



МАШИННЕ НАВЧАННЯ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці</i>
Статус дисципліни	<i>Вибіркова</i>
Форма навчання	<i>Очна (денна)</i>
Рік підготовки, семестр	<i>4-й курс, 1-й семестр</i>
Обсяг дисципліни	<i>4 кредити /120 год. (лекцій 36 год., практик 18 год., СРС 66 год.)</i>
Семестровий контроль/ контрольні заходи	<i>Екзамен, модульна контрольна робота</i>
Розклад занять	<i>http://schedule.kpi.ua/</i>
Мова викладання	<i>Українська/Англійська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: д. е. н., професор Сігайов Андрій Олександрович Практичні / Семінарські: Лабораторні: Сігайов А. О.</i>
Розміщення курсу	<i>https://course.fast.ai</i>

Програма навчальної дисципліни

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

Чому майбутньому фахівцю варто вчити саме цю дисципліну?

Машинне навчання — це метод аналізу даних, який автоматизує побудову аналітичної моделі. Це галузь штучного інтелекту, заснована на ідеї, що системи можуть вчитися на основі даних, виявляти закономірності та приймати рішення з мінімальним втручанням людини.

Завдяки новим обчислювальним технологіям сьогодні машинне навчання не схоже на машинне навчання минулого. Воно народилося завдяки розпізнаванню образів та теорії, згідно з якою комп'ютери можуть вчитися, не будучи запрограмованими на виконання конкретних завдань; дослідники, які цікавляться штучним інтелектом, хотіли дізнатись, чи можуть комп'ютери вчитися на даних. Ітераційний аспект машинного навчання важливий, оскільки, моделі можуть самостійно адаптуватися до нових даних. Вони вчать на попередніх обчисленнях для отримання надійних, повторюваних рішень та результатів. Це наука не нова, але така, що набрала нових обертів.

Хоча багато алгоритмів машинного навчання існують вже давно, можливість автоматичного застосування складних математичних обчислень до великих даних — знову і знову, все швидше і швидше — є недавньою розробкою. Ось кілька широко розрекламованих прикладів програм машинного навчання, які вам можуть бути знайомі:

- Сильно розпіарений самокерований автомобіль Google? Сутність машинного навчання.*

- *Онлайн-рекомендації, такі як пропозиції від Amazon, Netflix, Spotify та YouTube? Програми машинного навчання для повсякденного життя.*
- *Знаєте, що клієнти кажуть про вас у Twitter? Машинне навчання в поєднанні зі створенням лінгвістичних правил.*
- *Виявлення шахрайства? Одне з найбільш очевидних та важливих застосувань у нашому світі сьогодні.*

Хто цим користується? Більшість галузей, що працюють з великими обсягами даних, визнали цінність технології машинного навчання. Отримуючи інсайти з цих даних — часто в реальному часі — організації можуть працювати ефективніше або отримати перевагу над конкурентами.

Серед причин вивчення машинного навчання важливою, на наш погляд, є величезні зарплати та премії, які можуть отримувати експерти зі штучного інтелекту. Майже всі великі технологічні компанії мають якийсь проєкт штучного інтелекту, і вони готові платити експертам мільйони доларів, щоб допомогти виконати його. Хоча пул доступних фахівців-дослідників з машинного навчання зростає, це відбувається недостатньо швидко. На цей час попит на кваліфіковану робочу силу в галузі сильно перевищує пропозицію, зарплати стрімко зростають, і ця ситуація швидко не зміниться.

Мета дисципліни. *Ознайомити студентів з сучасним станом технологій машинного навчання.*

Предмет дисципліни. *Огляд машинного навчання, включаючи навчання зі вчителем та без вчителя, глибоке навчання, задачі комп'ютерного зору, колаборативну фільтрацію, згорткові та рекурентні нейронні мережі, дерева розв'язків, випадкові ліси, обробку природної мови.*

Очікувані результати навчання.

Фахові компетентності.

ФК 18. Здатність реалізовувати застосунки з використанням концепцій штучного інтелекту, інженерії даних та машинного навчання, зокрема для кібер-фізичних та енергетичних систем.

Програмні результати навчання.

ПРН 34. Вміти розробляти застосунки з використанням концепцій штучного інтелекту та машинного навчання з врахуванням специфіки предметної області енергетичної галузі.

