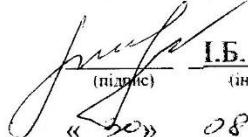


Міністерство освіти і науки України
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Кафедра інженерії програмного забезпечення (№ 603)

ЗАТВЕРДЖУЮ

Керівник проектної групи/


(підпис) І.Б. Туркін
(ініціали та прізвище)
«30» 08 2019 р.

**РОБОЧА ПРОГРАМА ОБОВ'ЯЗКОВОЇ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Алгоритми та структури даних
(назва навчальної дисципліни)

Галузь знань: 12 «Інформаційні технології»
(шифр і найменування галузі знань)

Спеціальність: 121 «Інженерія програмного забезпечення»
(код та найменування спеціальності)

Освітня програма: «Інженерія програмного забезпечення»
(найменування освітньої програми)

Форма навчання: денна

Рівень вищої освіти: перший (бакалаврський)

Харків 2019 рік

Робоча програма «Алгоритми та структури даних» для студентів за спеціальністю: 121 «Інженерія програмного забезпечення» освітньою програмою «Інженерія програмного забезпечення»

«20» 04 2019 р. – 12 с.

Розробник: Данова М. О., доц. кафедри №603, канд. техн. наук, доц.  (прізвище та ініціали, посада, науковий ступінь та вчене звання) (підпис)

Робочу програму розглянуто на засіданні кафедри інженерії програмного забезпечення

Протокол № 1 від «30» 08 2019 р. (назва кафедри)

Завідувач кафедри д-р техн. наук., проф.  (науковий ступінь і вчене звання) (підпис) І.Б. Туркін (ініціали та прізвище)

1. Опис навчальної дисципліни

Найменування показника	Галузь знань, спеціальність, освітня програма, рівень вищої освіти	Характеристика навчальної дисципліни (денна форма навчання)
Кількість кредитів – 6	<p>Галузь знань <u>12 «Інформаційні технології»</u> <small>(шифр і найменування)</small></p> <p>Спеціальність <u>121 «Інженерія програмного забезпечення»</u> <small>(код і найменування)</small></p> <p>Освітня програма <u>«Інженерія програмного забезпечення»</u> <small>(найменування)</small></p> <p>Рівень вищої освіти: перший (бакалаврський)</p>	Цикл професійної підготовки (1.2. Дисципліни загально-професійної підготовки)
Кількість модулів – 2		Навчальний рік
Кількість змістовних модулів – 4		2019/2020
Індивідуальне завдання: <u>розрахунково-графічна робота</u>		Семестр
Загальна кількість годин – 80/180		<u>4</u> -й
Кількість тижневих годин для денної форми навчання: аудиторних – 5 самостійної роботи студента – 6,25		Лекції*
		<u>40</u> годин
		Практичні, семінарські*
		<u> </u> годин
		Лабораторні*
	<u>40</u> годин	
Самостійна робота		
<u>100</u> годин		
Вид контролю		
модульний контроль, іспит		

Співвідношення кількості годин аудиторних занять до самостійної роботи становить: 80/100.

*Аудиторне навантаження може бути зменшене або збільшене на одну годину залежно від розкладу занять.

2. Мета та завдання навчальної дисципліни

Мета вивчення: придбання студентами знань з основних принципів конструювання алгоритмів за допомогою абстрактних операцій та їх аналіз. Розробка лінійних та нелінійних структур даних засобами алгоритмічних мов. Опис даних на абстрактному, логічному та фізичному рівні.

Завдання: вивчення впливу структур даних на ефективність виконання алгоритму, а також методів оцінки алгоритмів.

Результати навчання: студент має:

знати :

- методи побудови алгоритмів;
- фундаментальні алгоритми (сортування, пошуку, на графах);
- базові структури даних, абстрактні структури даних.

вміти: вибирати для конкретної задачі структури даних та мовні конструкції, що забезпечують можливість побудови ефективних алгоритмів.

