



Проектування алгоритмів

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення інформаційних систем</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>заочна</i>
Рік підготовки, семестр	<i>2 курс, осінній семестр</i>
Обсяг дисципліни	<i>4 кредити (120 год)</i>
Семестровий контроль/ контрольні заходи	<i>Залік / лабораторні роботи, модульна контрольна робота</i>
Розклад занять	<i>http://rozklad.kpi.ua/Schedules/ScheduleGroupSelection.aspx</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: ст. викл. PhD Головченко Максим Миколайович, k.o.o.v.g.v.n.s@gmail.com</i> <i>Лабораторні: ст. викл. PhD Головченко Максим Миколайович, k.o.o.v.g.v.n.s@gmail.com</i>
Розміщення курсу	<i>ipi.kpi.ua/navchannya/silabusy/</i>

Програма навчальної дисципліни

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

Навчальна дисципліна "Проектування алгоритмів", спрямована на вивчення теоретичних та методологічних основ побудови нових алгоритмів та модифікації існуючих з подальшою реалізацією на мовах програмування високого рівня з урахуванням сучасних концепцій і тенденцій розвитку при вирішенні прикладних задач різного ступеня складності. Окрім безпосереднього створення алгоритмів дисципліна ставить перед собою задачі дослідження ефективності розроблених алгоритмів чи модифікацій існуючих.

Відповідна теоретична та практична підготовка формує базові професійні навички з проектування алгоритмів і є основою для успішного опановування інших фахових дисциплін.

Метою дисципліни (кредитного модуля) є формування у студентів здатності проектувати та розробляти алгоритми вирішення прикладних задач і виконувати їх програмну реалізацію із подальшим дослідженням ефективності.

Предмет навчальної дисципліни (кредитного модуля) – алгоритми, повнота та оптимальність алгоритмів, часова та ємнісна складність алгоритмів, дослідження алгоритмів, процес розробки алгоритмів.

Вивчення дисципліни «Проектування алгоритмів» сприяє формуванню у студентів наступних компетентностей.

Загальні компетентності:

- ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.
- ЗК2. Здатність застосовувати знання у практичних ситуаціях.
- ЗК5. Здатність вчитися і оволодівати сучасними знаннями.
- ЗК6. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

Фахові компетентності, формуванню яких сприяє даний кредитний модуль:

- ФК1. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.
- ФК2. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.
- ФК3. Здатність розробляти архітектури, модулі та компоненти програмних систем.
- ФК4. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами.
- ФК5. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.
- ФК6. Здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки (в тому числі кібербезпеки)
- ФК7. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.
- ФК8. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.
- ФК9. Здатність оцінювати і враховувати економічні, соціальні, технологічні та екологічні чинники, що впливають на сферу професійної діяльності.
- ФК10. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.
- ФК11. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.
- ФК12. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.
- ФК13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.
- ФК14. Здатність до алгоритмічного та логічного мислення.

Формування зазначених компетентностей забезпечує досягнення наступних програмних результатів навчання:

- ПРН1. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.
- ПРН13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

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

Для успішного засвоєння дисципліни необхідно мати базові знання з дисципліни "Основи програмування", дисципліни "Комп'ютерна дискретна математика" та дисципліни "Алгоритми і структури даних". Успішне оволодіння знаннями з дисципліни готує студентів до вивчення наступних дисциплін, таких як:

- «Аналіз даних в інформаційних системах» (ПО 16),
- «Курсова робота з аналізу даних в інформаційно-управляючих системах» (ПО 17).

