

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

1. Загальна інформація про навчальну дисципліну

Повна назва навчальної дисципліни	Технічна підтримка програмного забезпечення
Повна офіційна назва закладу вищої освіти	Сумський державний університет
Повна назва структурного підрозділу	Факультет електроніки та інформаційних технологій. Кафедра комп'ютерних наук
Розробник(и)	Кузіков Борис Олегович
Рівень вищої освіти	Перший рівень вищої освіти, НРК – 6 рівень, QF-LLL – 6 рівень, FQ-EHEA – перший цикл
Семестр вивчення навчальної дисципліни	8 тижнів протягом 8-го семестру
Обсяг навчальної дисципліни	Обсяг становить 5 кред. ЄКТС, 150 год. Для денної форми навчання 40 год. становить контактна робота з викладачем (16 год. лекцій, 24 год. лабораторних занять), 110 год. становить самостійна робота.
Мова викладання	Українська

2. Місце навчальної дисципліни в освітній програмі

Статус дисципліни	Обов'язкова навчальна дисципліна для освітньої програми "Інформатика"
Передумови для вивчення дисципліни	Бази даних та інформаційні системи
Додаткові умови	Додаткові умови відсутні
Обмеження	Обмеження відсутні

3. Мета навчальної дисципліни

Навчити студентів універсальній методиці пошуку помилок, дати навички роботи у команді, ознайомити з основними видами програмного забезпечення, що застосовується для підтримки програмних продуктів.

4. Зміст навчальної дисципліни

Модуль 1. Технічна підтримка програмного забезпечення
Тема 1 Вступ до TSS
Місце дисципліни у професійній діяльності студента. Життєвий цикл програми. Основні підходи до організації розробки програмного забезпечення.
Тема 2 Узагальнений підхід до усунення дефектів
Узагальнений підхід до усунення дефектів. Систематичний підхід до пошуку та усунення помилок. Методи відокремлення причин та наслідків
Модуль 2. Спеціальне програмне забезпечення етапу розробки ПЗ
Тема 3 Базовий інструментарій програміста
Інструментарій програміста. Відлагоджування коду. Інтегроване середовище розробки JetBrains IntelliJ IDEA
Тема 4 Робота у команді
Базові принципи роботи у команді. Стиль кодування. Документування програм та самодokumentований код. Системи контролю версій. Системи відслідкування помилок. Програмне забезпечення для збірки коду.
Тема 5 Спеціальний інструментарій програміста
Статичний аналіз коду. Спеціальне програмне забезпечення для пошуку дефектів. Системи автоматичної збірки коду. Безперервна інтеграція: поняття і засоби.
Модуль 3. Спеціальне програмне забезпечення етапу експлуатації ПЗ
Тема 6 Швидкодія застосувань
Розв'язання проблем із продуктивністю застосувань. Інструментальні засоби вимірювання швидкодії
Тема 7 Пошук дефектів у багатокомпонентних системах
Діагностика дефектів, пов'язаних із стороннім ПО. Діагностика проблем з мережею, обладнанням, програмами третіх осіб, веб-серверами та серверами баз даних.
Тема 8 Програмний код у реальних проектах
Підходи, практики, програмне забезпечення, що використовується у сучасних колективах розробників. Аналіз досвіду провідних команд розробників.

5. Очікувані результати навчання навчальної дисципліни

Після успішного вивчення навчальної дисципліни здобувач вищої освіти зможе:

РН1	забезпечувати організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності програмного забезпечення.
РН2	організувати управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника. Провести документацію та виправлення дефектів.
РН3	вибрати та ефективно використовувати сучасні парадигми та мови програмування, платформи реалізації, методи і стандарти розробки програмного забезпечення.
РН4	застосовувати знання у практичних ситуаціях з урахуванням існуючих ресурсів та обмежень
РН5	проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління. Оцінювати його функціональну ефективність.

6. Роль навчальної дисципліни у досягненні програмних результатів

Програмні результати навчання, досягнення яких забезпечує навчальна дисципліна.