мати уявлення: про використання середовища Visual Studio 10.0 для створення, редагування та налагоджування Windows додатків.

Міждисциплінарні зв'язки: дисципліні передують курси: «Дискретні структури», «Основи програмування», «Програмування мовою С#», «Об'єктно-орієнтоване програмування»; дисципліна потрібна для подальшого вивчення курсів «Конструювання програмного забезпечення», «Web програмування мовою PHP», «Архітектура та проектування програмного забезпечення .Net».

3. Програма навчальної дисципліни

Модуль 1

Змістовий модуль 1. Базові структури даних

Тема 1. Вступ до дисципліни «Алгоритми та структури даних».

Об'єкти обробки даних: користувачі, носії, сховища та ЕОМ. Рівні опису структур даних: абстрактний (функціональна специфікація), логічний рівень та фізичний. Моделі даних. Моделі пам'яті. Типи даних у мовах програмування. Структура даних як об'єднання типу даних і моделі пам'яті.

Тема 2. Лінійні списки. Приклади лінійних списків.

Визначення поняття "лінійний список". Стек, черга і дек. Послідовне та зв'язне представлення лінійного списку у пам'яті. Стек з послідовним розміщенням. Функціональна специфікація стека. Властивості стека. Опис стека на логічному рівні. Недоліки послідовного розміщення. Черга з послідовним розміщенням. Функціональна специфікація черги. Властивості черги. Опис черги на логічному рівні. Реалізація черги в послідовній пам'яті в формі класу. Реалізація методів черги на мові С#. Недоліки послідовної черги. Способи усунення недоліків. Приклад включення нового елемента в лінійний список. Зв'язне розміщення списків. Керована послідовна пам'ять (КПП). Переваги КПП. Приклад розміщення в КПП стеків і черг. Реалізація стека в динамічній пам'яті. Приклад включення нового елемента в лінійний список після елемента з номером k (на ПСЕВДОКОДІ).

Фрагменти операцій включити (виключити) елемент в динамічний лінійний список. Приклад включення нового елемента в лінійний список з бар'єром.

Тема 3. Інші види лінійних списків.

Циклічні (симетричні) списки. Фрагменти операцій на циклічних списках. Обробка усіх елементів циклічного списку. Лінійні списки що мають два покажчика. Циклічні списки що мають два покажчика. Списки підсписків. Використання головного елемента в списках підсписків. Топологічне сортування. Приклади топологічного сортування. Алгоритм топологічного сортування. Деталізація пунктів алгоритму. Вибір основного елемента структури, реалізації основного списку елементів та черги вихідних елементів. Реалізація алгоритму на ПСЕВДОКОДІ.

Тема 4. Лінійні списки в C#.

Опис типів с параметрами в C#. Використання стека без параметра (упаковка, розпаковка). Приклад стека с параметром. Переваги стека с параметром. Обмеження на параметри. Основні методи класів (Stack<T>, Queue<T>, LinkedList<T> та LinkedListNode<T>) с параметром. Приклад динамічного масива с параметром. Інтерфейси. Відмінність інтерфейсів від абстрактних класів. Опис інтерфейсів. Використання інтерфейсів при описі класів Операції as, is. Використання інтерфейсів в колекціях C#. Інтерфейси IEnumerator, IEnumerable. Оператор foreach. Приклад використання списку LinkedList.

Тема 5. Нелінійні структури.

