



Штучний інтелект. Курсова робота

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	F Інформаційні технології
Спеціальність	F1 Прикладна математика
Освітня програма	Наука про дані та математичне моделювання
Статус дисципліни	Нормативна
Форма навчання	очна(денна)
Рік підготовки, семестр	2 курс, весняний семестр
Обсяг дисципліни	1 кредити ECTS / 30 год.
Семестровий контроль/ контрольні заходи	Залік
Розклад занять	Лекції та практики не передбачені. Заліки — 2 год наприкінці семестру (1 заняття, під час 16 тижня)
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Керівники: КМ-51, КМ-52: д.ф., доцент Жук Іван Сергійович, i.zhuk-2023@kpi.ua КМ-53: асистент Щоголев Максим Олександрович, shchogolev.maksym@gmail.com
Розміщення курсу	Classroom: https://classroom.google.com/c/ODE2MTg3NTQ0OTc2?cjc=r6buu56h Сайт кафедри: https://pma.fpm.kpi.ua/uk/studentam/silabusi-distiplin

Програма навчальної дисципліни

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

Метою дисципліни є **систематизація та поглиблення** теоретичних знань та практичних навичок, отриманих у рамках лекційного курсу "Штучний інтелект", шляхом виконання **індивідуального комплексного програмного проєкту**. Курс фокусується на переході від вирішення окремих навчальних завдань (як на практичних заняттях) до розробки завершеного програмного рішення (агента), здатного вирішувати складнішу, наближену до реальної, задачу.

Предметом навчальної дисципліни є **теоретичні основи, математичні моделі та практичні методи** побудови інтелектуальних систем, що розглядаються крізь призму **раціонального агента**.

Після засвоєння навчальної дисципліни студенти мають продемонструвати такі: фахові компетентності:

- ФК14. Здатність сформулювати математичну постановку задачі, спираючись на постановку мовою предметної галузі, та обирати метод її розв'язання, що забезпечує потрібні точність і надійність результату;
- ФК18. Навички розв'язування специфічних математичних та комп'ютерних задач штучного інтелекту, машинного та глибинного навчання

програмні результати навчання:

PH08. Поєднувати методи математичного та комп'ютерного моделювання з неформальними процедурами експертного аналізу для пошуку оптимальних рішень;

PH10. Володіти методиками вибору раціональних методів та алгоритмів розв'язання математичних задач оптимізації, дослідження операцій, оптимального керування і прийняття рішень, аналізу даних;

PH15. Уміти організувати власну діяльність та одержувати результат у рамках обмеженого часу.

PH21. Вміти формулювати та розв'язувати базові задачі штучного інтелекту, включаючи машинне та глибоке навчання;

уміння та навички:

- Проєктувати та реалізовувати програмного агента та середовище для нього, застосовуючи концепцію PEAS та навички ООП.

- Адаптувати та застосовувати класичні алгоритми пошуку (BFS, A*) або змагального пошуку (Minimax) для вирішення задач, що є складнішими за ті, що розглядались на практичних заняттях.

- Розв'язувати прикладні задачі задоволення обмежень (CSP), реалізувавши відповідний алгоритм (напр., backtracking).

- Створювати завершений програмний конвеєр (pipeline) для базових задач NLP або CV, що включає завантаження та обробку даних (pandas), виділення ознак (напр., TF-IDF , HOG) та навчання моделі (scikit-learn).

- Проводити експериментальний аналіз реалізованих алгоритмів, обґрунтовано обираючи метрики ефективності (напр., точність, час, повнота).

- Структурувати технічний звіт та захищати власні проєктні рішення.

2. Пререквізити та постреквізити дисципліни

3.1. Пререквізити (Передумови вивчення)

Для успішного виконання курсового проєкту студентам необхідні базові навички програмування, отримані з раніше вивчених дисциплін:

- **Програмування на Python.**
- **Об'єктно-орієнтоване програмування** (для проєктування архітектури агентів та середовищ).
- **Алгоритми та структури даних** (для ефективної реалізації алгоритмів пошуку).

