



Технології DevOps

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці
Статус дисципліни	Вибіркова
Форма навчання	Очна (денна)
Рік підготовки, семестр	3 курс, осінній семестр
Обсяг дисципліни	4 кредитів ЄКТС/120 год (36 год. лекції, 18 год. практичні заняття, 66 год. самостійна робота)
Семестровий контроль/ контрольні заходи	Залік
Розклад занять	Згідно розкладу на весняний семестр поточного навчального року (http://roz.kpi.ua/)
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: старший викладач, Колумбет В.П., kvplinux@gmail.com , тел. 093-405-35-39 Практичні: старший викладач, Колумбет В.П., kvplinux@gmail.com , тел. 093-405-35-39
Розміщення курсу	Кампус

Програма навчальної дисципліни

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

В цій дисципліні детально вивчається концепції, принципи та методи інформаційних технологій. Етапи розробки програмного забезпечення. Міжнародні стандарти програмної інженерії. Моделі та процеси, які впроваджуються в розробленні програмного забезпечення. Вимоги до інформаційної системи. Документування розроблених вимог та моделей для створення інформаційної системи. Управління змінами до вимог протягом циклу розробки програмного забезпечення.

Метою дисципліни є опанування студентами теоретичних та практичних знань і навичок систем тестування, інструментальних засобів розробки програмних систем, розгортання та функціонування програмного забезпечення, розробляти, аналізувати та застосовувати ефективні алгоритми для розв'язання професійних завдань в області інформаційних технологій, та узгодження розробки й постачання програмного забезпечення із його використанням, сформувати цілісне уявлення про сучасні технології розробки додатків з мікросервісною архітектурою, їх розгортання, масштабування та тестування.

Предмет дисципліни - вивчення принципів побудови, архітектури, основних функцій, режимів роботи і елементи та комплекси практик DevOps, розроблення адаптивних алгоритмів розв'язування задач і системного адміністрування.

Завдання. В результаті вивчення дисципліни у студентів повинні сформуватися наступні компетентності:

Вивчення дисципліни «Технології DevOps» сприяє формуванню у студентів фахової компетентності (ФК) за освітньою програмою:

ФК01 - Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.

ФК05 - Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів ЖИТТЕВОГО ЦИКЛУ.

ФК08 - Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.

ФК10 - Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.

Вивчення дисципліни «Технології DevOps» сприяє формуванню у студентів наступних програмних результатів навчання (ПРН) за освітньою програмою:

ПРН01 - Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.

ПРН03 - Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.

ПРН05 - Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.

ПРН07 - Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

ПРН09 - Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.

ПРН14 - Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.

ПРН18 - Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.

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

Пререквізити дисципліни. Знання, отримані при вивченні дисциплін: “Компоненти програмної інженерії”, “Алгоритми та структури даних” та “Операційні системи”.

Постреквізити дисципліни. Отримані знання при вивченні дисципліни «Архітектура системного програмного забезпечення» формує базові знання для вивчення наступних дисциплін: “Теорія ймовірностей та математична статистика”, “Алгоритми та структури даних”, “Бази даних”, “Моделювання та аналіз програмного забезпечення”, “Основи розробки трансляторів”, “Системи штучного інтелекту”, “Організація баз даних та знань”, “Криптографія та шифрування”, “Системи керування версіями”. Також викладений матеріал може бути використаний при вивченні дисциплін “Математичне моделювання систем і процесів”, “Дослідження операцій”, які викладаються в наступних семестрах. Компетенції, отримані студентами в процесі вивчення цієї дисципліни, використовуються ними при виконанні дипломної роботи.

3. Зміст навчальної дисципліни

Тема 1. Розуміння практик DevOps та основи Linux.

Тема 2. Мережеві технології та DevOps орієнтовані сервіси платформи AWS.

Тема 3. Інструменти розгортання та Docker контейнеризація.

Тема 4. Розгортання сервісів Amazon ECS. Вступ до системи контролю версій Git.