Приклади не лінійних структур. Визначення поняття "дерево". Приклади дерев. Бінарні дерева. Алгоритм перетворення m-арного дерева в бінарне. Типовий вузол (елемент) бінарних дерев. Основні операції з бінарними деревами. Обхід вузлів дерева в прямому, зворотному і кінцевому порядку. Прямий обхід дерева – рекурсивний варіант. Зворотний обхід дерева – ітераційний варіант. Операції з деревами. Збалансовані дерева. Алгоритм створення збалансованого дерева. Використання дерев для пошуку інформації Алгоритм пошуку в бінарнім дереві (з бар'єром та без бар'єра). Пошук в дереві з включенням. Алгоритм виключення елемента з бінарного дерева. Прошиті дерева Алгоритм зворотного обходу прошого дерева не рекурсивний варіант. Послідовне розміщення дерев. Пряме послідовне розміщення дерев. Фамільне послідовне розміщення дерев. Алгоритм зворотного перегляду елементів дерева для прямого послідовного розміщення. Структури складних списків. Визначення поняття "складний список". Розміщення складних списків в пам'яті ЕОМ. Приклад складного списку. Складні списки з головними елементами.

Модульний контроль

Модуль 2

Змістовий модуль 1. Аналіз алгоритму

Тема 1. Оцінка ефективності та трудомісткості алгоритму.

Визначення алгоритму. Характеристики алгоритму: існування алгоритму, можливість реалізації на ЕОМ, обсяг даних, складність алгоритму. Напрями

досліджень в теорії алгоритмів. Розробка алгоритмів. Аналіз алгоритмів на прикладі сортування включенням та сортування злиттям. Оцінка ефективності алгоритму. Асимптотичні оцінки функцій зростання. Приклади функцій зростання. Вплив подвійного зростання розміру задач на час виконання. Оцінка трудомісткості алгоритму. Визначення трудомісткості. Класифікація алгоритмів на основі функції трудомісткості алгоритму. Методика аналізу основних алгоритмічних конструкцій. Ресурсна ефективність алгоритму.

Модульний контроль

Змістовий модуль 2. Основні обчислювальні алгоритми

Тема 1. Алгоритми сортування.

Визначення сортування. Алгоритми внутрішнього та зовнішнього сортування. Характеристики сортування. Групи сортувань. Сортування вибором: лінійний вибір, вибір з обміном, вибір з підрахунком. Сортування обміном: парний обмін, стандартний обмін, просівання, швидке сортування. Сортування включенням: просте включення, включення з обміном, включення в двох напрямках, сортування Шелла. Сортування розподілом - окремий випадок сортування вибором з підрахунком. Сортування розподілом для прикладу з числовими ключами. Зовнішні сортування. Операція злиття файлів. Приклад алгоритма злиття файлів. Сортування злиття з трьома файлами. Сортування злиттям для чотирьох файлів (1-3,3-1). Сортування злиттям по симетричній схемі (2-2,2-2). Багатофазне сортування. Приклад багатофазного сортування ($N=129$, $p=6$). Числа Фібоначчі з порядком p . Початкове розподілення в багатофазнім сортуванні. Каскадне сортування. Особливості метода. Приклад послідовності злиття файлів для ($N=190$, $p=6$). Рекурентні відношення для каскадного сортування.

Тема 2. Пошук в масивах.

Загальне визначення масиву. Приклади масивів. Загальний простий довідник. Простий довідник зі зв'язками. Довідник підписків з одним рівнем. Час пошуку в простім довіднику та довіднику з підписками. Довідник підписків з багатьма рівнями. Відображення. Відображення та їх властивості. Приклади функцій відображення. Види підписків. Переповнення підписків. Переповнення в довідниках з одним рівнем. Переповнення в довідниках підписків з багатьма рівнями. Переповнення в відображенням. Лінійний спосіб усунення колізій, спосіб відкритої адресації. Алгоритм включення запису в масив підписків з одним записом. Таблиця порівняльного аналізу рішення проблеми переповнення для відображень.

Тема 3. Пошук в бінарних деревах.

Збалансоване бінарне дерево. AVL – дерево. Збалансоване AVL – дерево. Використання поворотів для усунення розбалансування дерева. RB – дерево. Властивості RB – дерева. Повороти в RB – дереві. Алгоритм повороту.

Тема 4. Обхід графів.

Способи представлення графів. Обхід графів в глибину. Обхід графів в ширину.