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

Дисципліна вивчається у сьомому семестрі. Пререквізитами є курси “Вища математика”, “Теорія ймовірностей” та один з курсів, де використовується мова Python. Постреквізитів у даного курсу на бакалаврському рівні немає.

3 Зміст навчальної дисципліни

- 1. Вступ до глибокого навчання.*
- 2. Від моделі до застосунку.*
- 3. Моделі класифікатора.*
- 4. Класифікація зображень*
- 5. Інші задачі комп'ютерного зору*
- 6. Тренування сучасної моделі.*
- 7. Колаборативна фільтрація.*
- 8. Табличні моделі.*
- 9. Обробка природної мови: рекурентні нейронні мережі.*

10. Обробка даних за допомогою бібліотеки *fastai*.
11. Побудова мовної моделі з нуля.
12. Згорткові нейронні мережі.
13. Моделі типу *ResNet*.
14. Архітектура застосунків.
15. Процес тренування моделей.
16. Побудова нейронної мережі з нуля.
17. Згорткові моделі з *CAM*.
18. Побудова класу *fastai Learner* з нуля. Підсумкові думки.

4 Навчальні матеріали та ресурси

Базова література:

Howard, J., Gugger, S. *Deep Learning for Coders with fastai and PyTorch: AI Applications Without a PhD*: Sebastopol, CA: O'Reilly Media, 2020. 624 с. URL: <http://gen.lib.rus.ec/book/index.php?md5=2668D405E21C8708BB3192CDD92D8DAF>

Додаткова література:

1. Albon, C. *Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning*: Sebastopol, CA: O'Reilly Media, 2018. 366 с. URL: <http://gen.lib.rus.ec/book/index.php?md5=1b8a00c4b487665f8c785761b3bb8f4b>
2. Harrison, M. *Machine Learning Pocket Reference: Working with Structured Data in Python*: Sebastopol, CA: O'Reilly Media, 2019. 320 с. URL: <http://gen.lib.rus.ec/book/index.php?md5=904606F860FEA95C671B3821CBBDE6C7>
3. Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*: Sebastopol, CA: O'Reilly Media, 2019. 856 с. URL: <http://gen.lib.rus.ec/book/index.php?md5=E98CBB67C93AD4EF9C550E215E6248BD>
4. Chollet, F. *Deep Learning with Python*: Shelter Island, NY: Manning Publications, 2017. 384 с. URL: <http://libgen.is/item/index.php?md5=584B39E75A5B9E072467AFD6A684D0FB>
5. *Machine Learning Crash Course | Google Developers*: URL: <https://developers.google.com/machine-learning/crash-course>
6. Elger, P., Shanaghy, E. *AI as a Service*: Shelter Island, NY: Manning Publications, 2020. 325 с. URL: <http://gen.lib.rus.ec/book/index.php?md5=68556641E787AFDC9C8BD105D707B27E>
7. Ameisen, E. *Building Machine Learning Powered Applications: Going from Idea to Product*: Sebastopol, CA: O'Reilly Media, 2020. 260 с. URL: <http://libgen.is/book/index.php?md5=7253B8C7533DAA5F7747F332B2ED7D2F>
8. Hurbans, R. *Grokking Artificial Intelligence Algorithms*: Shelter Island, NY: Manning Publications, 2020. 392 с. URL: <http://gen.lib.rus.ec/book/index.php?md5=050DE26CAF73D5D7196B5543ED15530F>
9. Trask, A. W. *Grokking Deep Learning*: Shelter Island, NY: Manning Publications, 2019. 325 с. URL: <http://gen.lib.rus.ec/book/index.php?md5=9beb350de2902304edb5e437a1c5ab3f>
10. Rao, D., McMahan, B. *Natural Language Processing with PyTorch: Build Intelligent Language Applications Using Deep Learning*: Sebastopol, CA: O'Reilly Media, 2019. 256 с. URL: <https://learning.oreilly.com/library/view/natural-language-processing/9781491978221/>