Тема 5. Безперервна інтеграція та безперервне постачання. Інтеграція з AWS.

Тема 6. Моніторинг та обробка даних.

4. Навчальні матеріали та ресурси

Основна література

1. Шевчук І. М. DevOps та хмарні сервіси: AWS, Azure, GCP / І. М. Шевчук. – Харків : Фоліо, 2024. – 368 с. ISBN 978-966-03-9789-3.
2. Інженерія програмного забезпечення: Посібник для студентів вищих навчальних закладів / І. Л. Бородкіна, Г. О. Бородкін ; М-во освіти і науки України, Національний університет біоресурсів та природокористування України. – Київ: , 2018. – с.
3. Волошин І. В. DevOps-технології: основи та практика / І. В. Волошин. – Київ : Довіра, 2019. – 304 с. ISBN 978-966-507-823-5.
4. DevSecOps: [Електронний ресурс]. Synopsys – 2021. – Режим доступу до ресурсу: <https://www.synopsys.com/glossary/what-is-devsecops.html>
5. What is DevOps?: [Електронний ресурс]. AWS – 2020. – Режим доступу до ресурсу: <https://aws.amazon.com/devops/what-is-devops>.
6. Коваленко О. П. Інфраструктура як код: Terraform, Ansible, Puppet / О. П. Коваленко. – Львів: Видавництво Львівської політехніки, 2020. – 288 с. ISBN 978-966-936-845-3
7. Смирнова Т. В. CI/CD: неперервна інтеграція та доставка / Т. В. Смирнова. – Харків: Фоліо, 2021. – 320 с. ISBN 978-966-03-9123-4.
8. Управління версіями програмних засобів проекту: навчальний посібник для здобувачів ступеня бакалавра за освітньою програмою «Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці» спеціальності 121 Інженерія програмного забезпечення / КПІ ім. Ігоря Сікорського ; уклад.: В. О. Кузьмініх, О. В. Коваль, Р. А. Тараненко. – Київ : КПІ ім. Ігоря Сікорського, 2023. – 114 с. – Режим доступу до ресурсу <https://ela.kpi.ua/handle/123456789/56605>)
9. Іванчук С. О. Kubernetes у DevOps-процесах / С. О. Іванчук. – Київ: Навчальна книга, 2020. – 272 с. ISBN 978-966-639-302-6.
10. Петренко М. В. Моніторинг та логування у DevOps / М. В. Петренко. – Київ: BHV, 2022. – 256 с. ISBN 978-617-7659-67-4.
11. Семененко В. І. Архітектура microservices у DevOps / В. І. Семененко. – Київ: MicroFormat, 2022. – 312 с. ISBN 978-617-7398-15-1.

Додаткова література

12. Гнатюк О. М., Савчук Т. В. GitOps: інструменти та практики / О. М. Гнатюк, Т. В. Савчук. – Львів : Видавництво Львівської політехніки, 2023. – 248 с. ISBN 978-966-936-923-8.
13. Поліщук Г. Б. Observability: Prometheus і Grafana / Г. Б. Поліщук. – Київ : TechBooks, 2023. – 232 с. ISBN 978-966-917-704-1.
14. Миронова А. І. DevSecOps: безпека у DevOps / А. І. Миронова. – Харків: КМ-Книги, 2021. – 304 с. ISBN 978-617-7409-88-0
15. What is DevOps?: [Електронний ресурс]. AWS – 2020. – Режим доступу до ресурсу: <https://aws.amazon.com/devops/what-is-devops>
16. Бондар О. П. Контейнеризація та оркестрація: Docker, Kubernetes, Helm / О. П. Бондар. – Київ: Довіра, 2018. – 336 с. ISBN 978-966-507-700-9.
17. J.Mulder. Enterprise DevOps for Architects. Packt Publishing. BIRMINGHAM—MUMBAI, 2021. 178 с.
18. Кравченко Л. І. Автоматизація тестування в CI/CD / Л. І. Кравченко. – Львів : Грані-Т, 2022. – 280 с. ISBN 978-617-649-301-6.
19. Федоренко Н. В. Serverless та DevOps: інтеграція AWS Lambda / Н. В. Федоренко. – Харків: Княжа, 2025. – 256 с. ISBN 978-966-9923-14-7.
20. Кузнєцова Т. О. Ansible в практиці DevOps / Т. О. Кузнєцова. – Харків: Фоліо, 2018. – 264 с. ISBN 978-966-03-7458-7.
21. Степаненко Р. В. DevOps 4.0: автоматизація AI/ML pipeline / Р. В. Степаненко. – Київ: Літопис, 2024. – 296 с. ISBN 978-966-917-820-8.