3. Зміст навчальної дисципліни

Розділ 1. Проектування алгоритмів зовнішнього сортування

Тема 1.1 Базові алгоритми зовнішнього сортування

Тема 1.2 Модифіковані алгоритми зовнішнього сортування

Розділ 2. Проектування алгоритмів пошуку у просторі станів

Тема 2.1. Проектування алгоритмів неінформативного пошуку

Тема 2.2. Проектування алгоритмів інформативного пошуку

Тема 2.3. Проектування алгоритмів локального пошуку

Розділ 3. Проектування алгоритмів обробки складних структур даних

Тема 3.1. Проектування алгоритмів обробки індексно-прямих та індексно-послідовних структур

Тема 3.2. Проектування алгоритмів обробки самобалансуючих деревовидних структур

Тема 3.3. Проектування алгоритмів обробки B-дерев

Розділ 4. Проектування евристичних та метаевристичних алгоритмів

Тема 4.1. Генетичні алгоритми та їх модифікації

Тема 4.2. Мурашині алгоритми та їх модифікації

Тема 4.3. Бджолині алгоритми та їх модифікації

4. Навчальні матеріали та ресурси

Базова література

1. Вступ до алгоритмів, 3-є видання: [Підручник, пер. з англ.] / Томас Г. Кормен, Чарльз Е. Лейзерсон, Рональд Л. Рівест, Кліфорд Стайн. – К. : К.І.С., 2019. – 1288с.
2. Artificial Intelligence: A Modern Approach, 4th US [Підручник] / Stuart Russell, Peter Norvig. - Pearson, 2021. – 1096р.

Допоміжна література

1. С++. Основи програмування. Теорія та практика: підручник / [О. Г. Трофименко, Ю. В. Прокоп, І. Г. Швайко, Л. М. Буката та ін.] ; за ред. О. Г. Трофименко. – Одеса : Фенікс, 2010. – 544 с.
2. Основи програмування. Частина 2. Модульне програмування [Електронний ресурс] : навч. посіб. для студ. спеціальності 121 «Інженерія програмного забезпечення» / КПІ ім. Ігоря Сікорського; уклад.: І.П. Муха, І.І.Вітковська, М.М. Головченко. – Київ: КПІ ім. Ігоря Сікорського, 2022. – 83 с.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

5.1 Лекційні заняття

Матеріали для вивчення дисципліни розміщені в електронному вигляді на сайті (<http://ipi.kpi.ua/study/iii-semester/>) та в Telegram-каналі дисципліни. Контент доступний студенту із будь-якого місця в мережі Інтернет.

Лекції по дисципліні проводяться із використанням сучасних мультимедійних презентаційних технологій.

№	Назва теми лекції та перелік основних питань
Розділ 1. Проєктування алгоритмів зовнішнього сортування	
1.	<i>Базові алгоритми зовнішнього сортування</i> Поняття зовнішнього сортування. Алгоритм прямого злиття, алгоритм природнього злиття, алгоритм збалансованого багатошляхового злиття, алгоритм багатофазного сортування.
2.	<i>Модифіковані алгоритми зовнішнього сортування</i> Відображення файлу у пам'ять. Сортування частинами.
Розділ 2. Проєктування алгоритмів пошуку у просторі станів	
3.	<i>Проєктування алгоритмів неінформативного пошуку</i> Поняття пошуку у просторі станів. Алгоритми пошуку у дереві та у графі. Пошук у ширину, пошук в глибину, пошук з обмеженням глибини, пошук з ітеративним заглибленням. Прикладні задачі.
4.	<i>Проєктування алгоритмів інформативного пошуку</i> Поняття евристичної функції. Пошук за першим кращим співпадінням, пошук A*, рекурсивний пошук за першим кращим співпадінням. Прикладні задачі.
5.	<i>Проєктування алгоритмів локального пошуку</i> Поняття локального розв'язку. Алгоритм із сходженням на вершину, алгоритм пошуку з імітацією відпалу, променевий пошук.
Розділ 3. Проєктування алгоритмів обробки складних структур даних	
6.	<i>Проєктування алгоритмів обробки індексно-прямих та індексно-послідовних структур</i> Файлові структури даних. Файли з щільним та нещільним індексом. Алгоритми додавання, видалення, пошуку та редагування у цих структурах.
7.	<i>Проєктування алгоритмів обробки самобалансуючих деревовидних структур</i> Самобалансуючі дерева пошуку. Червоно-чорне та AVL-дерево. Алгоритми додавання, видалення, пошуку та редагування у цих структурах.
8.	<i>Проєктування алгоритмів обробки B-дерев</i> Алгоритми додавання, видалення, пошуку та редагування у B-дереві.
Розділ 4. Проєктування евристичних та метаевристичних алгоритмів	
9.	<i>Генетичні алгоритми та їх модифікації</i> Генетичний алгоритм. Поняття хромосома, ген, селекція, схрещування, мутація, локальне покращення у рамках генетичного алгоритму. Прикладні задачі, які вирішує даний тип алгоритмів. Можливі модифікації даного типу алгоритмів.
10.	<i>Мурашині алгоритми та їх модифікації</i> Мурашиний алгоритм. Поняття мурах та елітної мурахи, феромон, нюх у рамках мурашиного алгоритму. Прикладні задачі, які вирішує даний тип алгоритмів. Можливі модифікації даного типу алгоритмів.
11.	<i>Бджолині алгоритми та їх модифікації</i> Бджолиний алгоритм. Поняття ділянка, розвідник, фуражир у рамках бджолиного алгоритму. Прикладні задачі, які вирішує даний тип алгоритмів. Можливі модифікації даного типу алгоритмів.