11. Morales, M. *Grokking Deep Reinforcement Learning: Shelter Island, NY: Manning Publications, 2019.* 450 c. URL: <http://gen.lib.rus.ec/book/index.php?md5=F83946E7A76E64E0C309CB9A2086551E>
12. Zai, A., Brown, B. *Deep Reinforcement Learning in Action: Shelter Island, NY: Manning Publications, 2020.* 277 c. URL: <http://gen.lib.rus.ec/book/index.php?md5=F2F28879523CAE239FFB89EA6A533A1A>
13. Koul, A., Ganju, S., Kasam, M. *Practical Deep Learning for Cloud, Mobile, and Edge: Real-World AI & Computer-Vision Projects Using Python, Keras & TensorFlow: Sebastopol, CA: O'Reilly Media, 2019.* 620 c. URL: <http://gen.lib.rus.ec/book/index.php?md5=4EE9011651DEBD8F5F3B5A9503E48921>
14. Pointer, I. *Programming PyTorch for Deep Learning: Creating and Deploying Deep Learning Applications: Sebastopol, CA: O'Reilly Media, 2019.* 220 c. URL: <http://gen.lib.rus.ec/book/index.php?md5=38984CF4F0C132A92DD1B20578A116DA>
15. Foster, D. *Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play: Sebastopol, CA: O'Reilly Media, 2019.* 330 c. URL: <http://libgen.is/book/index.php?md5=41fd02aff65fdb65143d0a9ff0c60fda>
16. Lane, H., Hapke, H., Howard, C. *Natural Language Processing in Action: Understanding, analyzing, and generating text with Python: Shelter Island, NY: Manning Publications, 2019.* 544 c. URL: <http://libgen.is/item/index.php?md5=dc168df28d64176a28151ccee6674d5a>
17. Bengfort, B., Bilbro, R., Ojeda, T. *Applied Text Analysis with Python: Enabling Language-Aware Data Products with Machine Learning: Sebastopol, CA: O'Reilly Media, 2018.* 332 c. URL: <http://libgen.io/book/index.php?md5=e0efddaa7b373e509d189655ba3189b0>
18. Raschka, S., Mirjalili, V. *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2: Birmingham, UK: Packt Publishing, 2019.* 770 c. URL: <http://gen.lib.rus.ec/book/index.php?md5=6ab53d6bd918569dbbf95eed84ac51ab>
19. Cai, S., Bileschi, S., Nielsen, E. *Deep Learning with JavaScript: Neural networks in TensorFlow.js: Shelter Island, NY: Manning Publications, 2020.* 350 c. URL: <http://gen.lib.rus.ec/book/index.php?md5=0c26a4ef4a5e3f77c4f3a20a76af4c48>

Навчальний контент

5 Методика опанування навчальної дисципліни (освітнього компонента)

1. Вступ до глибокого навчання.
 - 1.1. Нейронні мережі: стисла історія.
 - 1.2. Як вивчати глибоке навчання.
 - 1.2.1. Ваші проекти та ваш тип мислення.
 - 1.3. Бібліотеки: PyTorch, fastai, and Jupyter (і чому це не має особливого значення).
 - 1.4. Ваша перша модель.
 - 1.4.1. Робота з сервером GPU для глибокого навчання.
 - 1.4.2. Запуск першого ноутбуку.
 - 1.4.3. Що таке машинне навчання?
 - 1.4.4. Що таке нейронна мережа?
 - 1.4.5. Термінологія глибокого навчання.
 - 1.4.6. Невіддільні обмеження глибокого навчання.

- 1.4.7. *Як працює наш розпізнавач зображень.*
- 1.4.8. *Чому навчився на розпізнавач зображень.*
- 1.4.9. *Розпізнавачі зображень, які можуть розв'язувати задачі, не пов'язані із зображеннями.*
- 1.4.10. *Повторення термінології.*
- 1.5. *Глибоке навчання — це не тільки класифікація зображень.*
- 1.6. *Множина валідації та тестова множина.*
- 1.7. *Розважливість при визначенні тестових множин.*
- 1.8. *Момент “Вибери свою пригоду”.*
- 2. *Від моделі до продакшну.*
 - 2.1. *Практика глибокого навчання.*
 - 2.1.1. *Починаючи власний проєкт.*
 - 2.1.2. *Стан глибокого навчання.*
 - 2.1.3. *Підхід Drivetrain.*
 - 2.2. *Збирання даних.*
 - 2.3. *Від Data до DataLoaders.*
 - 2.3.1. *Аугментація даних.*
 - 2.4. *Тренування моделі та його використання для очищення даних.*
 - 2.5. *Перетворення моделі на вебзастосунок.*
 - 2.5.1. *Використання моделі для виведення.*
 - 2.5.2. *Створення ноутбуку-застосунку з моделі.*
 - 2.5.3. *Перетворення ноутбуку на справжній застосунок.*
 - 2.5.4. *Розгортання застосунку.*
 - 2.6. *Як уникнути катастрофи.*
 - 2.6.1. *Непередбачувані наслідки та зворотний зв'язок.*
 - 2.7. *Починайте вести блог!*
- 3. *Під капотом: тренування класифікатора цифр.*
 - 3.1. *Пікселі: основа комп'ютерного зору.*
 - 3.2. *Перша спроба: схожість пікселів.*
 - 3.2.1. *Масиви NumPy і тензори PyTorch.*
 - 3.3. *Обчислення метрик за допомогою широкомовлення.*
 - 3.4. *Стохастичний градієнтний спуск.*
 - 3.4.1. *Обчислення градієнтів.*
 - 3.4.2. *Ітерації з темпом навчання.*
 - 3.4.3. *Приклад СГС.*
 - 3.4.4. *Підсумки СГС.*
 - 3.5. *Функція втрат MNIST.*

