

ДЕРЖАВНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
СИСТЕМА УПРАВЛІННЯ ЯКІСТЮ
Система забезпечення якості освітньої діяльності та якості вищої освіти
сертифікована на відповідність ДСТУ ISO 9001:2015 / ISO 9001:2015
Кафедра інженерії програмного забезпечення та кібербезпеки

**КОГНІТИВНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ/
COGNITIVE INFORMATION TECHNOLOGIES**

**СИЛАБУС/
SILABUS**

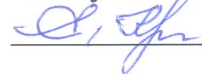
ЗАТВЕРДЖЕНО

засіданням кафедри

(протокол №. 1



від «07» серпня 2024 р.)

завідувач кафедри



Олена КРИВОРУЧКО

Київ 2024

Назва освітньої компоненти	КОГНІТИВНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ/ COGNITIVE INFORMATION TECHNOLOGIES
Спеціальність	121 «Інженерія програмного забезпечення»
Освітній ступінь	Другий (магістерський)
Освітньо-професійна програма	УПРАВЛІННЯ ПРОЄКТАМИ ПРОГРАМНИХ ПРОДУКТІВ
	<p>Лектор: Криворучко Олена</p> <p>-завідувач кафедри інженерії програмного забезпечення та кібербезпеки, гарант освітньої програми «Комп'ютерні науки» (PhD) -доктор технічних наук -пофесор</p> <p>Резюме викладача: https://knute.edu.ua/blog/read/?pid=39648&uk Науковий профіль: https://knute.edu.ua/blog/read/?pid=46714 е-пошта: kryvoruchko_ev@knute.edu.ua</p>
	<p>Асистент лектора: Сергій Бульба</p> <p>-ст.викладач, гарант освітньої програми «Інженерія програмного забезпечення» -к.е.н.</p> <p>Резюме викладача: https://knute.edu.ua/blog/read/?pid=48084&uk Практична діяльність: Senior System Analyst, Solution Architect е-пошта: s.bulba@knute.edu.ua</p>
Консультації	https://knute.edu.ua/blog/read/?pid=47103&uk
Програма освітньої компоненти	https://knute.edu.ua/blog/read/?pid=48215
ЗМІСТ ОСВІТНЬОЇ КОМПОНЕНТИ	
Тема 1. Вступ до когнітивних технологій	<p>Основні елементи когнітивної обчислювальної системи. Адаптивність, інтерактивність, самонавчання та контекстуальність когнітивних систем. Відповідність когнітивних систем своєму призначенню. Спрощення роботи людини зі своїм інформаційним оточенням.</p> <p>Сфери застосування когнітивних систем у бізнесі, охороні здоров'я, спорті. Залучення, експертиза, продукти та сервіси, відкриття та прийняття рішень – випадки використання когнітивних технологій в бізнесі</p>
Тема 2.	Збентеження як невід'ємна частина програмування. Типи

Збентеження при кодуванні	збентеження у кодї: недостача знань, недостача інформації, недостача обчислювальної потужності. Недостача знань як проблема з довготривалою пам'яттю при програмуванні. Довготривала пам'ять як жорсткий диск комп'ютера. Недостача інформації як проблема з короткотривалою пам'яттю в програмуванні. Короткотривала пам'ять як оперативна пам'ять комп'ютера. Недостача обчислювальної потужності як проблема з робочою пам'яттю в програмуванні. Робоча пам'ять як процесор комп'ютера. Опис когнітивних процесів та їх взаємодія один з одним. Активізація когнітивних процесів в процесі програмування коду.
Тема 3. Швидкочитання коду	Витрати часу на розуміння та написання програмного коду. Вміння знаходити необхідну інформацію в кодї. Когнітивні процеси, що виникають при запам'ятовуванні коду. Переперевірка відтвореного коду. Обмежена місткість короткотривалої пам'яті, як головна причина у складності відтворення коду по пам'яті. Подолання лімітів пам'яті. Адріан де Гроот та його теорія чанків. Переваги слів над незнайомими символами та окремими буквами. Експерименти Кетрін МакКітен стосовно початківців-програмістів. Сенсорна пам'ять як бувер введення-виведення. Іконічна пам'ять та інформація органів зору. Написання коду, що можна розділити на чанки. Використання патернів (шаблонів) проектування, коментарів для коду. Проставлення «маячків» для визначення в кодї структур даних, алгоритмів або концепцій. Зрозумілі без пояснень елементи синтаксису коду та крупні елементи коду
Тема 4. Синтаксис мов програмування	Запам'ятовування синтаксису, концепцій програмування та структур даних для швидшої обробки коду. Зниження ефективності читання коду. Експерименти Кріса Парніна стосовно відволікання у роботі програмістів. Використання дидактичних карток для швидкого вивчення синтаксису. Правила використання карток: їх розширення та вилучання. Необхідність в оновленні знань, перш ніж інформація запам'ятеється на тривалий час. Мережна модель зберігання інформації. Формула Ебінгауза про тривалість зберігання інформації. Правило вивчення коду зі збільшеними інтервалами. Механізми видобування інформації з довготривалої пам'яті. Проблема відтворення при пошуку певного синтаксису. Укріплення пам'яті шляхом активного мислення. Спогади та зв'язки між ними, що формують схеми. Пропрацювання для запам'ятовування концепцій програмування.
Тема 5. Читання складного коду	Складність коду для розуміння. Причини важкого розуміння складного коду. Відмінність між робочою та короткочасною пам'яттю. Типи когнітивного навантаження та їх зв'язок з програмуванням. Внутрішнє та зовнішнє когнітивне