За умови дистанційного навчання заняття, у тому числі контрольні заходи, проводяться з використанням сервісу Zoom.

5.2 Лабораторні роботи

№ з/п	Назва лабораторної роботи (комп'ютерного практикуму)	Кількість ауд. годин
1.	Зовнішні алгоритми сортування	4
2.	Неінформативний, інформативний та локальний пошук	4
3.	Проектування структур даних	4
4.	Сучасні евристичні алгоритми ч.1	4

Лабораторні заняття виконуються з використанням мов програмування C++, Python та ін. та версій середовищ розробки програм (IDE), які надаються розробниками для навчальних цілях безкоштовно. Під час їх проведення використовуються методичні вказівки до виконання лабораторних робіт по дисципліні, розміщені на сайті (<http://ipi.kpi.ua/study/iii-semester/>) та Telegram-каналі дисципліни а також в середовищі «Електронний кампус». Для виконання кожної лабораторної роботи студенту надається не більше 3 тижнів.

За умови дистанційного навчання заняття, у тому числі контрольні заходи, проводяться з використанням сервісу Google Meet (Zoom).

6. Самостійна робота студента

До самостійної роботи студента відноситься виконання індивідуальних завдань з тематики, яка виноситься на лабораторні роботи, а також опрацювання теоретичного матеріалу за наданими текстами лекцій та додатковою літературою, у тому числі за темами, які винесені на самостійне вивчення (згідно таблиці 1). Усі навчальні матеріали (тексти лекцій, методичні вказівки до виконання лабораторних робіт) розміщені в електронному вигляді на сайті (<http://ipi.kpi.ua/study/iii-semester/>), у Telegram-каналі дисципліни а також в середовищі «Електронний кампус». Навчальний контент доступний із будь-якого місця в мережі Інтернет.

На самостійну роботу студент має витратити кількість годин, що співмірна із кількістю годин, проведених ним на аудиторних заняттях.

Таблиця 1. Питання, які виносяться на самостійне опрацювання

№ з/п	Назва теми, що виноситься на самостійне опрацювання
1.	Алгоритми швидко читання та запису у файли.
2.	Додаткові евристичні функції
3.	Алгоритми пошуку у структурах даних
4.	Алгоритми хешування даних
5.	Робота з базовими деревовидними структурами
6.	Алгоритми локального покращення
7.	Алгоритми штучного інтелекту для карткових ігор та їм подібних

