



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФРАСТРУКТУРИ ТА ТЕХНОЛОГІЙ
ІНСТИТУТ УПРАВЛІННЯ, ТЕХНОЛОГІЙ ТА ПРАВА
ФАКУЛЬТЕТ УПРАВЛІННЯ І ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ



СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОНТОЛОГІЧНЕ МОДЕЛЮВАННЯ СКЛАДНИХ СИСТЕМ

Затверджено:

Протокол засідання кафедри
інформаційних технологій
№ 9 від 21 березня 2024 р.

Завідувач кафедри ІТ

Валерій ЗАВГОРОДНІЙ

Викладач	ТКАЧЕНКО Олександр Андрійович Кандидат фізико-математичних наук, доцент	
Посилання на профіль викладача на сайті ДУІТ	Ткаченко Олександр Андрійович	
E-mail	aatokg@gmail.com	
Факультет, Кафедра	Факультет Управління і технологій / Кафедра інформаційних технологій м. Київ, вул. Івана Огієнка, 19, каб. 601a	
Консультації	м. Київ, вул. Івана Огієнка, 19, каб. 601a	
Офіційна назва освітньої програми	Інженерія програмного забезпечення	
Рівень вищої освіти	перший (бакалаврський)	
Галузь знань, спеціальність	12 «Інформаційні технології» 121 «Інженерія програмного забезпечення»	
Статус дисципліни (обов'язкова, вибіркова)	Цикл дисциплін професійної підготовки, вибіркова	
Курс/ Семестр викладання	2/4	
Обсяг дисципліни	4 кредити ECTS / 120 загальна кількість годин	
Види та кількість аудиторних занять, денна/ заочна	Лекції – 20 годин / 4 години Практичні заняття – 24 години / 8 годин	
Форма контролю	Залік	
Локація та матеріально- технічне забезпечення	Аудиторія згідно з розкладом. Мультимедійний проектор, мережа Internet.	
Мова викладання	Українська	
Мета вивчення дисципліни	Формування теоретичних знань в області онтологічного моделювання об'єктів і процесів складних систем; розробки відповідного програмного забезпечення та практичних навичок використання сучасного програмного забезпечення при розв'язуванні задач онтологічного моделювання складних систем у різних предметних областях, тестування,	

	кодування та стандартизації такого програмного забезпечення і самостійної підготовки вищевказаних задач до розв'язування на ПК.
Загальні компетентності	ЗК 01. Здатність до абстрактного мислення, аналізу та синтезу. ЗК 02. Здатність застосовувати знання у практичних ситуаціях. ЗК 03. Здатність спілкуватися державною мовою як усно, так і письмово. ЗК 05. Здатність вчитися і оволодівати сучасними знаннями. ЗК 06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел
Спеціальні (фахові) компетентності	ФК 02. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування. ФК 03. Здатність розробляти архітектури, модулі та компоненти програмних систем. ФК 06. Здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки. ФК 07. Володіння знаннями про інформаційні моделі даних та системи, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних. ФК 08. Здатність застосовувати і розвивати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення. ФК 10. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя. ФК 11. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення. ФК 13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення. ФК 14. Здатність до алгоритмічного та логічного мислення.
Програмні результати навчання	ПР 01. Знати, аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки. ПР 03. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення. ПР 05. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізів та математичного моделювання для розробки програмного забезпечення. ПР 07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення. ПР 11. Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання. ПР 15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення. ПР 18. Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних. ПР 23. Вміти документувати та презентувати результати розробки програмного забезпечення.

ЧИМ ВАЖЛИВИЙ КУРС:

Курс дає можливість розширити світогляд та професійні компетенції, поглибивши теоретичні

знання та практичні вміння щодо:

Оснoв онтологічного моделювання складних систем.

Редактора онтологій Protégé та побудови за його допомогою онтологічної моделі складної системи.

Фреймворку Apache Jena для Java та його використання при створенні додатків Semantic Web з використанням онтологій.

Мови запитів SPARQL.

Відображення бази даних складної системи у онтологію.

Мов онтологічного моделювання складних систем.

Оцінки програмного забезпечення відповідного онтологічного моделювання складних систем.

ЗМІСТ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ ЗА ТЕМАМИ

Змістовий модуль 1. Основи онтологічного моделювання складних систем

Тема 1. Онтологічні моделі складних систем: основні поняття та визначення

Онтологія: основні поняття та визначення. Поняття онтологічної моделі. Онтологічна модель складної системи. Способи опису онтологічної моделі складної системи. Визначення моделі RDF та її розширення - моделі RDFS. Словники для відображення семантики в моделях RDF. Інтегрування RDF та HTML. Використання RDF-графів у семантичній мережі. Формати для запису та передачі RDF-графів: RDF/XML, RDF/JSON, N-Triples, Turtle, N3. Основні переваги використання моделі OWL. Діалекти OWL DL, OWL Lite та OWL Full.