3.2. Кореквізити (Обов'язкова супутня дисципліна)

Ця дисципліна ("Курсова робота") є практичним доповненням і не може вивчатися окремо від теоретичної бази: **"Штучний інтелект" (4 кредити ECTS)**, вивчається паралельно. Дисципліна "Штучний інтелект" закладає всю необхідну концептуальну базу (агенти, пошук, рішення) та надає первинні практичні навички роботи з бібліотеками (numpy, pandas, scikit-learn), які студент повинен синтезувати та застосувати в рамках курсового проєкту.

3.3. Постреквізити (Місце курсу у подальшому навчанні)

Успішне виконання комплексного проєкту в рамках цієї дисципліни формує у студента **цілісне практичне бачення** процесу розробки інтелектуальних систем.

Це забезпечує практичну готовність та створює проєктне портфоліо для поглибленого вивчення наступних дисциплін:

- Аналіз даних (3-й курс)
- Основи машинного навчання (3-й курс)
- Глибинне навчання (4-й курс)
- Навчання з підкріпленням (4-й курс)

3. Зміст навчальної дисципліни

Зміст дисципліни визначається логікою проєктної діяльності та охоплює повний цикл розробки програмного рішення задачі штучного інтелекту. Робота над курсовою роботою складається з наступних змістових модулів:

Змістовий модуль 1. Аналіз та постановка задачі

- Вибір та обґрунтування теми курсової роботи.
- Детальне формулювання прикладної задачі мовою штучного інтелекту (визначення мети, простору станів, дій, метрик успішності).
- Формалізація задачі з використанням концепцій, отриманих з лекційного курсу (напр., PEAS-опис агента, формулювання CSP, визначення функції корисності тощо).

Змістовий модуль 2. Дослідження та проєктування

- Проведення аналітичного огляду 2-3 релевантних джерел та існуючих підходів до вирішення задачі.
- Проєктування програмної архітектури рішення (опис основних класів, модулів, структур даних, взаємодії між ними).

Змістовий модуль 3. Програмна реалізація та тестування

- Створення програмного середовища (симулятора) для агента.
- Реалізація ключового(их) алгоритму(ів) ШІ, обраного(их) для вирішення задачі (напр., A*, Minimax з оптимізаціями, розв'язувач CSP, конвеєр scikit-learn).
- Розробка набору тестів (unit-тести або функціональні сценарії) для перевірки коректності роботи реалізованих алгоритмів.

Змістовий модуль 4. Проведення експериментів та аналіз результатів

- Розробка методики експериментального дослідження (визначення вхідних даних, параметрів, що варіюються, та метрик для оцінки).
- Проведення серії обчислювальних експериментів (напр., порівняння евристик, оцінка точності моделей, вимірювання часу роботи).
- Аналіз отриманих результатів, їх візуалізація (графіки, таблиці) та формулювання обґрунтованих висновків.

Змістовий модуль 5. Оформлення та захист проєкту

- Підготовка технічного звіту (пояснювальної записки) відповідно до встановлених вимог.
- Підготовка презентаційних матеріалів та демонстраційної версії проєкту для публічного захисту.

4. Навчальні матеріали та ресурси

5.1. Основна література

1. **Конспекти лекцій та матеріали практичних занять** з кореквізиту "Штучний інтелект".
2. Russell, S. J., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.

5.2. Додаткова література

1. Poole, D. L., & Mackworth, A. K. (2017). *Artificial Intelligence: Foundations of Computational Agents* (2nd ed.). Cambridge University Press.
2. Müller, A. C., & Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media.

5.3. Інформаційні ресурси та програмне забезпечення

1. **Лекційні матеріали:** Конспекти лекцій, презентації та приклади коду, надані викладачем (наприклад, у Moodle).
2. **Python Google Colab (Рекомендовано для практичних робіт):** Хмарне середовище (на базі Jupyter), яке не вимагає налаштування та містить всі необхідні бібліотеки (numpy, pandas, sklearn). <https://colab.research.google.com/>
3. **Python (Anaconda Distribution):** Для локального виконання робіт рекомендується встановити дистрибутив **Anaconda** або **Miniconda** для легкого керування бібліотеками. <https://www.anaconda.com/>
4. **Середовище розробки: Jupyter Notebook** (входить до Anaconda) або **Visual Studio Code** з розширеннями Python.
5. **Офіційна документація (обов'язкова для практичних робіт):**
 - **Numpy:** <https://numpy.org/doc/>
 - **Pandas:** <https://pandas.pydata.org/docs/>
 - **Scikit-learn:** <https://scikit-learn.org/stable/documentation.html>
 - **Scikit-image:** <https://scikit-image.org/docs/stable/>
6. **Stanford HAI (AI Ethics):** Ресурси Стенфордського інституту ШІ, орієнтованого на людину. <https://hai.stanford.edu/>