Для спеціальності 122 Комп'ютерні науки:

ПР11	Використовувати інструментальні засоби розробки клієнт-серверних застосувань, проектувати концептуальні, логічні та фізичні моделі баз даних, розробляти та оптимізувати запити до них, створювати розподілені бази даних, сховища та вітрини даних, бази знань, у тому числі на хмарних сервісах, із застосуванням мов веб-програмування.
ПР13	Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення.
ПР14	Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення.
ПР16	Розуміти концепцію інформаційної безпеки, принципи безпечного проектування програмного забезпечення, забезпечувати безпеку комп'ютерних мереж в умовах неповноти та невизначеності вихідних даних.
ПР21	Реалізувати заходи із оцінювання і підвищення ефективності функціонування інфокомунікаційних систем, керування їх ресурсами і задачами, якості обслуговування користувачів інфокомунікаційних сервісів.

7. Роль освітнього компонента у формуванні соціальних навичок

Загальні компетентності та соціальні навички, формування яких забезпечує навчальна дисципліна:

СН1	Здатність до абстрактного мислення, аналізу та синтезу.
СН6	Здатність вчитися й оволодівати сучасними знаннями
СН9	Здатність працювати в команді
СН10	Здатність бути критичним і самокритичним

8. Види навчальних занять

Тема 1. Вступ до TSS
Лк1 "Витоки та основні поняття технічної підтримки програмних продуктів" Основні поняття дисципліни. Діяльність Technical Solution Engeneer
Лб1 "Вступ." Загальна ний план виконання завдань дисципліни
Лб2 "Версіювання. Автоматична збірка та тестування. Внесення змін у програмні продукти." Стратегії створення гілок у системах управління вихідним кодом. Документування коду. Регресійне тестування.
Тема 2. Узагальнений підхід до усунення дефектів
Лк2 "Troubleshooting та підходи до нього" Узагальнений підхід до усунення дефектів (УМ). Методики "5 Чому", "Діаграма Ісікави"
Лб3 "Декомпіляція та влагоджування коду" Декомпіляція та влагоджування коду. Віддалене відлагоджування коду.
Тема 3. Базовий інструментарій програміста
Лк3 "Інструменти та підходи для організації взаємодії із кодом" Засоби компіляції, декомпіляції, документування, влагоджування. Програмне забезпечення для збірки коду.

<p>Лб4 "Обговорення результатів самостійного опрацювання матеріалу"</p> <p>Як ви організували розподіл задач та відслідковування активності? Стратегія створення гілок, яку ви обрали на проєкті. Чому? Як перевірити, що тестовий сервер може прийняти дані від клієнта? (Postman, ...) Тестування АРІ до виходу у Production. Програмне забезпечення, яке ви використовуєте (на проєкті та для розробки проєкту)</p>
<p>Лб5 "Обговорення результатів самостійного опрацювання матеріалу"</p> <p>Як налаштувати сервер? Як налаштувати https на сервері? Як налаштувати WAF, фایрвол, чи fail2ban. Як дізнатись які налаштування були змінені на сервері? (Etckeeper, аналоги)</p>
<p>Тема 4. Робота у команді</p>
<p>Лк4 "Командний підхід при роботі із кодом"</p> <p>Базові принципи роботи у команді. Стил ь кодування. Документування програм та самодокументований код. Системи контролю версій. Системи від слідкування помилок.</p>
<p>Тема 5. Спеціальний інструментарій програміста</p>
<p>Лк5 "Інструменти та підходи для підвищенні ефективності взаємодії із кодом"</p> <p>Статичний аналіз коду. Спеціальне програмне забезпечення для пошуку дефектів. Системи автоматичної збірки коду. Безперервна інтеграція: поняття і засоби.</p>
<p>Лб6 "Обговорення результатів самостійного вивчення матеріалу"</p> <p>Як реалізована практика Static Analyse на вашому проєкті. Які проблеми були виявлені? Як реалізована авторизація, збереження паролів? (до БД, серверів, користувачів сервісу) Як реалізовано журналювання (logging) у вашому проєкті ? Практика Code Review на вашому проєкті. Як впевнитись, що на проєкті не буде неперевіреного коду.</p>
<p>Лб7 "Журналювання додатку. Сервіси агрегації помилок"</p> <p>Logging, Log rotate Сервіси агрегації помилок (Rollbar, Sentry, Airbags)</p>
<p>Лб8 "Обговорення результатів самостійного опрацювання матеріалу"</p> <p>Моніторинг доступності сервісу. Сервіси збору та агрегації помилок. (Rollbar, Airbrake, ...) Скільки зберігаються журнали роботи ПЗ сервера. logrotate Процес оновлення ПЗ вашого серверу. Алгоритм дій команди. Який down-time при оновленні? Що бачать користувачі під час down-time?</p>
<p>Тема 6. Швидкодія застосувань</p>
<p>Лк6 "Критерії забособи та підходи до оцінки ефективності застосунків"</p> <p>Розв'язання проблем із продуктивністю застосувань. Інструментальні засоби вимірювання швидкодії.</p>

