

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ»



Ступінь освіти	бакалавр
Спеціальність	124 Системний аналіз
Тривалість викладання	2 семестр
Заняття:	3, 4 чверті
Лекції	2 години на тиждень
Лабораторні	1 години на тиждень
Мова викладання	українська

Кафедра, яка викладає: Інформаційних технологій та комп'ютерної інженерії

Сторінка курсу в СДО НТУ «ДП»: <https://do.nmu.org.ua/course/view.php?id=5113>

Консультації: за окремим розкладом, що попередньо погоджений зі здобувачами освіти.

Онлайн-консультації: MS Teams, електронна пошта

Інформація про викладача:



Викладач:	Соколова Наталя Олегівна, к.т.н., доц.
Персональна сторінка:	https://it.nmu.org.ua/ua/HR_staff/prepods/Sokolova.php
E-mail:	Sokolova.N.O@nmu.one

1. Анотація до курсу

Програма навчальної дисципліни «Алгоритми та структури даних», складена відповідно до освітньо професійної програми підготовки бакалавра 124 «Системний аналіз і» галузі знань 12 «Інформаційні технології».

Алгоритми та структури даних – фундаментальна дисципліна, яка лежить в основі розробки будь яких програмних засобів. В рамках теоретичної частини курсу розкривається різноманіття форм зберігання елементів базових типів даних у відповідних конструкціях структур їхнього зберігання, які підтримуються багатьма відомими та новітніми мовами програмування. У практичній частині розглянуто технологічні і методичні аспекти процесу навчання студентів дисципліні розробки структур зберігання даних і алгоритмів їх обробки, шляхом створення відповідних програмних одиниць на мові програмування C++ в сучасних середовищах розробки. Основу лабораторних робіт складають прийоми роботи з різноманітними структурами даних та відповідними алгоритмами їх обробки за допомогою технологій структурного проектування та конструювання відповідних програм методом «зверху-вниз».

Навчальну дисципліну розроблено таким чином, щоб ознайомити учасників з класичними методами й задачами теорії алгоритмів, структурами. Вивчення цієї дисципліни має на меті зрозуміння та засвоєння основних принципів розробки алгоритмів і програм, а також дає підґрунтя для самостійної практичної роботи в галузі комп'ютерних наук. У курсі вивчаються інформаційні структури даних та обчислювальні моделі.

2. Мета та завдання навчальної дисципліни

Мета – формування компетентностей ефективного вирішення алгоритмічних задач, освоєння фундаментальних ідей і методів теорії алгоритмів, формування системного підходу до вирішення алгоритмічних задач, освоєння базові інформаційних структур даних, формування практичних навичок розробки алгоритмів для розв'язання прикладних задач та їх програмування

Реалізація мети вимагає трансформації програмних результатів навчання в дисциплінарні та адекватний відбір змісту навчальної дисципліни за цим критерієм.

Основними завданнями вивчення дисципліни є

- ознайомити здобувачів вищої освіти з класифікацією структур даних (базових, статичних, полустатичних і динамічних) та їхніми моделями у вигляді абстрактних типів даних, а також видами реалізації елементарних структур у різних мовах програмування;
- розглянути різні види алгоритмів, а також базові алгоритми формування та обробки, як самих елементів структур, так і значень базових типів даних, які зберігаються у цих елементах;
- вивчити та навчитися застосовувати багаточисельні методи сортування та пошуку даних;
- навчити здобувачів вищої освіти формувати та опрацьовувати дані різних типів у спеціалізованих структурах даних: стеках, чергах, хеш-таблицях, деревах, графах та деяких інших.

3. Результати навчання

1. Вміти проектувати, будувати та реалізовувати базові алгоритми обробки різноманітних цифрових даних при створенні новітніх інформаційних технологій в структурі інформаційних систем.
2. Вміти конструювати та реалізовувати відповідні алгоритми обробки цифрових даних засобами сучасних мов програмування.
3. Вміти формалізувати алгоритмічні рішення на всіх етапах збору, передачі та обробки цифрових даних в структурі інформаційних систем та відповідних технологій.
4. Ефективно проводити адміністрування, автоматизувати процеси створення програмних продуктів..

4. Структура курсу

ЛЕКЦІЇ

- 1 Класифікація структур даних.
 - Базові типи даних.
 - Класифікація за видами мінливості.
 - Операції над структурами даних.
- 2 Алгоритми та методи їх опису.
 - Словесні. Графічні. Псевдокод.
 - Ітеративні та рекурсивні алгоритми.
 - Основні категорії, характеристики та аналіз алгоритмів.
- 3 Типи даних лінійної структури.
 - Типи даних лінійної структури з прямим доступом до даних.
 - Алгоритми обробки даних лінійної структури.
- 4 Сортування. Алгоритми сортування масивів
 - Сортування за допомогою вибору
 - Сортування обміном (бульбашка)
 - Сортування вставками

Сортування з поділом (швидке сортування).
Порівняння алгоритмів сортування масивів. Злиття відсортованих послідовностей.