	<p>навантаження в процесі читання коду. Способи зниження когнітивного навантаження. Рефакторинг як перепроєктування без зміни його інтерфейсів. Зворотній рефакторинг. Заміна незнайомих мовних конструкцій. Звичайна функція замість лямбда-функції. Заміна генератора списків циклом for. Заміна тернарного оператора умовною конструкцією if-else. Синонімізація як доповнення до дидактичних карток. Створення графу залежностей для опрацювання коду. Використання таблиць станів для когнітивної компіляції коду.</p>
<p>Тема 7. Рішення задач програмування</p>	<p>Розгляд різних варіантів рішення задач. Використання моделей для міркування про код. Переваги використання моделей. Важливість представлення та його вплив на рішення задач програмування. Ментальні моделі Кеннета Крейка в міркуванні про код. Характеристики ментальних моделей з прикладами програмування. Ефективність використання ментальних моделей під час міркування про код. Кроки формування ментальної моделі складного коду в робочій пам'яті. Створення ментальних моделей початкової програми в довготривалій пам'яті. Концепція умовної машини у дослідженнях мов програмування. Правильна абстракція виконання мови програмування. Приклади умовних машин та її рівні. Розширення набору умовних машин мов програмування. Взаємно конфліктні ментальні моделі умовних машин.</p>
<p>Тема 8. Концепції когнітивного навантаження</p>	<p>Написання коду з когнітивної точки зору. Огляд коду з вадами за Фаулером та рівнів, до яких вони відносяться. Код з вадами на рівні методу, класу, бази коду. Вплив коду з вадами. Види кодів з вадами: «всемогутній метод», «великий клас», «лінивий клас», «дублювання коду», «ланцюжок викликів», тощо. Вміст класу досить малої кількості полів та методів та його зрозумілість в абстрактному представленні. Вплив коду з вадами на мислення. «Довгий список параметрів» та перевантаження робочої пам'яті. «Всемогутній клас» та неможливість ефективно розбити код на чанки. «Клони коду», що не дозволяють правильно розбити код на чанки. Залежність когнітивного навантаження від поганих імен. Лінгвістичні антипатерни проектування програмного коду Арнаудової. Вимір когнітивного навантаження: шкала Пааса та біометрика. Дев'ятибальна шкала оцінювання когнітивного навантаження програмістів. Когнітивні проблеми, що виникають через лінгвістичні антипатерни</p>
<p>Тема 8. Рішення складних задач.</p>	<p>Стратегії, які покращують навички програмування та рішення задач. Елементи з яких складається рішення задач. Простір станів задач як допустимі кроки при рішенні задач. Роль довготривалої пам'яті при рішенні задач програмування. Рішення задач як окремий когнітивний процес. Робота кроків Полія з задачею програмування. Види пам'яті, що найбільш</p>