5.4. Ресурси з підготовки звіту та презентації

1. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання: https://kubq.edu.ua/images/stories/podii/2017/06_21_posylannia/dstu_8302.pdf
2. ДСТУ 3008:2015. Звіти у сфері науки і техніки. Структура та правила оформлювання: https://science.kname.edu.ua/images/dok/derzhstandart_3008_2015.pdf
3. **Вимоги до оформлення документації:** <https://pma.fpm.kpi.ua/uk/studentam/ofitsiyno/vimoqi-do-oformlennya-dokumentatsiyi>
4. Використання LaTeX: <https://pma.fpm.kpi.ua/uk/studentam/ofitsiyno/vikoristannya-la-te-x>
5. **Ресурси з технічного письма (Technical Writing):**
 - Google Technical Writing Courses: <https://developers.google.com/tech-writing>
 - Google developer documentation style guide: <https://developers.google.com/style/>
6. **Шаблон презентацій КПІ:** <https://brand.kpi.ua/download/>

5. Методика опанування та ключові етапи

Методика опанування дисципліни базується на самостійній проектній роботі з регулярними консультаціями з керівником курсової роботи (викладачем, закріпленим за групою).

Робота над проектом чітко структурована і вимагає дотримання 5 ключових етапів (контрольних точок) протягом семестру. Своєчасне проходження цих етапів є обов'язковим і формує компонент "Дотримання етапів (Workflow)" (10 балів) у фінальній оцінці.

Етап 1: Формулювання задачі (Орієнтовний дедлайн: Тиждень 4) Студент має обрати тему (із запропонованих або власну), узгодити її з керівником, проаналізувати задачу та визначити чіткі мету та очікувані результати.

Етап 2: Огляд та Архітектура (Орієнтовний дедлайн: Тиждень 7) Студент проводить огляд 2-3 схожих рішень або релевантних джерел. На основі аналізу розробляється та узгоджується з керівником архітектура проекту (опис класів, модулів, форматів даних або структури дослідницького звіту).

Етап 3: "Сира" реалізація / MVP (Орієнтовний дедлайн: Тиждень 10) На цьому етапі студент має надати першу робочу версію проекту ("Minimum Viable Product"). Ключовий ШІ-алгоритм має бути реалізований (напр., A^* знаходить шлях, Minimax робить ходи, але ще без оптимізацій; або, якщо для дослідження – то це, наприклад, зібрання даних завершено та проведено початковий розвідковий аналіз (EDA)).

Етап 4: Експерименти та фіналізація (Орієнтовний дедлайн: Тиждень 13) Студент реалізовує повний функціонал (напр., оптимізації, порівняння евристик, аналіз точності моделей). Проводиться серія експериментів. На основі цього готується фінальний код (або Colab-ноутбук) та **перший драфт пояснювальної записки (звіту)**, який надається керівнику на перевірку.

Етап 5: Захист проекту (Орієнтовний дедлайн: Тиждень 15-16) Підготовка фінальної презентації. Захист курсової роботи (усна доповідь, демонстрація роботи, відповіді на запитання), як зазначено в розділі 8.3

6. Орієнтовний перелік тем курсової роботи

Категорія 1. Методи пошуку та CSP

1. **Розробка навігаційного агента (A^*):** Реалізація алгоритму A^* для пошуку оптимального шляху на реальній або змодельованій карті міста. Порівняння ефективності 2-3 різних евристик.
2. **Розв'язувач логічних ігор (CSP):** Створення програми для вирішення Судоку або N-Queens з використанням *backtracking* та дослідженням впливу евристик (MRV, LCV) на швидкість пошуку.

3. **Автоматичний складач розкладу (CSP):** Реалізація CSP-розв'язувача для складання простого розкладу занять з урахуванням обмежень (аудиторії, викладачі).

