



ХАРКІВСЬКА ДЕРЖАВНА АКАДЕМІЯ КУЛЬТУРИ

Кафедра інформаційних технологій

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Алгоритмізація і структури даних
(шифр і назва навчальної дисципліни)

перший рівень

(назва рівня вищої освіти)

галузь знань

12 Інформаційні технології

(код і назва спеціальності)

спеціальність

126 Інформаційні системи та технології

(код і назва спеціальності)

кваліфікація

бакалавр з інформаційних систем та технологій

Харків 2017

Робоча програма Алгоритмізація і структури даних

РОЗРОБЛЕНО ТА ВНЕСЕНО: Харківська державна академія культури

Укладач:Побіженко І.О.

Робоча програма затверджена на кафедрі інформаційних технологій

Протокол від «06» жовтня 2017 року № 2

Завідувач кафедри інформаційних технологій



(підпис)

(Асєєв Г. Г.)
(прізвище та ініціали)

«06» жовтня 2017 року

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни
Кількість кредитів – 4	Галузь знань <u>12 Інформаційні технології</u>	Обов'язкова
	Напрямок підготовки <u>126 Інформаційні системи та технології</u>	
Модулів – 1	Спеціальність (професійне спрямування): <u>126 Інформаційні системи та технології</u>	Рік підготовки:
Змістових модулів – 3		2-й
Індивідуальне науково-дослідне завдання		Семестр
Загальна кількість годин – 120		4-й
Тижневих годин для денної форми навчання: аудиторних – 4 год; самостійної роботи студента – 3,5 год.	Освітньо-кваліфікаційний рівень: бакалавр	Лекції
		40 год.
		Семінарські
		-
		Практичні
		11 год.
		Самостійна робота
		69 год.
		Індивідуальні завдання:
		-
Вид контролю:		
екзамен		

Примітка.

Співвідношення кількості годин аудиторних занять до самостійної роботи становить 51:69.

2. Мета та завдання навчальної дисципліни

«Алгоритмізація і структури даних» – навчальна дисципліна, яка в контексті новітніх технологій обробки даних посилює теоретичну та практичну професійну підготовку бакалаврів з інформаційних технологій.

Мета дисципліни: формування знання про різноманітність структур даних, області їх використання, способи їх програмної обробки; формування умінь і навичок програмно обробляти статичні і динамічні дані з використанням різних методів та алгоритмів, у т.ч. розв'язування задач на пошук, сортування, обробку динамічних структур тощо.

Задачі дисципліни: засвоєння основних математичних методів, ефективних алгоритмів і методів реалізації функцій інформаційних систем і технологій в прикладних областях; оволодіння навичками самостійного використання сучасних методів побудови та аналізу ефективних алгоритмів та їх реалізації в конкретних застосуваннях.

Компетентності, якими повинен оволодіти здобувач	Програмні результати навчання
<p>ЗК2. Здатність застосовувати знання у практичних ситуаціях.</p> <p>ЗК3. Знання та розуміння предметної області та професійної діяльності.</p> <p>ЗК5. Навички використання інформаційних і комунікаційних технологій</p> <p>ФК1. Здатність проводити аналіз об'єкту проектування та предметної області</p> <p>ФК3. Здатність до проектування системного, комунікаційного і прикладного програмного забезпечення, технічних засобів та комунікаційних й інформаційних технологій, мереж та систем.</p> <p>ФК4. Здатність розробляти засоби реалізації ІСТ та ІСДС (методичні, інформаційні, алгоритмічні, технічні й програмні).</p> <p>ФК5. Здатність розробляти, налагоджувати та вдосконалювати програмне забезпечення комп'ютерно-інтегрованих систем.</p> <p>ФК6. Здатність використовувати сучасні технології проектування в розробці алгоритмічного та</p>	<p>РН2. Здатність використовувати знання з основних фундаментальних, природничих та загально-інженерних дисциплін, а також системного аналізу, моделювання систем, теорії алгоритмів та дискретної математики при розв'язанні типових задач, проектуванні та використанні ІСТ та ІСДС.</p> <p>РН3. Здатність використовувати: базові знання інформатики й сучасних ІСТ, навички програмування та застосування програмних засобів, безпечної роботи в комп'ютерних мережах, уміння створювати бази даних, використовувати інтернет-ресурси та демонструвати уміння розробляти алгоритми та комп'ютерні програми на мовах високого рівня та технологій об'єктно-орієнтованого програмування для реалізації задач проектування та використання ІСТ та ІСДС.</p> <p>РН4. Здатність проводити системний аналіз об'єктів проектування та обґрунтовувати вибір структури,</p>

<p>програмного забезпечення ІСТ та ІСДС.</p>	<p>алгоритмів та способів циркулювання інформації в ІСТ та спеціалізації ІСДС. РН6. Здатність демонструвати знання сучасного рівня та новітніх технологій ІСТ та ІСДС з метою їх запровадження у професійної діяльності РН9. Здатність демонструвати знання і практичні навички програмування та використання прикладних і спеціалізованих комп'ютерних систем та середовищ для розв'язання задач проектування</p>
--	--

В результаті вивчення навчальної дисципліни студент повинен:

знати:

основні принципи теорії алгоритмів та структур даних; - різні структури даних та методи їх обробки засобами різних мов програмування; - методику розробки алгоритмів

вміти:

складати алгоритм задачі та окремих її етапів, логічну схему програми визначати основні етапи повної побудови алгоритмів• визначати можливості використання готових алгоритмів розв'язку задач, розроблених раніше• розробляти структурні алгоритми розв'язання прикладних задач, на підставі базових• алгоритмічних структур, використовуючи алгоритмічні мови програмування, програмне забезпечення комп'ютерів описувати алгоритми• організовувати структури даних• визначати можливості використання готових алгоритмів розв'язку задач, розроблених раніше• визначати правильність алгоритму•

мати навички:

– вибору методів та сучасних програмних засобів розробки програм.