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

Тема 1. Розуміння практик DevOps та основи Linux.

Лекція 1. Вступ до курсу DevOps.

Що таке DevOps. Огляд курсу, культура та принципи. Розгляд цінностей CALMS (Culture, Automation, Lean, Measurement, Sharing) та їхнього впливу на організацію розробки й експлуатації.

Лекція 2. Адміністрування Linux.

Інсталяція ОС Linux. Підсистема довідки Базові команди. Робота з файлами та каталогами. Текстовий редактор vim. Керування користувачами та групами. Встановлення програмного забезпечення.

Лекція 3. Управління процесами.

Утиліти ps, top, htop, atop. Журналювання. Резервне копіювання. Налаштування параметрів мережі. Програмування на bash.

Тема 2. Мережеві технології та DevOps орієнтовані сервіси платформи AWS.

Лекція 4. Основи мережевих технологій.

Моделі OSI та TCP/IP Огляд технологій локальних мереж. Адресація канального рівня. Введення у маршрутизацію: IP-адресація. Протокол IP. Протокол ICMP. Технологія NAT.

Лекція 5. Протоколи транспортного рівня: UDP та TCP.

Протоколи прикладного рівня: протокол DNS, налаштування bind, unbound, протокол DHCP, налаштування, протокол HTTP, налаштування apache2, nginx.

Лекція 6. Огляд хмарних провайдерів. Введення в AWS.

Платформа AWS, інструментарій з розробки, розгортання, міграцію даних, послуги з безпеки, мобільні сервіси. Сервіси AWS DevOps: CloudFormation, Cloud Development Kit, CodePipeline, CodeCommit, CodeBuild, CodeDeploy, CodeStar, Device Farm, Lambda, Amazon EC2 Container Service.

Тема 3. Інструменти розгортання та Docker контейнеризація.

Лекція 7. Configuration Management System.

Інструменти для розгортання хмарного сервера з необхідними програмними та апаратними конфігураціями Terraform, Ansible.

Лекція 8. Системи контролю версій у DevOps

Використання Git та підхід GitOps як основа для автоматизації та відстеження стану інфраструктури й коду.

Лекція 9. Контейнеризація.

Введення до контейнеризації. Docker Hub. Основи Docker. Написання Dockerfile. Private docker registry. Робота з docker-compose. Docker Swarm.

Лекція 10. Оркестрація контейнерів

Kubernetes Архітектура кластера, управління Pod-ами, масштабування та підтримка самоодужання сервісів.

Лекція 11. Helm та управління пакетами

Пакетний менеджер для Kubernetes: створення чартів, версіонування, параметризація та повторні розгортання.

Тема 4. Розгортання сервісів Amazon ECS. Вступ до системи контролю версій Git.

Лекція 12. Робота з Docker та Amazon ECS.

Розгортання програми як набору взаємозалежних сервісів з урахуванням балансувальника навантаження додатків Application Load Balancer (ALB).

Лекція 13. Система контролю версій Git.

Огляд git. Встановлення: Git-server та Gitlab. Основи роботи з Git. Розгалуження в Git.

Тема 5. Безперервна інтеграція та безперервне постачання. Інтеграція з AWS.

Лекція 14. Методологія забезпечення надійності та швидкості створення продукту CI/CD.