Тема 2. Редактор онтологій Protégé

Редактори онтологій. Редактор онтологій Protégé. Створення нової онтології OWL у Protégé. Класи онтології та їх екземпляри (індивідуали). Визначені та примітивні класи. Об'єктні властивості та властивості даних, їх додаткові характеристики. Типи даних в онтологіях. Ієрархія класів та властивостей. Аксіоми – справедливі логічні твердження в онтології. Обмеження при визначенні онтологічних сутностей. Reasoner – система логічного виведення для отримання нових знань та перевірки онтології на наявність невідповідностей. Міграція даних з бази даних в онтологію.

Змістовий модуль 2. Програмні засоби онтологічного моделювання складних систем

Тема 3. Мови запитів для пошуку даних та отримання нових знань в онтології

Мова SPARQL: призначення, основні поняття та види запитів. Запити: SELECT, CONSTRUCT, ASK та DESCRIBE. Розділи запиту SELECT, його модифікатори та додаткові структури: FILTER, OPTIONAL, UNION, MINUS. Агрегатні функції SELECT та опції FILTER. Сутність автоматичного виведення нових знань з онтології. Призначення та синтаксис DL-запитів. Діалекти SWRL та SQWRL. Інструменти Protégé для роботи з DL, SWRL / SQWRL. Сховища триплетів (Graph Stores). Кінцеві точки - сервіси для отримання знань зі сховищ триплетів.

Тема 4. Фреймворк Apache Jena

Apache Jena – безкоштовний фреймворк для Java з відкритим кодом для створення додатків Semantic Web з використанням онтологій. Основні складові Apache Jena: Fuseki (автономний сервер), ARQ (засіб для виконання запитів SPARQL з командного рядка), TDB (сховище RDF-триплетів), повнотекстова пошукова система. Створення Jena-застосунків: створення та обробка rdf-моделей, виконання запитів, використання ризонерів. Основні відмінності між Jena та OWL-API.

Практичні заняття курсу передбачають виконання ситуаційних, тестових, розрахункових та інших завдань, опитування та дискусії за темами, короткі виступи та презентації з тематики дисципліни.

Тематика практичних занять:

1. Використання моделі RDF для представлення даних.

2. Інтеграція RDF та HTML
3. Створення онтологічної моделі предметної області в редакторі Protégé
4. Ознайомлення з мовою запитів SPARQL
5. Розробка SPARQL-запитів для пошуку даних та знань в онтологічних моделях RDF/RDFS та OWL
6. Створення програм з використанням Apache Jena.
7. Apache Jena як API для семантичних вебдодатків Java
8. Створення програми на мові Java для обробки даних онтології
9. OBDA: Відображення бази даних у owl-онтологію
10. Мови DL та SWRL/SQWRL

ОЦІНЮВАННЯ

Форми поточного та підсумкового контролю	Поточний контроль – 100 балів Підсумковий контроль – залік
КРИТЕРІЇ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ	
Підсумкові бали з навчальної дисципліни визначаються як сума балів, отриманих здобувачем протягом семестру.	

Оцінювання навчальних досягнень студентів за усіма видами навчальних робіт проводиться за *поточним* та *підсумковим* контролюми. Поточний контроль знань студентів з навчальної дисципліни проводиться у письмовій формі. Контрольні завдання за змістовим модулем включають теоретичні та тестові питання. Контроль самостійної роботи проводиться:

з лекційного матеріалу – шляхом перевірки конспектів;

з практичних робіт – за допомогою перевірки розв’язків задач, отриманих за допомогою ПК і відповідного програмного забезпечення, та усного контролю.

Усі контрольні заходи включено до 100-бальної шкали оцінювання.

Поточне тестування та самостійна робота				Сума
Змістовий модуль №1		Змістовий модуль № 2		
T1	T2	T3	T4	
25	25	25	25	100

T1, T2, T3, T4 – теми змістових модулів.

