



Відділення інженерії програмного забезпечення  
Циклова комісія програмування

## СИЛАБУС

Базова інформація про дисципліну	
<b>Назва дисципліни</b>	Алгоритми та структури даних
<b>Рівень вищої освіти / фахової передвищої освіти</b>	Фахова передвища
<b>Галузь знань</b>	12 «Інформаційні технології»
<b>Спеціальність</b>	121 «Інженерія програмного забезпечення»
<b>Освітня програма</b>	121 «Інженерія програмного забезпечення»
<b>Семестр</b>	5 семестр (9 кл), 3 семестр (11 кл)
<b>Факультет / відділення</b>	Програмування
<b>Курс</b>	3 курс (9 кл), 2 курс (11 кл)
<b>Анотація курсу</b>	Предмет передбачає вивчення структур даних та базових алгоритмів з їх використанням, розгляд популярних алгоритмічних задач. Розширені структури даних мають широке застосування в реалізації багатьох програмних підсистем, які управляють даними, зокрема, прокладають маршрути, аналізують структуру коду, оптимізують роботу з базами даних та їх розміри тощо. Знайомство з особливостями стандартних структур даних та алгоритмів, що працюють з ними, дозволяє компетентно керувати наявними наборами даних при їх програмній обробці, підбирати бібліотечні модулі в конкретних задачах чи писати власні, краще розуміти особливості доменного моделювання.
<b>Сторінка курсу в MOODLE</b>	<a href="http://78.137.2.119:1919/m72/course/view.php?id=84">http://78.137.2.119:1919/m72/course/view.php?id=84</a>
<b>Мова викладання</b>	Українська
<b>Лектор курсу</b>	Викладач II категорії Марченко С.В.

<b>Місце дисципліни в освітній програмі</b>	
<b>Освітня програма</b>	<a href="http://csbc.edu.ua/documents/otdel/moop_pr.pdf">http://csbc.edu.ua/documents/otdel/moop_pr.pdf</a>
<b>Перелік загальних компетентностей (ЗК)</b>	Здатність застосовувати знання у практичних ситуаціях.
<b>Перелік спеціальних компетентностей (СК)</b>	<p>Здатність до алгоритмічного та логічного мислення.</p> <p>Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводу програмного забезпечення та визнання важливості навчання протягом усього життя.</p> <p>Здатність застосовувати фундаментальні та міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.</p> <p>Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу програмного забезпечення.</p> <p>Здатність аналізувати, вибирати та застосовувати методи і засоби для забезпечення інформаційної та функціональної безпеки (в тому числі кібербезпеки).</p> <p>Здатність розробляти архітектури, модулі та компоненти програмного забезпечення.</p> <p>Здатність реалізовувати фази та ітерації життєвого циклу програмного забезпечення на основі відповідних моделей і підходів його розробки.</p>
<b>Перелік програмних результатів навчання</b>	<p>Вміти застосовувати методи обчислення та структури даних для вирішення задач аналізу та синтезу алгоритмів.</p> <p>Вміти систематизувати та узагальнювати інформацію про підходи, методи та засоби розробки супроводу програмного забезпечення, вдосконалювати відповідні знання, вміння й навички протягом усього життя.</p> <p>Знати основні методології та підходи до розробки програмного забезпечення, вміти обирати та реалізовувати модель життєвого циклу програмного забезпечення.</p> <p>Знати способи ідентифікації, формулювання та класифікації вимог до програмного забезпечення та вміти їх застосовувати в процесі</p>

	<p>аналізу отриманого завдання.</p> <p>Знати основні методи оптимізації алгоритмів, вміти розробляти ефективні алгоритми розв'язування завдань та на їх основі створювати програмний код.</p>
<b>Опис дисципліни</b>	
<b>Структура навантаження на студента</b>	<p>Загальна кількість годин – 120</p> <p>Кількість кредитів – 4</p> <p>Кількість лекційних годин – 30</p> <p>Кількість практичних занять – 30</p> <p>Кількість годин для самостійної роботи студентів – 60</p> <p>Форма підсумкового контролю – залік</p>
<b>Методи навчання</b>	<p>За подачею навчального матеріалу: методи готових знань, дослідницький метод.</p> <p>З огляду на мету навчання: методи здобуття нових знань, метод формування умінь і навичок, метод застосування знань на практиці, методи закріплення знань, умінь і навичок, методи перевірки і оцінювання знань, умінь і навичок</p>
<b>Зміст дисципліни</b>	
Тема 1. Алгоритми: методи побудови та аналізу	<p>Поняття алгоритму. Сфери використання алгоритмів. Знаходження найбільшого спільного дільника як алгоритмічний приклад.</p> <p>Аналіз алгоритмів. Використання RAM-моделі для оцінювання ефективності алгоритму. Оцінки ефективності алгоритмів.</p> <p>Загальні методи розробки алгоритмів. Рекурсія. Метод грубої сили. Пошук з поверненням. Метод декомпозиції. Жадібні алгоритми та динамічне програмування</p>
Тема 2. Алгоритми сортування	<p>Алгоритми сортування. Задача сортування та алгоритми її вирішення. Пірамідальне сортування. Швидке сортування. Сортування за лінійний час</p>
Тема 3. Елементарні структури даних	<p>Стеки та черги. Поняття динамічної множини та операції з нею. Реалізація стеків та черг</p> <p>Зв'язні списки. Пошук у зв'язному списку. Вставка та видалення елементів у зв'язному списку.</p>