Категорія 2. Змагальний пошук

4. **ШІ для настільної гри (Minimax):** Реалізація гри "Connect Four" або "Reversi" та ШІ-агента на основі Minimax з альфа-бета відсіканням та розробкою власної функції оцінки.
5. **Дослідження евристик у змагальному пошуку:** На базі реалізованої гри (або "Шашок"), дослідити, як 2-3 різні функції оцінки (evaluation function) впливають на "силу" агента.

Категорія 3. Логіка, планування та рішення

6. **Симулятор планувальника (STRIPS):** Реалізація простого планувальника для "Світу Блоків" (Blocks World), що приймає опис світу, початковий стан та ціль.
7. **Агент, що приймає рішення (Utility Theory):** Моделювання агента (напр., інвестор), що приймає рішення в умовах невизначеності, використовуючи pandas для розрахунку очікуваної корисності (Expected Utility).

Категорія 4. Базові NLP та CV (з scikit-learn)

8. **Порівняльний аналіз класифікаторів тексту (NLP):** Вирішення задачі аналізу тональності для набору відгуків. Порівняння 3 моделей scikit-learn (Naive Bayes, Logistic Regression, SVM) на ознаках TF-IDF.
9. **Класифікація зображень (CV):** Класифікація набору даних (напр., Fashion-MNIST). Порівняння точності класифікаторів (k-NN, SVM, MLPClassifier) при використанні "сирих" пікселів проти HOG-ознак.

Категорія 5. Дослідницько-аналітичні проекти

10. **(Дослідження A*)** Порівняльний аналіз ефективності 4-5 різних евристик (допустимих і недопустимих) для задачі "П'ятнашки". (Артефакт: звіт + Colab-ноутбук з експериментами).
11. **(Етика ШІ)** Аналіз 1-2 відомих випадків упередженості (bias) в реальних системах ШІ (напр., у кредитному скорингу чи розпізнаванні облич). (Артефакт: аналітичний звіт).
12. **(Представлення знань)** Порівняльний огляд методів представлення знань (логіка предикатів, семантичні мережі, фрейми) з прикладами задач для кожного. (Артефакт: аналітичний звіт).
13. **(Застосування CSP)** Аналітичний огляд застосування методів CSP у 2-3 реальних областях (напр., логістика, біоінформатика). (Артефакт: оглядовий звіт).
14. **(Дослідження NLP)** Порівняльний аналіз методів представлення тексту: "Bag-of-Words" проти TF-IDF проти простих Word Embeddings (Word2Vec) для задачі класифікації. (Артефакт: звіт + Colab-ноутбук).
15. **(Дослідження CV)** Огляд та класифікація класичних дескрипторів ознак у CV (HOG, SIFT, SURF) та їхніх ключових застосувань. (Артефакт: оглядовий звіт).

7. Самостійна робота здобувача вищої освіти

№	Вид самостійної роботи	Орієнтовні години
1.	Аналіз проблеми та огляд літератури (Вибір теми, формулювання задачі, пошук та аналіз 2-3 джерел)	5 год.
2.	Проектування архітектури (Планування структури коду, класів, модулів, або структури дослідницького звіту)	2 год.
3.	Основна програмна реалізація / Проведення дослідження (Написання коду (MVP) / збір даних та основний аналіз)	10 год.
4.	Тестування, експерименти та аналіз результатів (Запуск тестів, порівняння метрик, генерація графіків, формулювання висновків)	5 год.
5.	Написання пояснювальної записки (звіту) (Структурування, написання та форматування тексту роботи згідно зі стандартами)	5 год.
6.	Підготовка до захисту (Створення презентації, підготовка демонстрації та доповіді)	2 год.
	РАЗОМ	30 год.

Політика та контроль

8. Політика освітнього компонента

Академічна доброчесність: Робота над курсовим проектом є **індивідуальною**. Плагіат, копіювання коду або тексту звіту в інших студентів (включаючи попередні роки) або з нецитованих джерел суворо заборонено. У разі виявлення плагіату робота автоматично отримує 0 балів без права перескладання.

Використання зовнішніх ресурсів: Дозволяється використовувати зовнішні бібліотеки (scikit-learn, pandas тощо). Однак, ключова логіка (напр., реалізація алгоритму A*, евристичної функції Minimax, конвеєра ML) має бути розроблена студентом особисто.