- 3.5.1. *Сигмоїд.*
- 3.5.2. *СГС і мініпакети.*
- 3.6. *Складаючи все разом.*
 - 3.6.1. *Створення оптимізатора.*
- 3.7. *Додавання нелінійності.*
 - 3.7.1. *Йдемо глибше.*
- 3.8. *Повторення термінології.*
- 4. *Класифікація зображень.*
 - 4.1. *Від собак і котів до порід тварин.*
 - 4.2. *Попередє приведення до єдиного розміру.*
 - 4.2.1. *Перевірка та налагодження DataBlock.*
 - 4.3. *Крос-ентропійні втрати.*
 - 4.3.1. *Перегляд активацій та міток.*
 - 4.3.2. *Softmax.*
 - 4.3.3. *Логарифмічний критерій подібності.*
 - 4.3.4. *Взяття логарифму.*
 - 4.4. *Інтерпретація моделей.*
 - 4.5. *Покращання нашої моделі.*
 - 4.5.1. *Пошук значення темпу навчання.*
 - 4.5.2. *Розмороження та переніс навчання.*
 - 4.5.3. *Дискримінантний темп навчання.*
 - 4.5.4. *Вибір кількості епох.*
 - 4.5.5. *Глибокі архітектури.*
 - 4.6. *Висновки.*
- 5. *Інші задачі комп'ютерного зору.*
 - 5.1. *Класифікація з багатьма мітками.*
 - 5.1.1. *Дані.*
 - 5.1.2. *Побудова DataBlock.*
 - 5.1.3. *Бінарна крос-ентропія.*
 - 5.2. *Регресія.*
 - 5.2.1. *Збирання даних.*
 - 5.2.2. *Тренування моделі.*
 - 5.3. *Висновки.*
- 6. *Тренування сучасної моделі.*
 - 6.1. *Imagenette.*
 - 6.2. *Нормалізація.*
 - 6.3. *Progressive Resizing.*

- 6.4. *Test Time Augmentation.*
- 6.5. *Mixup.*
- 6.6. *Label Smoothing.*
- 6.7. *Висновки.*
- 7. *Занурення у колаборативну фільтрацію.*
 - 7.1. *Перший погляд на дані.*
 - 7.2. *Навчання на прихованих факторах.*
 - 7.3. *Створення DataLoaders.*
 - 7.4. *Колаборативна фільтрація з нуля.*
 - 7.4.1. *Розпад вагів.*
 - 7.4.2. *Створення власної моделі вкладення.*
 - 7.5. *Інтерпретація вкладень та зміщень.*
 - 7.5.1. *Використання fastai.collab.*
 - 7.5.2. *Відстань вкладення.*
 - 7.6. *Початкове завантаження моделі колаборативної фільтрації.*
 - 7.7. *Глибоке навчання для колаборативної фільтрації.*
 - 7.8. *Висновки.*
- 8. *Занурення у табличне моделювання.*
 - 8.1. *Категоріальні вкладення.*
 - 8.2. *Поза глибоким навчанням.*
 - 8.3. *Датасет.*
 - 8.3.1. *Змагання Kaggle.*
 - 8.3.2. *Подивимося на дані.*
 - 8.4. *Дерева ухвалення рішень.*
 - 8.4.1. *Обробка дат.*
 - 8.4.2. *Використання TabularPandas і TabularProc.*
 - 8.4.3. *Створення дерева ухвалення рішень.*
 - 8.4.4. *Категоріальні змінні.*
 - 8.5. *Випадкові ліси.*
 - 8.5.1. *Створення випадкового лісу.*
 - 8.5.2. *Помилка Out-of-Bag.*
 - 8.6. *Інтерпретація моделі.*
 - 8.6.1. *Варіація дерев для впевненості прогнозування.*
 - 8.6.2. *Важливість ознак.*
 - 8.6.3. *Усунення маловажливих змінних.*
 - 8.6.4. *Усунення надлишкових ознак.*
 - 8.6.5. *Часткова залежність.*