Додаткові бали до поточного контролю здобувач освіти може отримати, пройшовши навчальний курс у вигляді неформальної освіти з отриманням сертифікату в межах предмету вивчення дисципліни та пройшовши процедуру визнання згідно Положення про визнання результатів навчання, отриманих у неформальній освіті здобувачами вищої освіти ДУІТ

ШКАЛА ОЦІНЮВАННЯ: НАЦІОНАЛЬНА ТА ECTS			
Оцінка в балах	Оцінка за національною шкалою	Оцінка за шкалою ECTS	
		Оцінка	Пояснення
90-100	Відмінно («зараховано»)	A	«Відмінно» - теоретичний зміст курсу освоєний цілком, необхідні практичні навички роботи з освоєним матеріалом сформовані, всі навчальні завдання, які передбачені програмою навчання виконанні в повному обсязі, відмінна робота без помилок або з однією незначною помилкою.
82-89	Добре («зараховано»)	B	«Дуже добре» - теоретичний зміст курсу освоєний цілком, необхідні практичні навички роботи з освоєним матеріалом в основному сформовані, всі навчальні завдання, які передбачені програмою навчання виконанні, якість виконання більшості з них оцінено числом балів, близьким до максимального, робота з двома – трьома незначними помилками
75-81		C	«Добре» - теоретичний зміст курсу освоєний цілком, практичні навички роботи з освоєним матеріалом в основному сформовані, всі навчальні

			завдання, які передбачені програмою навчання виконанні, якість виконання жодного з них не оцінено мінімальним числом балів, деякі види завдань виконані з помилками, робота з декількома незначними помилками, або з однією – двома значними помилками
64-74	Задовільно («зараховано»)	D	«Задовільно» - теоретичний зміст курсу освоєний не повністю, але прогалини не носять істотного характеру, необхідні практичні навички роботи з освоєним матеріалом в основному сформовані, більшість передбачених програмою навчання навчальних завдань виконано, деякі з виконаних завдань, містять помилки, робота з трьома значними помилками
60-63		E	«Достатньо» - теоретичний зміст курсу освоєний частково, деякі практичні навички роботи не сформовані, частина передбачених програмою навчання навчальних завдань не виконані, або якість виконання деяких з них оцінено числом балів, близьким до мінімального, робота, що задовольняє мінімум критеріїв оцінки
35-59	Незадовільно («не зараховано»)	FX	«Умовно незадовільно» теоретичний зміст курсу освоєний частково, необхідні практичні навички роботи не сформовані, більшість передбачених програм навчання, навчальних завдань не виконано, або якість їхнього виконання оцінено числом балів, близьким до мінімального; при додатковій самостійній роботі над матеріалом курсу можливе підвищення якості виконання навчальних завдань (з можливістю повторного складання), робота що потребує доробки
1-34		F	«Безумовно незадовільно» теоретичний зміст курсу не освоєно, необхідні практичні навички роботи не сформовані, всі виконані навчальні завдання містять грубі помилки, додаткова самостійна робота над матеріалом курсу не приведе до значимого підвищення якості виконання навчальних завдань, робота, що потребує повної переробки

ІНФОРМАЦІЙНО-МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Методичне забезпечення:

1. Робоча програма навчальної дисципліни.
2. Конспект лекцій.

Електронні ресурси бібліотеки ДУІТ: <https://library.duit.in.ua>.

Список рекомендованої літератури

Базова (основна):

- 1 Ткаченко О.А., Ткаченко О.І., Овчарук І.В. Сучасні парадигми програмування. Ч.2: навч. посіб. Київ: Вид-во КНУКіМ, 2017. 308 с.
- 2 Онтологічне моделювання складних систем: методичні матеріали для студентів спеціальності 121 «Інженерія програмного забезпечення» та 122 «Комп'ютерні науки». Київ: Вид-во ДУІТ, 2021. 34 с.
- 3 Ткаченко О.І., Ткаченко О.А., Ткаченко К.О. Онтологічне моделювання ситуаційного управління. *Цифрова платформа: інформаційні технології в соціокультурній сфері*, 2020. Т. 3. № 1. С. 22-32.
- 4 Tkachenko K.O. Using Ontological Modeling by Intellectualization of Learning Processes, *Digital platform: information technology in the sociocultural area*, 2022. Vol. 5. № 2. pp. 261-270.
- 5 Ткаченко О.А., Ткаченко О.І. Деякі аспекти ситуаційно-семантичного моделювання складних об'єктів, процесів та систем. *Водний транспорт*, 2017. Вип. № 1 (26). С.129-133.
- 6 Ontolingua. [Електронний ресурс]. URL: <http://www.ksl.stanford.edu/software/ontolingua/>
- 7 Хоменко Л.М. Основи проектування і моделювання: навч.-метод. посіб. Умань: ФОП Жовтий О.О., 2016. 125 с.
- 8 Виклюк Я.І., Камінський Р.М., В Пасічник.В. Моделювання складних систем: посібник. Львів: Вид-во «Новий Світ–2000», 2020. 404 с.