Що таке CI/CD. Build та Deploy. Огляд існуючих інструментів. Gitlab CI. Jenkins.

Лекція 15. Централізоване логування

Налаштування ELK/EFK-стека (Elasticsearch, Logstash/Fluentd, Kibana) для аналізу та кореляції логів із різних компонентів.

Лекція 16. Налаштування простої схеми CI/CD за допомогою AWS CodeStar

Огляд спектру шаблонів проєктів для швидкого розробки програм та розгортання в сервісах Amazon EC2, AWS Lambda або AWS Elastic Beanstalk.

Тема 6. Моніторинг та обробка даних.

Лекція 17. Інструменти моніторингу.

Огляд існуючих інструментів моніторингу. Встановлення та використання Grafana.

Тема 6. Моніторинг та обробка даних.

Лекція 18. Збір, зберігання та аналіз даних. Elasticsearch, Logstash, Kibana.

Встановлення та налаштування ELK. Збір логів. Пошук в Elasticsearch. Налаштування фільтрів. Пошук у Kibana.

Практичні заняття

№ з/п	Назва теми заняття та перелік основних питань
1	Тема 1. Розуміння практик DevOps та основи Linux. Що таке DevOps. Огляд курсу. Інсталяція ОС Linux. Підсистема довідки Базові команди. Робота з файлами та каталогами. Текстовий редактор vim. Керування користувачами та групами. Встановлення програмного забезпечення. Управління процесами. Утиліти ps, top, htop, atop. Журналювання. Резервне копіювання. Налаштування параметрів мережі. Програмування на bash.
2	Тема 2. Мережеві технології та DevOps орієнтовані сервіси платформи AWS. Моделі OSI та TCP/IP Огляд технологій локальних мереж. Адресація канального рівня. Введення у маршрутизацію: IP-адресація. Протокол IP. Протокол ICMP. Технологія NAT. Протоколи транспортного рівня: UDP та TCP. Протоколи прикладного рівня: протокол DNS, налаштування bind, unbound, протокол DHCP, налаштування, протокол HTTP, налаштування apache2, nginx.
3	Тема 2 Мережеві технології та DevOps орієнтовані сервіси платформи AWS. Платформа AWS, інструментарій з розробки, розгортання, міграцію даних, послуги з безпеки, мобільні сервіси. Сервіси AWS DevOps: CloudFormation, Cloud Development Kit, CodePipeline, CodeCommit, CodeBuild, CodeDeploy, CodeStar, Device Farm, Lambda, Amazon EC2 Container Service. <i>Практична робота: Знайомство з командним рядком AWS та робота з шаблонами AWS CloudFormation</i>
4	Тема 3. Інструменти розгортання та Docker контейнеризація. Інструменти для розгортання хмарного сервера з необхідними програмними та апаратними конфігураціями Terraform, Ansible.
5	Тема 3. Інструменти розгортання та Docker контейнеризація.

	Введення до контейнеризації. Docker Hub. Основи Docker. Написання Dockerfile. Private docker registry. Робота з docker-compose. Docker Swarm <i>Практична робота: Робота з Docker та Amazon ECS</i>
6	Тема 4. Розгортання сервісів Amazon ECS. Вступ до системи контролю версій Git. Розгортання програми як набору взаємозалежних сервісів з урахуванням балансувальника навантаження додатків Application Load Balancer (ALB). Огляд git. Встановлення: Git-server та Gitlab. Основи роботи з Git. Розгалуження в Git.
7	Тема 5. Безперервна інтеграція та безперервне постачання. Інтеграція з AWS. Що таке CI/CD. Build та Deploy. Огляд існуючих інструментів. Gitlab CI. Jenkins. <i>Практична робота: Налаштування простої схеми CI/CD за допомогою AWS CodeStar</i>
8	Тема 5 Безперервна інтеграція та безперервне постачання. Інтеграція з AWS. Огляд спектру шаблонів проектів для швидкого розробки програм та розгортання в сервісах Amazon EC2, AWS Lambda або AWS Elastic Beanstalk. <i>Робота з AWS OpsWorks для Chef Automate</i>
9	Тема 6. Моніторинг та обробка даних. Огляд існуючих інструментів моніторингу. Встановлення та використання Grafana. Встановлення та налаштування ELK. Збір логів. Пошук в Elasticsearch. Налаштування фільтрів. Пошук у Kibana. <i>Практична робота: Ручне тестування. тестування не функціональних вимог</i>