5 Методи пошуку.

Методи, засновані на порівнянні ключів або на цифрових властивостях ключів.
Послідовний пошук
Бінарний пошук.

6 Типи даних лінійної структури з послідовним доступом до даних.

Стеки.
Черги, Черги пріоритетів
Лінійні зв'язні списки
Циклічні списки. Мультисписки

7 Файли. Операції з даними на зовнішніх носіях.

Поняття файлу, фізичні та логічні файли.
Зовнішній пошук, Зовнішнє сортування.
Сортування прямим злиттям. Сортування природним злиттям.

8 Типи даних нелінійної структури. Деревя.

Термінологія дерев. Способи відображення дерев.
Виконавчі (бінарні) дерева. Структура бінарного дерева.
Двійкові дерева виразів. Деревя бінарного пошуку.
Операції з двійковими деревами: пошук по дереву, алгоритми обходу дерева, копіювання і видалення дерев, видалення з дерева.
Збалансовані дерева. Основні визначення.
Вузли AVL-дерева. Включення в збалансоване дерево.
Повороти. Видалення з збалансованого дерева.

9 Графи.

Основні поняття і визначення.
Способи подання графів.
Алгоритми на графах.
Пошук в глибину. Пошук в ширину.
Оптимізаційні алгоритми.
Найкоротші шляхи.
Досяжність і алгоритм Уоршолла.
Найкоротші шляхи між усіма парами вершин. Знаходження центру орієнтованого графа.

ЛАБОРАТОРНІ ЗАНЯТТЯ

1. Обчислення рядів і добутоків з точністю
2. Алгоритми обробки даних лінійної структури
3. Рекурсивні алгоритми
4. Алгоритми сортування
5. Алгоритми пошуку
6. Символи. Рядки. Структури
7. Лінійні динамічні структури даних
8. Файли. Деревя

Технічне обладнання та/або програмне забезпечення

1. Персональний комп'ютер або ноутбук зі сталим доступом до мережі Інтернет
2. Активованій акаунт університетської пошти (student.i.p.@nmu.one) на Ofic365.
3. Активний обліковий запис у системі дистанційної освіти Moodle.
4. Програмне забезпечення:
 - Microsoft Office;
 - Середовище розробки C++.

5. Система оцінювання та вимоги

5.1. Навчальні досягнення здобувачів вищої освіти за результатами вивчення курсу оцінюватимуться за шкалою, що наведена нижче:

Рейтингова шкала	Інституційна шкала
90 – 100	відмінно
75 – 89	добре
60 – 74	задовільно
0 – 59	незадовільно

5.2. Здобувач вищої освіти може отримати **підсумкову оцінку** з навчальної дисципліни на підставі поточного оцінювання знань за умови, якщо набрана кількість балів з поточного тестування та самостійної роботи складатиме не менше 60 балів.

Поточна успішність складається з оцінок за лекційну частину курсу та практичні заняття. Отримані бали осереднюються і є підсумковою оцінкою за вивчення навчальної дисципліни. Максимально за поточною успішністю здобувач вищої освіти може набрати 100 балів.

Максимальне оцінювання:

Теоретична частина	Практична частина		Разом (середнє)
	При своєчасному складанні	При несвоєчасному складанні	
20	80	70 і нижче	100

Практичні завдання приймаються за контрольними запитаннями до кожної роботи.

Теоретична частина оцінюється за результатами здачі контрольних робіт, яка містить 3 задачі.

5.3. Критерії оцінювання теоретичної частини курсу.

Під час проведення контрольних заходів наприкінці третьої, четвертої чверті здобувачі вищої освіти складають контрольну, яка містить 3 задачі. Максимальна оцінка за тест складає 10 балів. Контрольна проводиться з використанням системи дистанційної освіти Ofis365.

5.4. Критерії оцінювання практичних робіт.

З кожної практичної роботи здобувач вищої освіти оформлює звіт, що містить завдання, результати його виконання та висновки. При своєчасному (протягом тижня від дати проведення заняття) та вірному виконанні завдання здобувач отримує оцінку 100. При несвоєчасному виконанні завдання та/або наявності помилок оцінка складає 90 балів і нижче.

6. Політика курсу

6.1. Політика щодо академічної доброчесності

Академічна доброчесність здобувачів вищої освіти є важливою умовою для опанування результатами навчання за дисципліною і отримання задовільної оцінки з поточного та підсумкового контролів. Академічна доброчесність базується на засудженні практик списування (виконання письмових робіт із залученням зовнішніх джерел інформації, крім дозволених для використання), плагіату (відтворення опублікованих текстів інших авторів без зазначення авторства), фабрикації (вигадування даних чи фактів, що

використовуються в освітньому процесі). Політика щодо академічної доброчесності регламентується положенням "Положення про систему запобігання та виявлення плагіату у Національному технічному університеті "Дніпровська політехніка". http://www.nmu.org.ua/ua/content/activity/us_documents/System_of_prevention_and_detection_of_plagiarism.pdf.