Модульний контроль

Змістовий модуль 3. Рекурсія.

Тема 1. Рекурентні послідовності та рекурсивні алгоритми.

Базові поняття та визначення. Приклади рішення обчислювальних задач заданих рекурентними відношеннями. Рекурсивно-задані послідовні функції. Класифікація рекурсивно-заданих послідовностей та функцій. Основні визначення. Методи дослідження та рішення рекурентних відношень. Методи асимптотичної оцінки функцій заданих рекурсивними відношеннями (використання мажоритарної функції або метода підстановки). Окремий випадок знаходження точного рішення. Рекурсивні алгоритми. Види рекурсивних алгоритмів: пряма рекурсія, непряма рекурсія, двічі рекурсивні функції.

Модульний контроль

4. Структура навчальної дисципліни

Назви змістового модуля і тем	Кількість годин				
	усього	у тому числі			
		л	п	лаб	с.р.
1	2	3	4	5	6
Модуль 1					
Змістовий модуль 1. Базові структури даних					
Тема 1. Вступ до дисципліни «Алгоритми та структури даних»	2	2	-	-	-
Тема 2. Лінійні списки	15	4	-	6	5
Тема 3. Другі види лінійних списків	15	2	-	6	7
Тема 4. Лінійні списки в C#	21	4	-	8	9
Тема 5. Нелінійні структури	26	6	-	10	10
Модульний контроль	2	-	-	-	2
Разом за змістовим модулем 1	84	18	-	30	33
Усього годин	81	18	-	30	33
Модуль 2					
Змістовий модуль 1. Аналіз алгоритму					
Тема 1. Оцінка ефективності та трудомісткості алгоритму	10	4	-	2	4
Модульний контроль	2	-	-	-	2
Разом за змістовим модулем 1	12	4	-	2	6
Змістовий модуль 2. Пошук в масивах					
Тема 1. Алгоритми сортування.	24	6	-	8	10
Тема 2. Пошук в масивах	13	4	-	-	9
Тема 3. Пошук в бінарних деревах	6	2	-	-	4
Тема 4. Обхід графів	4	2	-	-	2

1	2	3	4	5	6
Модульний контроль	2	-	-	-	2
Разом за змістовим модулем 2	49	14	-	8	27
Змістовий модуль 3. Рекурсія					
Тема 1. Рекурентні послідовності та рекурсивні алгоритми	8	4	-	-	4
Модульний контроль	2	-	-	-	2
Разом за змістовим модулем 3	10	4	-	-	6
Усього годин	71	21	-	10	39
Індивідуальне завдання	20	-	-	-	20
Контрольний захід	8	-	-	-	8
Усього годин	180	40	-	40	100

5. Теми семінарських занять

№ п/п	Назва теми	Кількість годин
1		
2		
	Разом	

6. Теми практичних занять

№ п/п	Назва теми	Кількість годин
1		
2		
	Разом	

7. Теми лабораторних занять

№ п/п	Назва теми	Кількість годин
1	Базові динамічні структури. Списки зі зв'язками.	8
2	Лінійні динамічні структури. Циклічні списки.	6
3	Використання вбудованих типів стек та черга з параметром та без параметра	10
4	Нелінійні динамічні структури. Бінарні дерева.	10
5	Сортування масивів.	6
	Разом	40

8. Самостійна робота

№ з/п	Назва теми	Кількість годин
1	Лінійні списки	5
2	Другі види лінійних списків	7
3	Лінійні списки в C#.	9
4	Нелінійні структури	9
5	Оцінка ефективності та трудомісткості алгоритму	4
6	Алгоритми сортування	10
7	Пошук в масивах	9
8	Пошук в бінарних деревах	4
9	Обхід графів.	2
10	Рекурентні послідовності та рекурсивні алгоритми	5
	Разом	64