Самостійна робота студента

№ з/п	Вид самостійної роботи	Кількість годин СРС
1	Тема 1. Розуміння практик DevOps та основи Linux. Напрацювання практичного досвіду в середовищі операційної системи сімейства Linux.	10
2	Тема 2. Мережеві технології та DevOps орієнтовані сервіси платформи AWS. Налаштування віртуальної мережі в середі ОС Linux для практичних занять з курсу.	10
3	Тема 3. Інструменти розгортання та Docker контейнеризація. Розгортання Docker та налаштування взаємодії з різними контейнерами	10
4	Тема 4. Розгортання сервісів Amazon ECS. Вступ до системи контролю версій Git. Створення навчального акаунту в Amazon ECS. Апробація сервісів.	10
5	Тема 5. Безперервна інтеграція та безперервне постачання. Інтеграція з AWS. Розгортання CI/CD. Використання шаблонів AWS.	10
6	Тема 6. Моніторинг та обробка даних. Встановлення Grafana, ELK, підключення до напрацьованих сервісів.	10
7	Підготовка до заліку	6

Контрольна робота

Метою контрольної роботи є закріплення та перевірка теоретичних знань із освітнього компонента, набуття студентами практичних навичок самостійного вирішення задач та складанні та компіляції програм.

Модульна контрольна робота (МКР) виконується після вивчення тем 1-6 та виконання практичних занять 1-5. Контрольна робота проводиться у середовищі Linux з використанням інструментарію Docker. Кожен студент отримує індивідуальне завдання, відповідно до якого необхідно виконати розгортання програмного комплексу, реалізація поставленої задачі та моніторинг якості виконання програмної складової задачі.

6. Політика навчальної дисципліни (освітнього компонента)

Відвідування лекційних та лабораторних занять є обов'язковим за винятком поважних причин (хвороби, форс-мажорних обставин).

В разі пропущення занять з поважних причин викладач надає можливість студенту виконати усі або деякі лабораторні завдання (винятком є виконання деяких завдань у зв'язку із закінченням навчального процесу).

В разі пропущення занять без поважних причин, а також через порушення граничного терміну виконання завдання (deadline) студент може отримати 80% від максимальної оцінки відповідне завдання.

Протягом семестру студенти:

- виконують та захищають лабораторні роботи у відповідні терміни (на кожен лабораторну роботу відводиться два тижні для здачі),
- пишуть модульну контрольну роботу,
- повинні позитивно закрити дві атестації (в кінці березня та в середині травня),
- по закінченні навчального процесу складають екзамен.

Система вимог, які викладач ставить перед студентом:

- правила відвідування занять: заборонено оцінювати присутність або відсутність здобувача на аудиторному занятті, в тому числі нараховувати заохочувальні або штрафні бали. Відповідно до РСО даної дисципліни бали нараховують за відповідні види навчальної активності на лекційних та практичних заняттях.
- правила поведінки на заняттях: студент має можливість отримувати бали за відповідні види навчальної активності на лекційних та практичних заняттях, передбачені РСО дисципліни. Використання засобів зв'язку для пошуку інформації на гугл-диску викладача, в інтернеті, в дистанційному курсі на платформі Сікорський здійснюється за умови вказівки викладача;
- політика дедлайнів та перескладань: якщо студент не проходив або не з'явився на МКР (без поважної причини), його результат оцінюється у 0 балів. Перескладання результатів МКР не передбачено;
- політика щодо академічної доброчесності: Кодекс честі Національного технічного університету України «Київський політехнічний інститут» <https://kpi.ua/files/honorcode.pdf> встановлює загальні моральні принципи, правила етичної поведінки осіб та передбачає політику академічної доброчесності для осіб, що працюють і навчаються в університеті, якими вони мають керуватись у своїй діяльності, в тому числі при вивченні та складанні контрольних заходів з дисципліни «Системи автоматизації»;
- при використанні цифрових засобів зв'язку з викладачем (мобільний зв'язок, електронна пошта, переписка на форумах та у соцмережах тощо) необхідно дотримуватись загальноприйнятих етичних норм, зокрема бути ввічливим та обмежувати спілкування робочим часом викладача.