Лб9 "Засоби моніторингу додатку"
Засоби моніторингу додатку
Лб10 "Обговорення результатів самостійного вивчення матеріалу"
Моніторинг стану додатку (які метрики збираються і як) Профілювання? Які частини коду виконуються найдовше? АОТ компіляція Tomkat (або інші сервлет-контейнери)
Тема 7. Пошук дефектів у багатокомпонентних системах
Лк7 "Пошук дефектів у багатокомпонентних системах"
Діагностика дефектів, пов'язаних із стороннім ПО. Діагностика проблем з мережею, обладнанням, програмами третіх осіб, веб-серверами та серверами баз даних.
Тема 8. Програмний код у реальних проєктах
Лк8 "Особливості організації коду у сучасних проєктах"
Підходи, практики, програмне забезпечення, що використовується у сучасних колективах розробників. Аналіз досвіду провідних команд розробників.
Лб11 "Обговорення результатів самостійного опрацювання матеріалу"
Горизонтальне масштабування. На поточний момент ресурсів сервера вистачає. Для проєкту можна залучити ще один, два або X веб-серверів з такими самими або іншими (якими) характеристиками. Надайте для замовника 3 можливі варіанти, як можна витратити його гроші, так щоб він погодився на це. Горизонтальне масштабування. На поточний момент ресурсів сервера НЕ вистачає. Для проєкту можна залучити ще один, два або X веб-серверів з такими самими або іншими (якими) характеристиками. Надайте для замовника 3 можливі варіанти, як можна витратити його гроші, так щоб він погодився на це. Що я буду робити, якщо сервер раптово "помре"? Як знизити час недоступності сервера у разі перезавантаження? (автоматичний старт ПЗ)
Лб12 "Захист проєкту"
Демонстрація програмно-технічних заходів забезпечення якості та стабільності ПЗ

9. Стратегія викладання та навчання

9.1 Методи викладання та навчання

Дисципліна передбачає навчання через:

МН1	Командно-орієнтоване навчання (TBL)
МН2	Лекційне навчання
МН3	Самостійне навчання

У основу дисципліни покладено командну роботу по підтримці програмного проєкту та вирішення практичних задач, що є типовими для етапу підтримки. Власний досвід студенти

поширюють через серію доповідей "Lessons learned". Лекційний матеріал є проблемно-орієнтованим та висвітлює ключові проблеми та шляхи їх вирішення у сучасних ІТ проєктах.

навички комунікації, здатність брати на себе відповідальність і працювати в критичних умовах, вміння управляти своїм часом, розуміння важливості дедлайнів, здатність логічно і системно мислити.

9.2 Види навчальної діяльності

НД2	Виконання групового практичного завдання
НД3	Вебінари
НД4	Робота з підручниками та релевантними інформаційними джерелами

10. Методи та критерії оцінювання

10.1. Критерії оцінювання

Визначення	Чотирибальна національна шкала оцінювання	Рейтингова бальна шкала оцінювання
Відмінне виконання лише з незначною кількістю помилок	5 (відмінно)	$90 \leq RD \leq 100$
Вище середнього рівня з кількома помилками	4 (добре)	$82 \leq RD < 89$
Загалом правильна робота з певною кількістю помилок	4 (добре)	$74 \leq RD < 81$
Непогано, але зі значною кількістю недоліків	3 (задовільно)	$64 \leq RD < 73$
Виконання задовольняє мінімальним критеріям	3 (задовільно)	$60 \leq RD < 63$
Можливе повторне складання	2 (незадовільно)	$35 \leq RD < 59$
Необхідний повторний курс з навчальної дисципліни	2 (незадовільно)	$0 \leq RD < 34$