9. Індивідуальне завдання

Виконання розрахункової роботи на тему «Аналіз продуктивності лінійних списків. Дослідження продуктивності метода сортування в трьох випадках: кращий, гірший і випадковий»

10. Методи навчання

1. За джерелами придбання знань – словесні: лекція (вступна, традиційна, проблемна, з помилками), бесіда (евристична), диспут, дискусія, робота з друкованими та інтернет-джерелами; наочні: ілюстрація, спостереження; практичні: лабораторна робота, курсовий проект.
2. За характером пізнавальної діяльності тих, хто навчається – інформаційно-репродуктивний, репродуктивний, проблемне викладання, частково-пошуковий.
3. За логікою пізнання – індуктивний, дедуктивний, аналогій, вивідних знань.
4. Методи перевірки й оцінки знань, умінь, навичок: спостереження, усне опитування, контрольні роботи, програмований контроль, тестування (традиційне та машинне).

11. Методи контролю

1. Опитування.
2. Лабораторні роботи.
3. Модульні контрольні роботи.
4. Індивідуальна розрахункова робота
5. Форма підсумкового контролю успішності навчання: іспит (письмово)

12. Критерії оцінювання та розподіл балів, які отримують студенти

12.1. Розподіл балів, які отримують студенти (кількісні критерії оцінювання)

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовний модуль 1			
Робота на лекціях			
Робота на практичних заняттях			
Виконання і захист лабораторних (практичних) робіт	6...10	4	24...40
Модульний контроль	6...10	1	6...10
Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовний модуль 1			
Робота на лекціях			
Робота на практичних заняттях			
Виконання і захист лабораторних (практичних) робіт			
Модульний контроль	6...10	1	6...10
Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовний модуль 2			
Робота на лекціях			
Робота на практичних заняттях			
Виконання і захист лабораторних (практичних) робіт	6...10	1	6...10
Модульний контроль	6...10	1	6...10
Змістовний модуль 3			
Робота на лекціях			
Робота на практичних заняттях			
Виконання і захист лабораторних (практичних) робіт			
Модульний контроль	6...10	1	6...10
Виконання і захист РГР (РР, РК)	6..10	1	6..10
Усього за семестр			60...100

Семестровий контроль (іспит) проводиться у разі відмови студента від балів поточного тестування й за наявності допуску до іспиту. Під час складання семестрового іспиту студент має можливість отримати максимум 100 балів.

Білет для іспиту/заліку складається з двох теоретичних питань (кожне питання 35 балів) та одного практичного питання (питання оцінюється в 30 балів).

12.2. Якісні критерії оцінювання

Необхідний обсяг знань для одержання позитивної оцінки: методи побудови алгоритмів; фундаментальні алгоритми (сортування, пошуку, на графах); базові структури даних, абстрактні структури даних.

Необхідний обсяг вмінь для одержання позитивної оцінки: вибирати для конкретної задачі структури даних та мовні конструкції, що забезпечують можливість побудови ефективних алгоритмів.

12.3 Критерії оцінювання роботи студента протягом семестру

Задовільно (60-74). Показати мінімум знань та умінь. Здати індивідуальне завдання та основні лабораторні. Здати тестування. Знати методи побудови алгоритмів; фундаментальні алгоритми (сортування, пошуку, на графах); базові структури даних. Вміти вибирати для конкретної задачі структури даних, що забезпечують можливість побудови алгоритмів.

Добре (75-89). Твердо знати мінімум, здати всі лабораторні роботи, індивідуальне завдання та тестування. Досконало знати методи побудови алгоритмів; фундаментальні алгоритми (сортування, пошуку, на графах); базові структури даних, абстрактні структури даних. Досконало вміти вибирати для конкретної задачі структури даних та мовні конструкції, що забезпечують можливість побудови ефективних алгоритмів.

Відмінно (90-100). Здати всі контрольні точки з оцінкою «відмінно». Досконально знати всі теми та вміти застосовувати їх.