	Реалізація зв'язних списків на базі трьох масивів. Виділення та вивільнення пам'яті за відсутності вказівників
Тема 4. Хешування та хеш-таблиці	Хеш-таблиці. Таблиці з прямою адресацією. Вирішення колізій за допомогою методу ланцюжків. Хеш-функції. Критерії якості хеш-функції. Методи ділення та множення для побудови хеш-функцій. Універсальні хеш-функції. Хеш-таблиці з відкритою адресацією. Методи зондування для вставки елементів у хеш-таблицю
Тема 5. Незбалансовані дерева та їх застосування	Дерево як структура даних. Бінарні дерева пошуку. Побудова та обхід бінарного дерева пошуку. Пошук у бінарному дереві. Вставка і видалення елементів у бінарному дереві пошуку
Тема 6. Збалансовані дерева пошуку	Самозбалансовані бінарні дерева пошуку. Червоно-чорні дерева. Повороти. Вставка елементів у червоно-чорне дерево. Видалення елементів з червоно-чорного дерева. AVL-дерева: означення та основні операції. В-дерева: означення та основні операції.
Тема 7. Представлення графів та пошук у них	Представлення графів. Обхід графа за допомогою алгоритму пошуку в ширину. Обхід графа за допомогою алгоритму пошуку вглиб. Топологічне сортування
Тема 8. Алгоритми кластеризації на основі графів	Системи неперетинних множин. Мінімальні кістякові дерева. Алгоритм Крускала. Алгоритм Пріма
Тема 9. Алгоритми знаходження оптимального шляху	Найкоротші шляхи та релаксація. Алгоритм Дейкстри. Алгоритм Беллмана-Форда. Найкоротші шляхи для всіх пар вершин. Алгоритм Флойда-Уоршола
Тема 10. Швидкі алгоритми роботи з матрицями	Матриці та їх властивості. Алгоритм Штрассена для перемноження матриць. Алгоритм Вінограда для перемноження матриць. Обернення матриць.
Тема 11. Задача пошуку підрядків	Наївний алгоритм пошуку підрядків. Алгоритм Рабіна-Карпа. Пошук підрядків за допомогою скінченних автоматів. Алгоритм Кнута-Морріса-Пратта. Алгоритм Боєра-Мура.

<b>Політика дисципліни</b>													
<b>Політика відвідування</b>	Регулярне відвідування всіх видів занять, своєчасність виконання самостійної роботи. За об'єктивних причин (наприклад, хвороба, міжнародне стажування) навчання зорганізується в он-лайн формі за погодженням із керівником курсу.												
<b>Політика щодо дедлайнів та перескладання</b>	Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку.												
<b>Академічна доброчесність</b>	У випадку недотримання політики академічної доброчесності (плагіат, самоплагіат, фабрикація, фальсифікація, списування, обман, хабарництво) передбачено повторне проходження оцінювання.												
<b>Система оцінювання</b>													
<p>Поточний контроль здійснюється протягом семестру під час проведення практичних, семінарських та інших видів занять і оцінюється сумою набраних балів (максимальна сума – 100 балів; мінімальна сума, що дозволяє студенту отримати атестацію з предмету – 60 балів); підсумковий/ семестровий контроль, проводиться у формі заліку або іспиту, відповідно до графіку навчального процесу.</p> <p style="text-align: center;"><b>Накопичування рейтингових балів з навчальної дисципліни</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Види навчальної роботи</th> <th style="text-align: right;">Мах кількість балів</th> </tr> </thead> <tbody> <tr> <td>Практичні завдання ( <u>11</u> тем)</td> <td style="text-align: right;">55</td> </tr> <tr> <td>Тестування (5 тем)</td> <td style="text-align: right;">15</td> </tr> <tr> <td>Індивідуальна практична робота</td> <td style="text-align: right;">30</td> </tr> <tr> <td>Разом</td> <td style="text-align: right;">100</td> </tr> <tr> <td> </td> <td> </td> </tr> </tbody> </table>		Види навчальної роботи	Мах кількість балів	Практичні завдання ( <u>11</u> тем)	55	Тестування (5 тем)	15	Індивідуальна практична робота	30	Разом	100		
Види навчальної роботи	Мах кількість балів												
Практичні завдання ( <u>11</u> тем)	55												
Тестування (5 тем)	15												
Індивідуальна практична робота	30												
Разом	100												
<b>Шкала оцінювання</b>													
ECTS	Бали	Зміст											
<b>A</b>	90-100	Бездоганна підготовка в широкому контексті											
<b>B</b>	80-89	Повні знання, міцні вміння											
<b>C</b>	70-79	Хороші знання та вміння											
<b>D</b>	65-69	Задовільні знання, стереотипні вміння											

<b>E</b>	60-64	Виконання мінімальних вимог діяльності в стандартних умовах
<b>FX</b>	35-59	Слабкі знання, відсутність умінь
<b>F</b>	1-34	Необхідний повторний курс

### **Список рекомендованих джерел**

1. Луридаc П. Алгоритмы для начинающих: теория и практика для разработчика. Москва: Эксмо, 2018. 608с.
2. Рафгарден Т. Совершенный алгоритм. Основы. СПб.: Питер, 2019. 256 с.
3. Солтис М. Введение в анализ алгоритмов / пер. с англ. А. В. Логунова. М.: ДМК Пресс, 2019. 278 с.: ил.
4. Кормен Т, Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ. Том 2. 3-е издание. Москва: Диалектика, 2020. 716с.
5. Рафгарден Т. Совершенный алгоритм. Графовые алгоритмы и структуры данных. СПб.: Питер, 2019. 256 с.