- 8.6.6. *Витік даних.*
- 8.6.7. *Інтерпретатор дерев.*
- 8.7. *Екстраполяція та нейронні мережі.*
 - 8.7.1. *Задача екстраполяції.*
 - 8.7.2. *Знаходження позадоменних даних.*
 - 8.7.3. *Використання нейромережі.*
- 8.8. *Ансамблеве навчання.*
 - 8.8.1. *Бустінг.*
 - 8.8.2. *Поєднання вкладень з іншими методами.*
- 8.9. *Висновки.*
- 9. *Занурення у обробку природної мови: рекурентні нейромережі.*
 - 9.1. *Попередня обробка тексту.*
 - 9.1.1. *Токенизація.*
 - 9.1.2. *Токенизація слів з fastai.*
 - 9.1.3. *Токенизація підрядків.*
 - 9.1.4. *Нумерикалізація з fastai.*
 - 9.1.5. *Розкладання текстів у пакети для мовної моделі.*
 - 9.2. *Тренування класифікатора текстів.*
 - 9.2.1. *Мована модель з використанням DataBlock.*
 - 9.2.2. *Тонке налаштування мовної моделі.*
 - 9.2.3. *Збереження та завантаження моделей.*
 - 9.2.4. *Генерація тексту.*
 - 9.2.5. *Створення DataLoaders класифікаторів.*
 - 9.2.6. *Тонке налаштування класифікаторів.*
 - 9.3. *Дезінформація та мовна модель.*
 - 9.4. *Висновки.*
- 10. *Обробка даних з fastai's Layered API.*
 - 10.1. *Занурюємося глибше у fastai's Layered API.*
 - 10.1.1. *Трансформації.*
 - 10.1.2. *Написання власних трансформацій.*
 - 10.1.3. *Конвеєр (Pipeline).*
 - 10.2. *TfmdLists і Datasets: трансформовані колекції.*
 - 10.2.1. *TfmdLists.*
 - 10.2.2. *Datasets.*
 - 10.3. *Applying the Mid-Level Data API: SiamesePair.*
 - 10.4. *Conclusion.*
- 11. *Мовна модель з нуля.*

- 11.1. Дані.
 - 11.2. Наша перша мовна модель з нуля.
 - 11.2.1. Наша мовна модель у PyTorch.
 - 11.2.2. Наша перша рекурентна нейромережа (RNN).
 - 11.3. Покращення RNN.
 - 11.3.1. Підтримка стану RNN.
 - 11.3.2. Створюємо більше сигналів.
 - 11.4. Багатшарові RNN.
 - 11.4.1. Модель.
 - 11.4.2. Вибухове розширення або зникнення активацій.
 - 11.5. LSTM.
 - 11.5.1. Побудова LSTM з нуля.
 - 11.5.2. Тренування мовної моделі з використанням LSTM.
 - 11.6. Регуляризація LSTM.
 - 11.6.1. Виключення (Dropout).
 - 11.6.2. Регуляризація активації та тимчасова регуляризація активації.
 - 11.6.3. Тренування прив'язаної до вагів регуляризованої LSTM.
 - 11.7. Висновки.
12. Згорткові нейромережі (CNN).
- 12.1. Чародійство згортки.
 - 12.1.1. Відбивання згорткового ядра.
 - 12.1.2. Згортки у PyTorch.
 - 12.1.3. Переступлення та набивка (Strides and Padding).
 - 12.1.4. Розуміння згорткових рівнянь.
 - 12.2. Наша перша згорткова нейромережа.
 - 12.2.1. Створення CNN.
 - 12.2.2. Розуміння згорткової арифметики.
 - 12.2.3. Сприймальні поля.
 - 12.2.4. Зауваження щодо Twitter.
 - 12.3. Кольорові зображення.
 - 12.4. Покращення стабільності тренування.
 - 12.4.1. Проста основа.
 - 12.4.2. Збільшення розміру пакету.
 - 12.4.3. Тренування 1cycle.
 - 12.4.4. Пакетна нормалізація.
 - 12.5. Висновки.
13. ResNets.