Використання генеративного ШІ: Дозволяється використовувати генеративні моделі ШІ (напр., GitHub Copilot, ChatGPT) як допоміжний інструмент. Проте, студент несе повну відповідальність за наданий код та текст звіту і має бути готовим детально пояснити будь-яку його частину під час захисту.

Дедлайни та Workflow: Курсова робота має загальний обсяг 30 годин (1 ECTS). Робота оцінюється кумулятивно. Дотримання проміжних дедлайнів (етапів), узгоджених з керівником, є обов'язковим та враховується у компоненті "Дотримання етапів (Workflow)".

Комунікація: Основна комунікація (консультації, затвердження теми, звітність за етапами) ведеться з викладачем, який закріплений за відповідною академічною групою.

9. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Формою підсумкового семестрового контролю з дисципліни є **залік**.

Залік виставляється на основі **рейтингового балу (PCO)**, який студент накопичує протягом семестру. Максимальний рейтинг складає **100 балів**.

9.1. Компоненти рейтингу

Рейтинг студента формується з 4 компонентів:

- **Дотримання етапів (Workflow) – 10 балів.** Оцінюється своєчасне виконання та звітність за ключовими етапами (напр., Етап 1: Затвердження теми, Етап 3: MVP, Етап 4: Драфт звіту). Також враховується якість наданих проміжних матеріалів та дотримання дедлайнів, погоджених з керівником.
- **Пояснювальна записка (Звіт) – 25 балів.** Оцінюється повнота розкриття теми, логіка викладу, глибина аналізу, якість проведених експериментів, обґрунтованість висновків, а також грамотність та відповідність стандартам оформлення. Цей компонент є ключовим, особливо для проєктів дослідницького типу.
- **Практичний артефакт – 25 балів.** Оцінюється конкретний результат роботи. Детальні критерії для програмних та дослідницьких проєктів наведені у п. 8.2.
- **Захист та Демонстрація – 40 балів.** Оцінюється якість фінальної презентації, чіткість доповіді, "жива" демонстрація робочого проєкту (або результатів дослідження) та глибина відповідей на запитання.

9.2. Оцінювання компоненту "Практичний артефакт" (25 балів)

- **Для програмних проєктів:** Оцінюється **програмний код**. Критерії: коректність роботи алгоритмів, ефективність, структурованість та чистота коду (напр., застосування ООП), наявність коментарів.
- **Для дослідницьких проєктів:** Оцінюється **дослідницький супровід**. Критерії: глибина огляду джерел, коректність аналізу, відтворюваність експериментів (напр., у Google Colab), якість візуалізації даних, обґрунтованість запропонованої класифікації.

9.3. Організація захисту

Захист курсових робіт проводиться в останні тижні семестру. Кожен студент захищає роботу перед викладачем, закріпленим за його групою. Регламент захисту: **до 20 хвилин на доповідь та демонстрацію + 10 хвилин на запитання**.

9.4. Умови допуску до заліку та шкала оцінювання

Необхідною умовою допуску до заліку є семестровий рейтинг не менше **60 балів**.

При семестровому рейтингу (сума балів за 4 компоненти) **не менше 60 балів**, студент отримує залік "автоматично" відповідно до таблиці:

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

Офіційний залік відбувається під час 16 тижня семестру. У разі якщо на момент офіційного проведення заліку студент не виконує умови допуску до заліку, у день проведення заліку студент складає залік.

Політика перескладання (отримання заліку при < 60 балів):

Політика перескладання

- Студенти, які набрали менше 60 балів, **зобов'язані** складати залік.
- Студенти, які набрали 60 і більше балів, але **бажають підвищити** свою оцінку, також можуть складати залік.
- **Увага:** При складанні заліку всі бали, набрані студентом у семестрі, **анулюються**. Залік складається "з нуля" та повністю визначає фінальну оцінку за шкалою (0-100).
- Формат заліку (перескладання) – повторний захист проекту, більш детальний захист, тобто захист з більшою кількістю питань як по презентації, так і за пояснювальною запискою та програмним кодом.

Робочу програму навчальної дисципліни (силабус):

Складено доктором філософії, доцентом кафедри ПМА Жуком Іваном Сергійовичем

Ухвалено кафедрою прикладної математики (протокол №14 від 13.05.2026)

Погоджено Методичною комісією ФПСМ (протокол №11 від 22.05.2026)