Шкала оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційований залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	
0 – 59	Незадовільно	Не зараховано

13. Методичне забезпечення

1. Розроблений лекційний курс
(//master/студенти/2019-2020/6 факультет/2курс/Алгоритми та структури даних)
<https://mentor.khai.edu/course/view.php?id=238>
2. Розроблені тестові питання в системі mentor для модульних контрольних робіт
<https://mentor.khai.edu/mod/quiz/view.php?id=11060>
3. Розроблені питання для модульних контрольних робіт
(//master/студенти/2019-2020/6 факультет/2курс/Алгоритми та структури даних)
4. Розроблені питання для підсумкового контролю успішності навчання

(//master/студенти/2019-2020/6 факультет/2курс/ Алгоритми та структури даних /Питання до іспиту)

5. Лабораторні роботи

Мокляк, М.Г., Лучшев, П.О., Дегтярьова, Т.Г., Соколова, Є.В. Алгоритми та структури даних [Електронний ресурс] : навч. посіб. до лаб. робіт. – Харків : Нац. аерокосм. ун-т ім. М. Є. Жуковського «Харків. авіац. ін-т», 2012. – 50 с.

6. Індивідуальні розрахункові роботи (домашні завдання)

Мокляк, М.Г., Лучшев, П.О., Дегтярьова, Т.Г., Соколова, Є.В. Алгоритми та структури даних [Електронний ресурс] : навч. посіб. до вик. розрах. роботи. – Харків : Нац. аерокосм. ун-т ім. М. Є. Жуковського «Харків. авіац. ін-т», 2014. – 20 с.

7. Навчально-методичний комплекс дисципліни розміщено у системі mentor

<https://mentor.khai.edu/course/view.php?id=238>

14. Рекомендована література

Базова

- 1 Мокляк, М.Г., Лучшев, П.О., Дегтярьова, Т.Г., Соколова, Є.В. Алгоритми та структури даних [Електронний ресурс] : навч. посіб. до лаб. робіт. – Харків : Нац. аерокосм. ун-т ім. М. Є. Жуковського «Харків. авіац. ін-т», 2012. – 50 с.
- 2 Мокляк, М.Г., Лучшев, П.О., Дегтярьова, Т.Г., Соколова, Є.В. Алгоритми та структури даних [Електронний ресурс] : навч. посіб. до вик. розрах. роботи. – Харків : Нац. аерокосм. ун-т ім. М. Є. Жуковського «Харків. авіац. ін-т», 2014. – 20 с.
- 3 Н.Вирт. Алгоритмы и структуры данных: Пер.с англ. М., Мир,1989. -360с.
- 4 Т.Кормен, Ч.Лейзерсон, Р.Ривест. Алгоритми: Конструювання та аналіз. М.: МЦНМО, 2000, -960с.
- 5 Ф. Ахо, Д. Хопкрофт, Д. Ульман. Структури даних та алгоритми : пер. з англ. М.: “Вільмс”, 2000. -384 с.
- 6 Д. Кнут. Мистецтво програмування для ЕОМ. Т. 1,2,3, 2003.

Допоміжна

- 1 Р. Седжвнік. Фундаментальные алгоритмы на С++. М. 2001.
- 2 Д.Х. Грин, Д.Э. Кнут. Математические методы анализа алгоритмов.
- 3 Д. Гасфилд. Строки, деревья и последовательности в алгоритмах: Информатика и вычислительная биология / пер. с англ. И.В. Романовского.- СПб.: Невский Диалект, БХВ- Петербург, 2003.- 654с.
- 4 Дж. Макконнелл. Основы современных алгоритмов. 2 – дополненное издание Москва: Техносфера, 2004.-368с.

15. Інформаційні ресурси

- 1 Колдаев В.Д. Структуры и алгоритмы обработки данных: учеб. пособие [Электронный ресурс] – Режим доступа: <http://www.znaniium.com/bookread.php?book=418290>