10.2 Методи поточного формативного оцінювання

	Характеристика	Дедлайн, тижні	Зворотний зв'язок
МФО1 Опитування та усні коментарі викладача за його результатами	Фронтальні опитування та обговорення на лекціях та лабораторних заняттях	Згідно до регламенту	Zoom

10.3 Методи підсумкового сумативного оцінювання

	Характеристика	Дедлайн, тижні	Зворотний зв'язок
МСО2 Звіт за результатами виконання лабораторних робіт	Весь обсяг робіт по роботі над проектом розбито на кілька етапів. Звітність передбачається за кожним з етапів.	Згідно до регламенту	Zoom
МСО3 Виконання практичного кейсу (підготовка, презентація, захист)	Студент розглядає індивідуальну тему у рамках групового задавання, що розширює розуміння предмету. За виконаною роботою звітує у форматі доповіді.	Згідно до регламенту	Zoom

Контрольні заходи:

	Максимальна кількість балів	Мінімальна кількість балів	Можливість перескладання з метою підвищення оцінки
8 семестр	100 балів		
МСО2. Звіт за результатами виконання лабораторних робіт	70		
	70	Не передбачено	Так
МСО3. Виконання практичного кейсу (підготовка, презентація, захист)	30		
	30	Не передбачено	Ні

Практична складова дисципліни передбачає командне виконання проекту із наперед виділеними етапами. Звітністю про виконання окремих етапів є презентації щодо набутого досвіду. Сумарно доповіді оцінюються у 30 балів. Звітністю про виконання проекту є підсумковий звіт із описом виконаної роботи. Звіт оцінюється оцінкою до 70 балів відповідно до повноти виконання завдання.

11. Ресурсне забезпечення навчальної дисципліни

11.1 Засоби навчання

ЗН1	Інформаційно-комунікаційні системи
ЗН2	Прикладне програмне забезпечення (перелік конкретизується викладачем)
ЗН3	Проекційна апаратура

11.2 Інформаційне та навчально-методичне забезпечення

Основна література	
1	Чиста архітектура: мистецтво розроблення програмного забезпечення [Текст] / Р. Мартін; пер. з англ. І. Бондар-Терещенко. — Х. : Ранок; Фабула, 2019. — 368 с. — ISBN 978-617-09-5286-8.
Допоміжна література	
2	Software Development Fundamentals [Електронний ресурс]. — University of Pennsylvania, 2020.
3	Якість програмного забезпечення та тестування: базовий курс [Електронний ресурс] : навч. посіб. / С. Я. Крепич, І. Я. Співак. — Тернопіль : Паляниця В.А., 2020. — 478 с.
4	4561 Методичні вказівки до практичних занять із дисципліни "Реінжиніринг і верифікація програмного забезпечення" на тему "Пошук недоліків коду й використання інструментів рефакторингу" [Електронний ресурс] : для студ. спец. 122 "Комп'ютерні науки" всіх форм навчання / В. В. Москаленко. — Електронне видання каф. Комп'ютерних наук. — Суми : СумДУ, 2019. — 37 с.
5	Проектування програмних доданків [Електронний ресурс]. Ч.1 : Комп'ютерні практикуми / уклад.: В. І. Бендюг, Б. М. Комариста. — Київ : КПІ ім. Ігоря Сікорського, 2018. — 285 с.
Інформаційні ресурси в Інтернеті	
6	Конспект лекцій. Кузіков Б.О. Дистанційний курс з дисципліни “Технічна підтримка програмних рішень” [Електронний ресурс]. Режим доступу: https://mix.sumdu.edu.ua/info/nmk/fb51a407-d221-4ef4-ab51-a29940287358
7	Tutorials [електронний ресурс] Режим доступу: https://javalin.io/tutorials