- 13.1. Повернення до *Imagenette*.
- 13.2. Побудова сучасної CNN: *ResNet*.
 - 13.2.1. Пропустити зв'язки (*Skip Connections*).
 - 13.2.2. Сучасна *ResNet*.
 - 13.2.3. Шари пляшкової горлянки (*Bottleneck Layers*).
- 13.3. *Conclusion*.
- 14. Глибоке занурення до архітектури застосунків.
 - 14.1. Комп'ютерний зір.
 - 14.1.1. *cnn_learner*.
 - 14.1.2. *unet_learner*.
 - 14.1.3. Сіамська мережа.
 - 14.2. Обробка природної мови.
 - 14.3. Модуль *Tabular*.
 - 14.4. Висновки.
- 15. Процес тренування.
 - 15.1. Створення бази.
 - 15.2. Загальний оптимізатор.
 - 15.3. Імпульс.
 - 15.4. *RMSProp*.
 - 15.5. *Adam*.
 - 15.6. Від'єднаний розпад вагів.
 - 15.7. Відкликання.
 - 15.7.1. Створення відкликання.
 - 15.7.2. Впорядкування відкликаних та виключні ситуації.
 - 15.8. Висновки.
- 16. Нейромережа з самого початку.
 - 16.1. Побудова шару нейромережі з нуля.
 - 16.1.1. Моделювання нейрону.
 - 16.1.2. Добування матриць з нуля.
 - 16.1.3. Поелементна арифметика.
 - 16.1.4. Широкомовлення (*Broadcasting*).
 - 16.1.5. Ейнштейнівське підсумовування.
 - 16.2. Прямі та зворотні проходи.
 - 16.2.1. Визначення та ініціалізація шару.
 - 16.2.2. Градієнти та зворотний прохід.
 - 16.2.3. Рефакторинг моделі.
 - 16.2.4. Звернення до *PyTorch*.

17. Інтерпретація CNN з відбиванням активації класів (CAM).

17.1. Відбивання активації класів (Class Activation Map CAM) та хуки (Hooks).

17.2. Градієнт CAM.

17.3. Висновки.

17.4. Подальші дослідження.

18. Побудова fastai Learner з нуля. Підсумки курсу.

18.1. Дані.

18.1.1. Датасет.

18.2. Класи Module та Parameter.

18.2.1. Проста CNN.

18.3. Втрати.

18.4. Клас Learner.

18.4.1. Зворотні виклики (Callbacks).

18.4.2. Складання розкладу та темп навчання.

18.5. Висновки.

6 Самостійна робота студента/аспіранта

Студент витратить 3-4 година на тиждень на самостійну роботу з матеріалом курсу.

Політика та контроль

7 Політика навчальної дисципліни (освітнього компонента)

Студенти отримують бали за правильне та вчасне виконання лабораторних робіт. Загальний рейтинг (кількість балів) складається з: 1) лабораторних робіт (у формі практичних завдань з програмування) 60%, 2) екзамену 40%.

Наразі в курсі наявні три лабораторні роботи, кожне оцінюється до 20 балів. Студент повинен здати правильно виконану лабораторну роботу протягом двох тижнів з дня видачі завдання для отримання повної кількості балів, в іншому випадку застосовуються штрафні бали не більше 40% від загальної кількості за лабораторну роботу.

8 Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Поточний контроль: МКР, лабораторні роботи

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

Семестровий контроль: екзамен

Умови допуску до семестрового контролю: зарахування усіх лабораторних робіт, семестровий рейтинг не менше 40 балів.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

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

9 Додаткова інформація з дисципліни (освітнього компонента)

- *перелік питань, які виносяться на семестровий контроль (див. додаток до силабусу).*

Робочу програму навчальної дисципліни (силабус):

Складено Професор кафедри інженерії програмного забезпечення в енергетиці, д.е.н., професор А. О. Сігайов

Ухвалено кафедрою інженерії програмного забезпечення в енергетиці (протокол № 28 від 15 травня 2023 р.)

Погоджено Методичною комісією Навчально-наукового інституту атомної і теплової енергетики КПІ ім. Ігоря Сікорського (протокол № 9 від 26 травня 2023 р.)