МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ АВТОМОБІЛЬНО-ДОРОЖНІЙ УНІВЕРСИТЕТ

Кафедра комп'ютерних технологій і мехатроніки

«ЗАТВЕРДЖУЮ»
Гарант освітньо-професійної програми
«Програмне забезпечення систем»
першого (бакалаврського) рівня вищої
освіти, завідувач кафедри КТМ, д.т.н.,
професор _____ Нікопов О.Я.
«03» 09 2019 р.

СИЛАБУС
АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ / ALGORITHMS AND DATA STRUCTURES SYLLABUS

освітній ступінь	**бакалавр** / bachelor
галузь знань	**12 Інформаційні технології** / Information Technology
спеціальність	**121 Інженерія програмного забезпечення** / Software Engineering
освітня програма	**Програмне забезпечення систем** / Systems Software

Харків 2019

Автор: Подоляка Оксана Олександрівна, доцент кафедри комп'ютерних технологій і мехатроніки

Силабус розглянуто та затверджено на засіданні кафедри комп'ютерних технологій і мехатроніки, протокол № 18 від «27» червня 2019 р.

СИЛАБУС

АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ /

ALGORITHMS AND DATA STRUCTURES
SYLLABUS

освітній ступінь	**бакалавр** / bachelor
галузь знань	**12 Інформаційні технології /** Information Technology
спеціальність	**121 Інженерія програмного забезпечення /** Software Engineering
освітня програма	**Програмне забезпечення систем /** Systems Software

Вступ

Сучасна методологія програмування передбачає, що запис алгоритму на мові програмування і вибір структур представлення даних заслуговують однакової уваги. Програмістам добре відомо, що рішення про те, як представляти дані, неможливо приймати без розуміння того, які алгоритми будуть до них застосовуватися, і навпаки, вибір алгоритму часто дуже сильно залежить від будови даних, до яких він застосовується. Без знання структур даних і алгоритмів неможливо створити серйозний програмний продукт без якого не може існувати жодна інформаційна система.

1. МЕТА, ЗАВДАННЯ ТА РЕЗУЛЬТАТИ ВИВЧЕННЯ ДИСЦИПЛІНИ

Мета вивчення навчальної дисципліни - формування у майбутніх фахівців сукупності загальнонаукових, інструментальних та професійних компетенцій, що забезпечують професійне вирішення інженерних задач, пов'язаних з використанням лінійних та нелінійних структур даних і комп'ютерним моделюванням.

Завдання даного курсу полягає в наступному: познайомити з розмаїттям наявних структур даних, показати, як ці структури реалізовані в мовах програмування, розглянути основні операції, які виконуються над структурами даних, показати особливості підходу до розробки алгоритмів та аналізу їх складності.

Результати вивчення дисципліни

З н а т и:

- базові типи і структури даних мови програмування високого рівня (наприклад, C++, C#);
- архітектуру, системні та функціональні властивості базових алгоритмів та структур даних;
- поняття та особливості використання рекурсії;
- методи аналізу алгоритму та його складності.

В м і т и:

- обирати необхідні структури даних для представлення інформаційних об'єктів;
- розробляти алгоритми, використовуючи схеми, методи і прийоми побудови алгоритмів;
- розв'язувати задачі за допомогою рекурсії;

– виконувати аналіз та визначати часову складність алгоритму.

2. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Характеристика навчальної дисципліни	
	Характеристика навчальної дисципліни	**заочна (дистанційна) форма навчання**
Кількість кредитів - 4 **Кількість годин** - 120	обов'язкова (обов'язкова, вибіркова)	
Семестр викладання дисципліни	3	
Вид контролю:	екзамен (залік, екзамен)	
Розподіл часу:		
- лекції (годин)	16	
- практичні, семінарські (годин)	32	
- лабораторні роботи (годин)		
- самостійна робота студентів (годин)	42	
- курсовий проект (годин)		
- курсова робота (годин)		
- розрахунково-графічна робота (контрольна робота)		
- підготовка та складання екзамену (годин)	30	

Пререквізити та постреквізити навчальної дисципліни:

- пререквізити: «Основи інформаційних технологій», «Алгоритмізація та програмування», «Об'єктно-орієнтоване програмування»;

- постреквізити: «Кросплатформне програмування», «Математичні методи дослідження операцій», виконання дипломної роботи.