7. Види контролю та рейтингова система оцінювання результатів навчання (РСО)

Поточний контроль: вправи на лекційних заняттях, тестування, МКР, виконання завдань до практичних занять, виконання та захист лабораторних робіт.

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

Семестровий контроль: залік.

Система рейтингових (вагових) балів та критерії оцінювання

1) Робота на лекціях

На лекціях може бути проведено бліцопитування студентів. Такі опитування проводяться на довільних лекціях 5 разів протягом семестру, наприкінці лекції. Ваговий бал за вірну відповідь - 1. Максимальна кількість балів, що може отримати кожен студент за семестр - 5.

2) Лабораторні практикуми

Максимальна кількість балів за усі виконані комп'ютерні практикуми дорівнює 60 балів. Розподіл балів серед лабораторних практикумів наступний:

№ з/п	Назва лабораторного практикуму	Кількість балів
1	Знайомство з командним рядком AWS та робота з шаблонами AWS CloudFormation	12
2	Налаштування простої схеми CI/CD за допомогою AWS CodeStar	12
3	Робота з AWS OpsWorks для Chef Automate	12
4	Робота з Docker та Amazon ECS	12
5	Ручне тестування. тестування не функціональних вимог	12
Всього:		60

Критерії оцінювання:

Виконання лабораторного практикуму:

- виконаний своєчасно (протягом двох тижнів з моменту видачі), у повному обсязі – відповідний бал згідно номеру комп'ютерного практикуму;
- виконаний із запізненням – знімається 10 – 30% від максимальної кількості балів в залежності від терміну запізнення;
- виконаний не самостійно, із запізненням – знімається 50% від максимальної кількості балів;
- невиконаний протягом відведеного часу – 0 балів.

3) Модульна контрольна робота

Максимальна кількість балів за модульну контрольну роботу дорівнює 10 балів.

Якість виконання роботи:

- усі відповіді вірні та повні – 10 балів,
- у відповідях допущені несуттєві неточності – 8 балів,
- половина відповідей вірна – 5 балів,
- відповіді з суттєвими неточностями, але без критичних помилок – 2 бали,
- менше половини відповідей вірна – 0 балів.

Штрафні та заохочувальні бали за:

- | | |
|--|--------------|
| - активність на комп'ютерних практикумах | + 2 бали |
| - виконання комп'ютерного практикуму з використанням власного оптимального алгоритму | + 1 бали |
| - відсутність на занятті без поважної причини | - 2 бали |
| - несвоєчасна здача комп'ютерного практикуму (пізніше ніж за тиждень) | - 0,5 балів; |

4) Складання заліку

Максимальний ваговий бал $r_{\text{ісп}}=40$

На іспиті студент виконує письмову контрольну роботу, яка містить два теоретичних питання і одне практичне питання. Теоретичні питання оцінюються максимально по 10 балів, практичне – 20 балів.

Умови позитивної проміжної атестації

Для отримання „зараховано” з першої проміжної атестації студент матиме не менше ніж 30 балів (за умови, що за 8 тижнів згідно з календарним планом контрольних заходів „ідеальний” студент має отримати $12 + 12 + 12 = 36$ бал).

Для отримання „зараховано” з другої проміжної атестації студент матиме не менше ніж 50 балів (за умови, що за 14 тижнів згідно з календарним планом контрольних заходів „ідеальний” студент має отримати $36 + 12 + 12 = 60$ балів).