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Студент має вивчати дисципліну (кредитний модуль) протягом семестру, дотримуючись календарного плану виконання завдань комп'ютерного практикуму, вивчення тем лекційного матеріалу та виконання контрольних і модульних контрольних робіт. Усі завдання студент має виконувати **самостійно** і **вчасно**. Завдання з лабораторної роботи вважається виконаним, якщо студент захистив його у викладача (показав працездатність, відповів на усі питання) та розмістив

звіт з виконання даної роботи на Github / Classroom / «Електронний кампус». Несвоєчасним вважається виконання завдання із затримкою більше 1 тижня без поважної причини. За несвоєчасну здачу лабораторних робіт передбачені штрафні бали. Такі обмеження стимулюють студента організувати систематичне виконання завдань та не допускати значного накопичення незданих лабораторних робіт на кінець семестру.

Оцінювання студентів здійснюється згідно рейтингової оцінки рівня підготовки студентів з дисципліни. Поточний стан успішності студенти можуть бачити в системі «Електронний кампус».

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

8.1 Поточний контроль

Вагові бали кожної лабораторної роботи наведені у таблиці 1. Сумарний ваговий бал за даний контрольний захід (r_3) складає **100 балів**.

Критерії оцінювання лабораторних робіт включають якість її виконання, захисту та оформлення звіту (таблиця 1).

Таблиця 1. Вагові бали та критерії оцінювання комп'ютерних практикумів

№ п/п	Назва роботи	Бали					
		Проектування алгоритму	Аналіз алгоритму	Програмна реалізація	Дослідження алгоритму	Звіт	Сума
1.	№ 1 Зовнішні алгоритми сортування	5	5	5	5	5	25
2.	№ 2 Неінформативний, інформативний та локальний пошук	5	5	5	9	1	25
3.	№ 3 Проектування структур даних	1	1	10	10	3	25
4.	№ 4 Сучасні евристичні алгоритми ч.1	5	5	5	5	5	25
Разом за лабораторні роботи							100

Критерії оцінювання лабораторних робіт унікальні для кожної роботи і записані в кінці протоколу лабораторної роботи.

У випадку несвоєчасного виконання та подання на захист лабораторної роботи верхня межа оцінки знижується відповідно до критеріїв у кінці протоколу лабораторної роботи.

1. Умови позитивної проміжної атестації

Для отримання “зараховано” з першої проміжної атестації (8 тиждень) студент повинен мати не менше ніж 15 балів (за умови, якщо на початок 8 тижня згідно з календарним планом контрольних заходів “ідеальний” студент має отримати 30 балів).

Для отримання “зараховано” з другої проміжної атестації (14 тиждень) студент повинен мати не менше ніж 25 балів (за умови, якщо на початок 14 тижня згідно з календарним планом контрольних заходів “ідеальний” студент має отримати 50 балів).

2. Розрахунок шкали рейтингу R :

Максимальна сума вагових балів контрольних заходів протягом семестру складає:

$$R_c = 100 \text{ балів}$$

Таким чином, рейтингова шкала з дисципліни складає $R = R_c = 100$ балів.

8.2 Календарний контроль

Проводиться двічі на семестр як моніторинг поточного стану виконання вимог робочої програми.

8.3 Семестровий контроль

Семестровий контроль результатів навчання проводиться у вигляді заліку.

Необхідною умовою допуску до заліку є:

– виконання і захист усіх лабораторних робіт, при цьому має бути виконано не менше ніж 60% від загального обсягу завдань у кожній лабораторній.

Розмір шкали рейтингу $R = 100$ балів.

Підсумкова оцінка переводиться до залікової оцінки згідно з таблицею:

<i>Кількість балів</i>	<i>Оцінка</i>
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

Робочу програму навчальної дисципліни (силабус):

Складено ст.вик. PhD кафедри ІІІ М. Головченко

Ухвалено кафедрою ІІІ (протокол № 16_ від _29.05.2024)

Погоджено Методичною комісією факультету (протокол № 10 від 21.06.2024)