	<p>суттєві при рішенні задачі програмування. Процедурна та декларативна пам'ять. Семантична або пам'ять фактів. Етапи збереження інформації: когнітивний, автономний. Вплив автоматизації на пришвидшене програмування. Використання усвідомленої практики для покращення навичок. Навчання на основі коду та його пояснення. Відповідне навантаження як процес передачі інформації з програмування знову в довготривалу пам'ять. Джерела, що допомагають вивчати код.</p>
<p>Тема 9. Проектування великих систем.</p>	<p>Методика бази коду з когнітивної точки зору. Технічна частина аспектів мов програмування: парадигма, система типів, компіляція. Концепція когнітивних вимірювань, CDN, для оцінки зручності використання баз коду. Схильність до помилок через відсутність строгої системи типів. Демонстрація погодженості як визначення функцій та схожості елементів. Розмитість в позначенні місця, яке займає елемент коду з точки зору функціональності. Вимір схованих залежностей для користувача. Завчасна фіксація рішення в легкості розуміння використання інструменту розробки програмного забезпечення. Складність внесення змін у вже існуючу кодову базу через в'язкість. Легкість перевірки або виконання незакінченого коду в поетапному порівнянні. Виразність ролей у легкості розуміння різних частин коду. Визначення наближеності мови програмування або коду до предметної області задачі у вимірі близькості відповідності. Створення користувачем системи власних потужних абстракцій. Проектні маневри та їх переваги і недоліки. Вплив вимірів на різні активності</p>
<p>Тема 10. Елементи машинного навчання як складова когнітивних інформаційних технологій.</p>	<p>Поняття машинного навчання. Види машинного навчання. Сучасні бізнес-кейс проектів машинного навчання. Розуміння основних складових та параметрів машинного навчання. Регресійні моделі. Класифікаційні моделі. Кластерні моделі. Deep-learning моделі. Архітектурні принципи формування системи з елементами машинного навчання. Особливості роботи з даними та питання compliance.</p>
<p style="text-align: center;">СПИСОК ОСНОВНИХ РЕКОМЕНДОВАНИХ ДЖЕРЕЛ Основний</p> <ol style="list-style-type: none"> 1. Felienne Hermans “The Programmer's Brain: What every programmer needs to know about cognition”. Manning, 2021. 256 pages 2. John Sonmez “Soft Skills: The Software Developer's Life Manual”. Simple Programmer, LLC, 2020. 501 pages 3. Dr. X.Y. Wang “Ace the Coding Interview: Must-know Questions (Advanced Topics in Programming)”. Independently published, 2023. 280 pages 4. Базурін, В. М. Теорія інформації та кодування [Електронний ресурс] [Текст] : зб. тест. завдань / В. М. Базурін ; каф. комп'ютер. наук та інформ. систем. – 05-13/23. – Київ : ДТЕУ, 2023. – 52 с. 	

5. Функціональне та логічне програмування [Текст] [Текст] : опор. конспект лекцій / авт.: Т.В. Савченко, М.В. Сашньова; каф. інженерії програм. забезп. та кібербезпеки. – Київ : КНТЕУ, 2020. – 146 с. – Бібліогр.: с. 146. – 58.00.

Додатковий

6. Thomas Mannigel “FAST Solo Development: A step-wise method for crafting lucrative single programmer projects rapidly”. Independently published, 2022. 311 pages
7. Нестерова, М. Інформаційно-когнітивні технології в системі вищої освіти суспільства знань [Текст] / М. Нестерова // Вища освіта в Україні. – Київ, 2015. – № 1 (56). – С. 40-45.
8. Рубанець, О. Когнітивні технології у вищій освіті [Текст] / О. Рубанець // Вища освіта України. – Київ, 2017. – № 4. – С. 28-34.
9. Об'єктно-орієнтоване програмування [Електронний ресурс] [Текст] : метод. рек. до виконання курс. роботи / О. В. Криворучко, С. Л. Рзаєва, Б. Т. Бебешко, ін. та ; каф.інженерії програм. забезп. та кібербезпеки. – ПП 04-12/22. – Київ : ДТЕУ, 2022. – 70 с.
10. Cognitive Technology:
<https://www.sciencedirect.com/topics/computer-science/cognitive-technology>
11. What is cognitive technology?
<https://www.icaew.com/technical/technology/technology-and-the-profession/risks-and-assurance-of-emerging-technologies/what-is-cognitive-technology>
12. Cognitive technology – supporting the use of AI in business processes
<https://tuatara.pl/blog/cognitive-technology/>

РЕЗУЛЬТАТИ ВИВЧЕННЯ ОСВІТНЬОЇ КОМПОНЕНТИ

Дисципліна забезпечує оволодіння здобувачами вищої освіти загальними та фаховими компетентностями і досягнення ними програмних результатів навчання:

ЗК01	Здатність до абстрактного мислення, аналізу та синтезу.
ЗК02	Здатність спілкуватися іноземною мовою як усно, так і письмово.
ЗК05	Здатність генерувати нові ідеї (креативність).
СК01	Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення.
СК02	Здатність розробляти і реалізовувати наукові та/або прикладні проекти у сфері інженерії програмного забезпечення.
СК03	Здатність проектувати архітектуру програмного забезпечення / програмного продукту, моделювати процеси функціонування окремих підсистем і модулів.
СК04	Здатність розвивати і реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення.
СК07	Здатність критично осмислювати проблеми у галузі інформаційних технологій та на межі галузей знань, інтегрувати відповідні знання та розв'язувати складні задачі у широких або мультидисциплінарних контекстах.
СК08	Здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення/програмного продукту на основі застосування сучасних моделей, методів та технологій

	розроблення програмного забезпечення / програмного продукту.
СК09	Здатність забезпечувати якість програмного забезпечення / програмного продукту.
РН01	Знати і застосовувати сучасні професійні стандарти і інші нормативно-правові документи з інженерії програмного забезпечення
РН08	Розробляти і модифікувати архітектуру програмного забезпечення для реалізації вимог замовника.
РН11	Забезпечувати якість на всіх стадіях життєвого циклу програмного забезпечення, у тому числі з використанням релевантних моделей та методів оцінювання, а також засобів автоматизованого тестування і верифікації програмного забезпечення.
РН14	Прогнозувати розвиток програмних систем та інформаційних технологій.
РН17	Збирати, аналізувати, оцінювати необхідну для розв'язання наукових і прикладних задач інформацію, використовуючи науково-технічну літературу, бази даних та інші джерела.

ОЦІНЮВАННЯ ЗНАНЬ ЗДОБУВАЧІВ ОСВІТИ

Сума балів, накопичених здобувачем вищої освіти за виконання всіх видів поточних навчальних завдань (робіт) на лабораторних/практичних заняттях, свідчить про ступінь оволодіння ним програмою освітньої компоненти на конкретному етапі її вивчення. Протягом семестру здобувачі освіти можуть набрати від 0 до 100 балів, що переводяться у національну шкалу оцінювання і відповідно у шкалу ЄКТС. Кількість балів відповідає певному рівню засвоєння дисципліни

Довідник з розподілу оцінок ДТЕУ (Шкала ЄКТС):

Бали ДТЕУ	Відсоток балів відносно загальної кількості одержаних прохідних балів	Кумулятивний відсоток отриманих прохідних балів
90-100	20	20
82-89	10	30
75-81	20	50
69-74	10	60
60-68	40	100

Роподіл балів за видами робіт:

Вид роботи	Бали	Вид роботи	Бали
Лабораторна робота 1	10	Самостійна робота 1	5
Лабораторна робота 2	10	Самостійна робота 2	5
Лабораторна робота 3	10	Самостійна робота 3	5
Лабораторна робота 4	10	Самостійна робота 4	5
Лабораторна робота 5	10		
Додаткові бали	20	Наукова робота	10

Вимоги до критеріїв оцінювання самостійної роботи студента (оцінювання одного завдання у відсотковому еквіваленті)