Умови допуску до заліку

Необхідною умовою допуску до заліку є зарахування усіх комп’ютерних практикумів та виконання модульної контрольної роботи, а також стартовий рейтинг (R_c) не менше 40 балів.

Розрахунок шкали (R) рейтингу:

Сума вагових балів контрольних заходів протягом семестру (шкала рейтингу) складає:

$$R = r_{\text{лек}} + r_{\text{практ}} + r_{\text{мод}} + r_{\text{ісп}} = 5 + 45 + 10 + 40 = 100 \text{ балів.}$$

Максимальний стартовий рейтинг становить $R_c = r_{\text{лек}} + r_{\text{практ}} + r_{\text{мод}} = 60$ балів.

Рейтинг заліку дорівнює 40 балів. Мінімальний рейтинг допуску до заліку становить 40 балів.

Таким чином, рейтингова шкала з кредитного модуля складає

$$R = 60 + 40 = 100 \text{ балів.}$$

Для отримання студентом відповідних оцінок рейтингова оцінка студента переводиться згідно таблиці відповідності рейтингових балів оцінкам за університетською шкалою:

Бали	Оцінка
95 - 100	Відмінно
85 - 94	Дуже добре
75 - 84	Добре
65 - 74	Задовільно
60 - 64	Достатньо
Менше 60	Незадовільно
$R < 40$ є незараховані роботи комп’ютерного практикуму або не виконані інші умови допуску до екзамену	Не допущено

8. Додаткова інформація з дисципліни (освітнього компонента)

Перелік питань, які виносяться на семестровий контроль:

1. DevOps. Адаптивні алгоритми. Стандартизація середовищ розробки.
2. Керування користувачами та групами Linux.
3. Управління процесами Linux.
4. Налаштування параметрів мережі Linux.
5. Програмування на bash Linux.
6. Маршрутизація: IP-адресація. Протокол IP. Протокол ICMP. Технологія NAT.
7. Git. Управління версіями.
8. Протоколи транспортного рівня: UDP та TCP.
9. Протоколи прикладного рівня: протокол DNS, налаштування bind, unbound, протокол DHCP, налаштування, протокол HTTP, налаштування apache2, nginx.
10. Платформа AWS, інструментарій з розробки, розгортання, міграція даних, послуги з безпеки, мобільні сервіси.
11. Робота з AWS OpsWorks для Chef Automate
12. Інструмент для розгортання хмарного сервера з необхідними програмними та апаратними конфігураціями Terraform.
13. Інструмент для розгортання хмарного сервера з необхідними програмними та апаратними конфігураціями Ansible.
14. Використання програм безперервної інтеграції. Розробка через тестування. Пропоновані методики.
15. Автоматичні приймальні випробування.

16. Рекомендовані методики автоматичного приймального тестування. Ручне тестування.
17. Автоматизація розгортання та постачання релізу. Відкат змін.
18. Тестування не функціональних вимог. Підготовка до випуску.
19. Сценарії збірки і розгортання.
20. Розгортання тестових середовищ та їх тестування.
21. Альтернативні підходи до безперервної інтеграції.
22. Сервіс Docker Hub.
23. Написання Dockerfile.
24. Private docker registry.
25. Робота з docker-compose.
26. Платформа оркестрування контейнерів з відкритим кодом Docker Swarm.
27. Робота з Git.
28. Розгалуження в Git.
29. Системи CI/CD реалізація Build та Deploy.
30. Gitlab CI. Jenkins.
31. Використання Grafana.
32. Налаштування ELK.
33. Збір логів ELK.
34. Налаштування фільтрів ELK.

Складено старшим викладачем ІПЗЕ, Колумбетом Вадимом Петровичем

Ухвалено кафедрою ІПЗЕ (протокол №42 від 18.06.2025 р.)

Погоджено Методичною комісією факультету (протокол № 09 від 27.06.2025 р.)