3. ЗМІСТ ДИСЦИПЛІНИ

Зміст навчальної дисципліни відповідає робочій програмі

4. ПЛАН ВИВЧЕННЯ ДИСЦИПЛІНИ

Результати навчання	**Навчальна діяльність**	**Робочий час студента (год.)**	**Оцінювання (бал.)**
1	**2**	**3**	**4**
Тема 1. Вступ. Поняття алгоритму та його властивості.			
Знати: властивості алгоритмів; способи запису алгоритмів	**Лекція 1.** Вступ. Поняття алгоритму та його властивості. *План лекції:* 1. Вступ. 2. Поняття алгоритму 3. Властивості алгоритмів. 4. Способи запису алгоритмів. **Список рекомендованих джерел:** Основний: 1, 2, 3 Додатковий: 1, 3 Інтернет-ресурси: 1, 2, 3	2	
	Завдання для самостійної роботи: Самостійне опрацювання лекційного матеріалу та джерел, зазначених у списку. *Питання, винесені на самостійне опрацювання:* 1. Використання псевдокоду та UML-діаграм для запису алгоритму.	4	5
Тема 2. Поняття сортування та його властивості. Прості методи сортування.			
Знати: властивості алгоритмів сортування, методи сортування бульбашкою, вибором, вставками, злиттям, особливості реалізації **Вміти:** виконувати реалізацію алгоритмів сортування на мові C#.	**Лекція 2.** Поняття сортування та його властивості. Прості методи сортування. *План лекції:* 1. Поняття сортування. 2. Властивості алгоритмів сортування. 3. Методи сортування бульбашкою, вибором, вставками, злиттям. 4. Особливості реалізації на мові C#. **Список рекомендованих джерел:** Основний: 1, 2, 3 Додатковий: 1, 3 Інтернет-ресурси: 1, 2, 3	2	
	Практичне заняття 1. Алгоритми сортування вставками, бульбашкою, вибором. Аналіз найгіршого та найкращого випадків. *Мета роботи*: реалізація та аналіз алгоритмів сортування на мові C#. *Завдання*: 1. Реалізувати алгоритми сортування	4	10

1	2	3	4
	вставками, бульбашкою, вибором. 2. Вивести на екран результати кожної ітерації. 3. Проаналізувати роботу алгоритмів для неупорядкованого, частково упорядкованого та упорядкованого масивів.		
	Завдання для самостійної роботи: Самостійне опрацювання літературних джерел, які зазначені у списку. *Питання, винесені на самостійне опрацювання:* 1. Базові алгоритми сортування і пошуку даних.	4	5
Тема 3. Складність алгоритмів. Математичні основи аналізу алгоритмів.			
Знати: поняття часової складності та ефективності алгоритмів; правила аналізу алгоритмів; класи складності алгоритмів; класи складності задач. **Вміти:** визначати часову складність алгоритму	**Лекція 3.** Складність алгоритмів. Математичні основи аналізу алгоритмів. План лекції: 1. Визначення поняття часової складності алгоритму та ефективності. 2. Асимптотичний аналіз складності алгоритму. Поняття про О-нотацію. 3. Правила аналізу алгоритмів. Властивості асимптотичних функцій. 4. Класи складності алгоритмів. 5. Класи складності задач. **Список рекомендованих джерел:** Основний: 1,2,3,4 Додатковий: 1,2,3 Інтернет-ресурси: 1, 2, 3	2	
	Практичне заняття 2. Сортування злиттям. Аналіз складності алгоритму. *Мета роботи*: реалізація та аналіз алгоритму сортування злиттям на мові C#. *Завдання*: 1. Реалізувати алгоритм сортування злиттям. 2. Вивести на екран результати кожної ітерації. 3. Проаналізувати роботу алгоритму для неупорядкованого, частково упорядкованого та упорядкованого масивів. 4. Довести, що складність алгоритму становить O (NlogN)	4	10
	Завдання для самостійної роботи: Самостійне опрацювання літературних джерел, які зазначені у списку. *Питання, винесені на самостійне опрацювання:* 1. Нижня межа часу сортування і як її перевищити. 2. Метод швидкого сортування.	4	5