40%	Детальний розгляд сутності та вмісту основних джерел. Подання
-----	---

	фактів, ідей і результатів досліджень у логічній послідовності. Правильно проаналізовано поточний стан дослідження проблеми та зроблено огляд перспектив подальшого розвитку даного питання.
40%	Обґрунтованість аргументів, підтвердження особистого ставлення, пропозиції стосовно вирішення завдання, встановлення напрямків аналізу.
20%	Оформлення звіту у відповідності вимог
Критерії оцінювання самостійної роботи студента (оцінювання одного завдання у відсотковому еквіваленті)	
100%	В повному обсязі володіє навчальним матеріалом, вільно самостійно та аргументовано його викладає під час усних виступів та письмових відповідей, глибоко та всебічно розкриває зміст теоретичних питань та лабораторних завдань, використовуючи при цьому обов'язкову та додаткову літературу. Правильно вирішив усі тестові завдання.
80%	Достатньо повно володіє навчальним матеріалом, обґрунтовано його викладає під час усних виступів та письмових відповідей, в основному розкриває зміст теоретичних питань та лабораторних завдань, використовуючи при цьому обов'язкову літературу. Але при викладанні деяких питань не вистачає достатньої глибини та аргументації, допускаються при цьому окремі несуттєві неточності та незначні помилки. Правильно вирішив більшість тестових завдань
60%	В цілому володіє навчальним матеріалом викладає його основний зміст під час усних виступів та письмових відповідей, але без глибокого всебічного аналізу, обґрунтування та аргументації, без використання необхідної літератури допускаючи при цьому окремі суттєві неточності та помилки. Правильно вирішив половину тестових завдань.
40%	Не в повному обсязі володіє навчальним матеріалом. Фрагментарно, поверхово (без аргументації та обґрунтування) викладає його під час усних виступів та письмових відповідей, недостатньо розкриває зміст теоретичних питань та лабораторних завдань, допускаючи при цьому суттєві неточності, правильно вирішив меншість тестових завдань.
20%	Частково володіє навчальним матеріалом не в змозі викласти зміст більшості питань теми під час усних виступів та письмових відповідей, допускаючи при цьому суттєві помилки. Правильно вирішив окремі тестові завдання.
0%	Не володіє навчальним матеріалом та не в змозі його викласти, не розуміє змісту теоретичних питань та практичних завдань. Не вирішив жодного тестового завдання.
ОСНОВНІ ПОЛОЖЕННЯ, ЩО РЕГЛАМЕНТУЮТЬ ОСВІТНІЙ ПРОЦЕС	

діючі положення	https://knute.edu.ua/blog/read/?pid=44402
нормативно-правова база організації освітнього процесу	https://knute.edu.ua/blog/read/?pid=7330&uk
студенту	https://knute.edu.ua/#forstudent
НЕФОРМАЛЬНА ОСВІТА Рекомендовані сертифікаційні програми, курси, посібники користувача	
Intro to AI	https://365datascience.com/courses/machine-learning-deep-dive-business-applications-and-coding-walkthroughs/
Machine Learning Deep Dive: Business Applications and Coding Walkthroughs	https://365datascience.com/courses/intro-to-data-engineering/
The Machine Learning Algorithms A-Z	https://365datascience.com/courses/the-machine-learning-algorithms-a-z/
Machine Learning with Python	https://www.coursera.org/learn/machine-learning-with-python?msockid=2e692b416d486e6f1f35248b6c236f07
Machine Learning on AWS	https://www.udemy.com/course/amazon-web-services-machine-learning/learn/lecture/17244966#reviews
Microsoft Azure AI Fundamentals	https://learn.microsoft.com/en-us/training/courses/ai-900t00
Machine Learning on MS Azure	https://www.udemy.com/course/machine-learning-using-azureml/
ПОЛІТИКА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ:	
Відвідування лекційних та лабораторних занять: відвідування	Відвідування лекційних та лабораторних занять є обов'язковим. Допускаються пропуски занять з таких поважних причин, як хвороба (викладачу надається копія довідки від медичного закладу), участь в олімпіаді, творчому конкурсі тощо за попередньою домовленістю та згодою викладача за умови дозволу деканату (надаються документи чи інші матеріали, які підтверджують заявлену участь у діяльності студента).
Відпрацювання пропущених занять:	відпрацювання пропущених занять є обов'язковим незалежно від причини пропущеного заняття. Лекційне заняття має бути відпрацьоване до наступної лекції на консультації викладача з використанням ПЗ 365 Office Teams. Відпрацювання лекційного матеріалу передбачає вивчення пропущеного теоретичного матеріалу та складання тесту за цим матеріалом. Лабораторне заняття відпрацьовується під час консультації викладача (розклад консультацій на сайті).
Правила поведінки під час занять	обов'язковим є дотримання техніки безпеки в комп'ютерних лабораторіях. Студенти повинні приймати активну участь в обговоренні навчального матеріалу ознайомившись з ним напередодні (навчальний матеріал надається викладачем).

	<p>Мобільні пристрої дозволяється використовувати лише під час он-лайн тестування та підготовки практичних завдань в процесі заняття. Задля зручності, дозволяється використання ноутбуків та інших електронних пристроїв під час навчання в комп'ютерних аудиторіях (за взаємною згодою всіх учасників освітнього процесу)</p>
<p>Політика академічної доброчесності ДТЕУ</p>	<p>https://knute.edu.ua/blog/read/?pid=38987&uk</p>