- 9 Овдій О.М., Проскудіна Г.Ю. Онтології у контексті інтеграції інформації: представлення, методи та інструменти побудови. [Електронний ресурс]. URL: <http://dspace.nbu.gov.ua/bitstream/handle/123456789/1683/48%20-%20Ovdiy.pdf?sequence=1>.
- 10 Козак І.А. Інфраструктура онтологічного моделювання інформаційних систем віртуальних організацій. [Електронний ресурс]. URL: <https://core.ac.uk/download/pdf/300239563.pdf>.
- 11 Бойко Н.І. Методи та інструменти моделювання інформаційних процесів. [Електронний ресурс]. URL: http://science.lp.edu.ua/sites/default/files/Papers/40_81.pdf
- 12 Піднебесна Г.А. Конструювання комплексу інструментальних засобів індуктивного моделювання з використанням онтологій. [Електронний ресурс]. URL: <http://usim.org.ua/arch/2020/6/7.pdf>.
- 13 Мейтус В.Ю. Інтелектуальні системи, онтології та онтологічні простори. [Електронний ресурс]. URL: <https://core.ac.uk/download/pdf/149242313.pdf>
- 14 Буров Є.В., Пасічник В.В. Програмні системи на базі онтологічних моделей задач. [Електронний ресурс]. URL: <https://science.lpnu.ua/sites/default/files/journalpaper/2018/jun/12873/4burovievpasichnikvvp.pdf>.
- 15 Рогушина Ю.В. Використання онтологічних знань інтелектуальними агентами. [Електронний ресурс]. URL: <http://dspace.nbu.gov.ua/bitstream/handle/123456789/144478/07-Rogushina.pdf?sequence=1>
- 16 Патракеєв І., Денисюк Б. Моделювання баз знань в редакторі онтологій PROTÉGÉ. [Електронний ресурс]. URL: <http://mtp.knuba.edu.ua/article/view/220746>
- 17 Rogushina J.V., Grishanova I.J. Ontological methods and tools for semantic extension of the media WIKI technology. *Проблеми програмування*, 2020. № 2-3. С. 61-73.
- 18 Keet C.M. An Introduction to Ontology Engineering, 2020. 289 p.
- 19 Euzenat J., Shvaiko P. Ontology Matching. Springer, 2016. 511 p.
- 20 Jakus G., Milutinovich V., Omerovich S., Tomazich S. Concepts, Ontologies, and Knowledge Representation. Springer, 2016. 67 p.
- 21 Bjorner D. Domain analysis and description principles, techniques and modelling languages. ACM Transactions on Software. *Engineering and Methodology* (TOSEM), 2019. Vol.28(2). pp.1-67.
- 22 Rabiser R., Schmid K., Eichelberger H., Vierhauser M., Guinea S., Grünbacher P. A domain analysis of resource and requirements monitoring: Towards a comprehensive model of the software monitoring domain. *Information and Software Technology*, 2019. Vol. 111. P. 86–109.
- 23 Chebanyuk O.V., Palahin O.V., Markov K.K. Domain engineering approach of software requirements analysis. *Проблеми програмування*, 2020. № 2-3. С. 164-172.

Додаткова інформація

Детальнішу інформацію щодо методів навчання, форм оцінювання, самостійної роботи та повного списку літератури наведено у Робочій програмі навчальної дисципліни.

ПОЛІТИКА ДИСЦИПЛІНИ

Щодо академічної доброчесності

Дотримання академічної доброчесності засновується на ряді положень та принципів академічної доброчесності, що регламентують діяльність здобувачів вищої освіти та викладачів ДУІТ:

Кодекс академічної доброчесності Державного університету інфраструктури та технологій

Положення про систему забезпечення академічної доброчесності у Державному університеті та технологій

Положення про Комісію з академічної доброчесності у ДУІТ та Комісію з етики та управління конфліктами у сфері академічної доброчесності у ДУІТ

Порушення Кодексу академічної доброчесності ДУІТ є серйозним порушенням, навіть якщо воно є ненавмисним.

Списування під час контрольних заходів заборонені.

Усі письмові роботи, виконані в електронному вигляді (реферати), перевіряються на наявність плагіату згідно з Положенням про порядок перевірки навчальних, кваліфікаційних, науково-методичних наукових та інших робіт на наявність ознак академічного плагіату у ДУІТ. У випадках виявлення порушення – реагування відповідно до Кодексу академічної доброчесності ДУІТ.

Щодо відвідування

Відвідування занять є обов'язковим компонентом оцінювання, за яке нараховуються бали. За об'єктивних причин (хвороба, міжнародне стажування, індивідуальний графік) навчання може відбуватися в онлайн (або змішаній) формі за погодженням із деканом факультету.

Неформальна освіта

Можливість зарахування результатів неформальної освіти регламентується «Положенням про визнання результатів навчання, отриманих у неформальній освіті здобувачами вищої освіти ДУІТ».

Укладач



Олександр ТКАЧЕНКО