1	2	3	4
Тема 4. Класифікація структур даних.			
Знати: загальну класифікацію структур даних, особливості застосування **Вміти**: обрати відповідні структури даних для розв'язання певних задач	**Лекція 4.** Класифікація структур даних. План лекції: 1. Загальна класифікація структур даних. 2. Особливості використання. 3. Переваги і недоліки. **Список рекомендованих джерел:** Основний: 1-4 Додатковий: 1-3 Інтернет-ресурси: 1-3	2	
	Завдання для самостійної роботи: Самостійне опрацювання літературних джерел, які зазначені у списку. *Питання, винесені на самостійне опрацювання:* 1. Файлові структури даних	4	5
Тема 5. Зв'язані списки та хеш-таблиці.			
Знати: властивості списків та переваги та недоліки їх використання; особливості побудови та використання хеш-таблиць **Вміти**: реалізовувати та використовувати динамічні структури даних однозв'язний та двозв'язний списки.	**Лекція 5.** Зв'язані списки та хеш-таблиці. План лекції: 1. Поняття списку. Класифікація списку. 2. Особливості використання. 3. Особливості реалізації. 4. Переваги і недоліки лінійних списків. 5. Поняття хеш-таблиці та особливості реалізації. 6. Використання хеш-функцій. **Список рекомендованих джерел:** Основний: 1-8 Додатковий: 1, 2, 3 Інтернет-ресурси: 1, 2, 3	2	
	Практичне заняття 3. Побудова та використання класів Однозв'язний лінійний список та Двозв'язаний лінійний список. *Мета роботи*: реалізація та використання класів однозв'язний та двозв'язаний лінійні списки. *Завдання*: 1. Виконати реалізацію класу однозв'язний лінійний список. 2. Виконати реалізацію методів класу, що дозволяють: вставити елемент на початок списку, в кінець списку, на задану позицію та інших методів, перелічених у методичних вказівках до практичної роботи . 3. Вивести елементи списку у зворотному порядку. 4. Аналогічні завдання виконати для двозв'язного списку. 5. Виконати сортування списку методом бульбашки без використання масивів.	8	10
	Завдання для самостійної роботи: Самостійне	6	5

1	2	3	4
	опрацювання літературних джерел, які зазначені у списку. *Питання, винесені на самостійне опрацювання:* 1. Хеш-таблиці. Алгоритми виключення колізій. 2. Методи, реалізовані у класі Hashtable C#		
Тема 6. Стеки та черги.			
Знати: особливості структур даних стек, черга, дек; особливості реалізації операцій у цих структурах даних. Вміти: реалізовувати класи стеку, черги та деку; ефективно застосовувати напівстатичні структури даних	**Лекція 6.** Стеки та черги. План лекції: 1. Стек. Поняття, використання, операції, особливості реалізації. 2. Черга. Поняття, використання, операції, особливості реалізації. 3. Дек. Поняття, використання, особливості реалізації. **Список рекомендованих джерел:** Основний: 1-6, 8 Додатковий: 1, 2, 3 Інтернет-ресурси: 1, 2, 3	2	
	Практичне заняття 4. Побудова та використання класів Стек та Черга. *Мета роботи*: реалізація та використання класів стек та черга. *Завдання*: 1. Виконати реалізацію класу стек. 2. Виконати реалізацію класу Черга на основі масиву і на основі списку. 3. Реалізувати наступні методи класу: a. додавання k однакових елементів; b. додавання елементів від a до b з кроком n; c. видалення k елементів з черги; d. видалення кожного k-того елементу з черги e. видалення елементів, що поторюються. 4. Розвернути чергу у зворотному порядку, використовуючи стек. 5. Скласти чергу з кожного k-того елементу даної черги. 6. Розбити чергу на дві частини і поміняти їх місцями. 7. Створити дві черги і знайти їх поєднання та перетинання (елементи, що повторюються, видалити).	6	10
	Завдання для самостійної роботи: Самостійне опрацювання літературних джерел, які зазначені у списку. *Питання, винесені на самостійне опрацювання:* 1. Стеки, черги та деки. Особливості реалізації на C#	6	5

