виконали умови допуску до заліку та мають рейтингову оцінку 60 і більше балів, отримують відповідну до набраного рейтингу оцінку без додаткових випробувань.

Зі студентами, які виконали умови допуску до заліку та мають рейтингову оцінку менше 60 балів, а також зі студентами, хто бажає підвищити свою рейтингову оцінку, на останньому за розкладом занятті з дисципліни в семестрі проводиться семестровий контроль у вигляді заліку.

4. Відповідь на заліку оцінюється у 100 балів. Залікова робота складається з двох практичних завдань із ваговим балом – 50 балів.

Практичне завдання оцінюється таким чином:

- повне, безпомилкове розв'язування завдання – 45-50 балів;
- повне, розв'язування завдання із несуттєвими невідповідностями – 38-44 бали;
- завдання виконане з певними недоліками – 30-37 балів;
- завдання не виконано – 0 балів.

5. Рейтингова оцінка за освітній компонент за бажанням студента визначається одним з таких способів:

- 1) кількість балів, отриманих за поточний контроль, або
- 2) результат виконання залікової контрольної роботи (тоді не враховуються бали, отримані в семестрі).

Таблиця переведення рейтингових балів у оцінку за університетською шкалою:

Кількість балів	Оцінка
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

Робочу програму навчальної дисципліни (силабус):

Складено доцент, к.т.н., Демчишин Анатолій Анатолійович

Ухвалено кафедрою цифрових технологій в енергетиці (протокол № 21 від 30.05.2024)

Погоджено Методичною комісією НН ІАТЕ КПІ ім. Ігоря Сікорського (протокол № 9 від 31.05.2024)