У разі порушення здобувачем вищої освіти академічної доброчесності (списування, плагіат, фабрикація), робота оцінюється незадовільно та має бути виконана повторно. При цьому викладач залишає за собою право змінити тему завдання.

6.2. Комунікаційна політика

Здобувачі вищої освіти повинні мати активовану університетську пошту.

Усі письмові запитання до викладачів стосовно курсу мають надсилатися на університетську електронну пошту.

6.3. Політика щодо перескладання

Роботи, які здаються із порушенням термінів без поважних причин оцінюються на нижчу оцінку. Перескладання відбувається із дозволу деканату за наявності поважних причин (наприклад, лікарняний).

6.4. Відвідування занять

Для здобувачів вищої освіти денної форми відвідування занять є обов'язковим. Поважними причинами для неявки на заняття є хвороба, участь в університетських заходах, академічна мобільність, які необхідно підтверджувати документами. Про відсутність на занятті та причини відсутності здобувач вищої освіти має повідомити викладача або особисто, або через старосту.

За об'єктивних причин (наприклад, міжнародна мобільність) навчання може відбутись в он-лайн формі за погодженням з керівником курсу.

6.5. Політика щодо оскарження оцінювання

Якщо здобувач вищої освіти не згоден з оцінюванням його знань він може опротестувати виставлену викладачем оцінку у встановленому порядку.

6.6. Студентоцентризований підхід

Для врахування інтересів та потреб студентів на початку вивчення курсу здобувачам вищої освіти пропонується відповісти у системі Moodle на низку питань щодо інформаційного наповнення курсу. Відповідно до результатів опитування формується траєкторія навчання з урахуванням потреб студентів.

Під час навчання студенти реалізують своє право вибору індивідуальних завдань практичних завдань.

Наприкінці вивчення курсу та перед початком сесії здобувачам вищої освіти пропонується анонімно заповнити у системі Moodle електронні анкети для оцінки рівня задоволеності методами навчання і викладання та врахування пропозицій стосовно покращення змісту навчальної дисципліни. За результатами опитування вносяться відповідні корективи у робочу програму та силабус.

7. Рекомендовані джерела інформації

1. R. Sedgewick, K. Wayne: Algorithms. Fourth Edition. Addison-Wesley, 2011. - 955 p.
2. Sedgewick R., Wayne K. Algorithms. Part 2. New York: Addison-Wesley, 2014. - 437 p.
3. B. Stroustrup: A Tour of C++ (Second Edition). Addison-Wesley, 2018. - 240 pages.
4. G.L. McDowell: Cracking the Coding Interview. 6th Edition. 189 Programming Questions and Solutions. CareerCup, LLC, Palo Alto, CA. 2016. - 696 p.

5. Data Structure & Algorithms Books Every Programmer Should Read, available at <https://hackernoon.com/10-data-structure-algorithms-books-every-programmer-should-read-d50487313127> (дата звернення: 12.12.2020).
6. Algorithm Design. Foundations, Analysis, and Internet Examples / Michael T. Goodrich and Roberto Tamassia. - N.Y.: John Wiley & Sons, Inc., 2014. - 816 p.
7. Clifford A. Shaffer. Data Structures and Algorithm Analysis. Edition 3.2 (C++ Version). Copyright © 2009-2012 by Clifford A. Shaffer. - 596 p.
8. Clifford A. Shaffer. Data Structures and Algorithm Analysis. Edition 3.2 (Java Version). Copyright © 2009-2013 by Clifford A. Shaffer. - 601 p.
9. Algorithm Design. Foundations, Analysis, and Internet Examples / Michael T. Goodrich and Roberto Tamassia. - N.Y.: John Wiley & Sons, Inc., 2014. - 816 p.
10. Алгоритми та структури даних : навч. Посібник / .О. Коротєєва. - Львів : Видавництво Львівської політехніки, 2014. - 280 с.
11. Алгоритми та структури даних: Навчальний посібник / В.М.Ткачук. - Івано-Франківськ: Видавництво Прикарпатського національного університету імені Василя Стефаника, 2016. - 286 с.
12. Google C++ . Style Guide, available at <https://google.github.io/styleguide/cppguide.html> (дата звернення: 12.12.2020).
13. The Stanford University C++ Style Guide, available at <https://hownot2code.com/2017/01/18/the-stanford-university-c-style-guide/> (дата звернення: 12.12.2020).