1	2	3	4
Тема 7. Дерева та графи. Обхід в глибину та в ширину.			
Знати: особливості динамічних структур даних таких як бінарна купа, бінарне дерево пошуку, граф **Вміти**: виконувати реалізацію класів бінарна купа та бінарне дерево; використовувати методи класів бінарна купа та бінарне дерево для побудови таких структур даних з масивів, реалізовувати пошук інформації в таких структурах даних	**Лекція 7.** Дерева та графи. Обхід в глибину та в ширину. План лекції: 1. Графи основні поняття. 2. Бінарна купа. Властивості, особливості реалізації. 3. Бінарне дерево пошуку. Властивості, особливості реалізації. 4. Обходи бінарного дерева в ширину та глибину. **Список рекомендованих джерел:** Основний: 2, 6 Додатковий: 2, 3 Інтернет-ресурси: 1, 2, 3	2	
	Практичне заняття 5. Побудова бінарної купи, бінарного дерева. *Мета роботи*: Реалізація та використання структур даних бінарна купа та бінарне дерево. *Завдання*: 1. Виконати реалізацію класу Бінарна купа. 2. Реалізувати метод сортування Heapsort. 3. Виконати реалізацію класу Бінарне дерево пошуку. 4. Виконати реалізацію методів обходу бінарного дерева в глибину і ширину.	6	10
	Завдання для самостійної роботи: Самостійне опрацювання літературних джерел, які зазначені у списку. *Питання, винесені на самостійне опрацювання:* 1. Метод сортування Heapsort 2. Сортування деревом Tree sort 3. Червоно-чорні дерева та їх властивості.	8	5
Тема 8. Алгоритми з поверненням. Розв'язання задач за допомогою рекурсії.			
Знати: поняття алгоритмів з поверненням; закони рекурсії **Вміти**: Реалізовувати рекурсивні алгоритми для розв'язання задач.	**Лекція 8.** Алгоритми з поверненням. Розв'язання задач за допомогою рекурсії. План лекції: 1. Поняття рекурсії та де вона використовується. 2. Закони рекурсії. 3. Приклади. **Список рекомендованих джерел:** Основний: 6, 7, 8 Додатковий: 2, 3 Інтернет-ресурси: 1, 2, 3	2	
	Практичне заняття 6. Рекурсивні алгоритми. *Мета роботи*: Реалізація та використання рекурсивних алгоритмів. *Завдання*:	4	10

1	2	3	4
	1. Виконати пошук максимального елемента масиву. 2. Виконати пошук елемента в упорядкованому масиві (двійковий пошук). 3. Задане натуральне число n>0. Перевірте, чи є воно простим, вибравши за параметр рекурсії дільник числа.		
	Завдання для самостійної роботи: Самостійне опрацювання літературних джерел, які зазначені у списку. *Питання, винесені на самостійне опрацювання:* 1. Алгоритм Євкліда 2. Алгоритм пошуку факторіалу числа	6	5
Підсумковий контроль: іспит		30	
Разом:		**120** ***годин/ 4 кредити***	**100** ***балів***

5.СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

Основний

1. Ахо А.В., Хопкрофт Д., Ульман Д.Д. Структуры данных и алгоритмы. – М.: Вильямс, 2001. – 384 с.
2. Кормен Т.Х., Лейзерсон Ч.И., Ривест Р.Л., Штайн К. Алгоритмы: построение и анализ. – 2-е изд. – М.: Вильямс, 2005. – 1296 с.
3. Левитин А.В. Алгоритмы: введение в разработку и анализ. – М.: Вильямс, 2006. – 576 с.
4. Седжвик Р. Фундаментальные алгоритмы на С++. Анализ/Структуры данных/Сортировка/Поиск. – К.: ДиаСофт, 2001. – 688 с.
5. Скиена С.С. Алгоритмы. Руководство по разработке. – 2-е изд. – СПб: БХВ, 2011 – 720 с.
6. Макконнелл Дж. Основы современных алгоритмов. – 2е изд. – М.: Техносфера, 2004. – 368 с.
7. Миллер Р. Последовательные и параллельные алгоритмы: общий подход. – М.: БИНОМ, 2006. – 406 с.
8. Бакнелл Д.М. Фундаментальные алгоритмы и структуры данных в Delphi. – СПб.: ДиаСофт, 2003. – 560 с.

Додатковий

1. Томас Х. Кормен Алгоритмы: вводный курс Томаса Х. Кормена. - М: Вильямс, 2014. - 208 с.

2. Генри С. Уоррен Алгоритмические трюки для программистов, 2-е издание. – М: Вильямс, 2013. – 512 с.

3. Дональд Э. Кнут Искусство программирования, том 1. Основные алгоритмы, 3-е изданиею – М: Вильямс, 2015. – 720 с.

Інформаційні ресурси

1. https://metanit.com/sharp/algoritm
2. https://tproger.ru/translations/algorithms-and-data-structures
3. https://proglib.io/p/data-structure-algorithms