Міждисциплінарні зв'язки: для засвоєння матеріалу використовуються знання, отримані при вивченні курсів «Інформаційні системи та технології», «Основи програмування», «Вища математика», «Комп'ютерна дискретна математика»/

Змістовий модуль 1. Структури даних

Тема 1. Структури та об'єднання.

1.1. Структури.

Структури, оголошення та визначення об'єктів типу структура. Відмінності структур мов С та С++. Методи в структурах. Статичні члени структури. Структури з бітовими полями. Вкладені структури. Доступ до елементів структур. Операції з структурами. Структури як параметри функцій. Структури як значення функції, що повертається.

Об'єднання. Операції з об'єднаннями. Переліки. Змінні структури.

1.2. Масиви структур.

Масиви структур. Показчики на структури. Передача по посиланню масивів структур.

Тема 2. Введення в систему вводу-виводу С++. Файлові структури даних

2.1. Система вводу-виводу С++.

Базові положення системи вводу-виводу С++. Потоки і буфери. Стандартні об'єкти вводу-виводу. Ієрархія класів вводу-виводу С++.

Ввід даних за допомогою глобального об'єкту cin.

Вивід рядків. Введення одного символу. Використовування функції get(): без параметрів, з параметрами. Ввід рядків із стандартного пристрою введення. Використовування функції getline().

Вивід даних за допомогою глобального об'єкту cout. Очищення буфера виводу – flush(). Використовування функцій put() і write(). Ввід – вивід даних, що форматується.

Маніпулятори вводу-виводу. Функції width(), precision(), fill(). Маніпулятори, визначувані користувачем.

2.2. Файлові структури даних.

Використовування файлів для вводу-виводу даних. Створення файлу. Створення потоку. Відкриття потоку. «Приєднання» файлу до потоку. Режими відкриття та доступу до файлів. Обміни з файлом з допомогою потоку. «Від'єднання» потоку від файлу. Закриття файлу. Знищення файлу. Визначення стану потоку. Функції управління вводу-виводу.

Тема 3. Динамічні структури даних

3.1. Загальні поняття про динамічні структури даних.

Визначення, класифікація динамічних структур даних, способи оголошення, ініціалізація динамічних структур, методи доступу до даних динамічних структур, розміщення їх у пам'яті, переваги і недоліки використання динамічних структур в програмах.

3.2. Списки, черги, дерева.

Зв'язні списки, однозв'язні і двонаправлені списки, циклічні списки. Створення однозв'язного списку. Прохід однозв'язного списку. Включення нового елемента в існуючий список. Виключення елемента із списку. Створення двонаправленого списку. Прохід двонаправленого списку. Включення нового елемента в існуючий список. Виключення елемента із списку.

Змістовий модуль 2. Бібліотека шаблонів

Тема 4. Стандартна бібліотека шаблонів. Шаблони функцій

4.1. Поняття шаблону функції.

Основи апарату шаблонів. Шаблони функцій. Перевантаження шаблонів функцій. Шаблони функцій сортування.

4.2. Бібліотека шаблонів STL.

Стандартна бібліотека шаблонів (STL). Призначення та склад STL. Контейнери. Алгоритми. Функціональні об'єкти. Ітератори, ітератори потоку. Робота з векторами,

списками, стеками, чергами.

Тема 5. Стандартний клас String

5.1. Типи даних для роботи з рядками.

Тип даних string. Зв'язок класу string з STL. Оголошення та визначення об'єктів типу string. Методи та властивості класу string. Операції з рядками.

5.2. Функції роботи з рядками. Функції роботи з рядками. Ввід-вивід рядків. Перетворення між string та рядками в стилі C.

Тема 6. Обробка виключень. Особливості стандартів C11, C++11

6.1. Основні принципи обробки виключень в мові програмування. Обробка виключень в мові C, функції abort(), exit(), atexit().

Принципи обробки виключень в мові C++. Генерація виключень. Послідовність обробки виключень. Особливості обробки виключень при динамічному виділенні пам'яті.

6.2. Особливості нових стандартів.

Цикл по колекції. Списки ініціалізації, універсальна ініціалізація. Переліки зі строгою типізацією. Додаткові контейнери та зміни в контейнерах. Зовнішні шаблони. Шаблони зі змінною кількістю аргументів. Клас std::tuple. Вивід типів. Лямда-функції, альтернативний синтаксис функцій. Регулярні вирази.

Змістовий модуль 3. Аналіз алгоритмів та алгоритмічні стратегії **Тема 7. Математичні основи аналізу алгоритмів.**

7.1. Основи аналізу алгоритмів.

Асимптотичний аналіз верхньої та середньої оцінок складності алгоритмів; порівняння найкращих, середніх і найгірших оцінок; O -, o -, ω -а

θ -нотації; емпіричні вимірювання ефективності алгоритмів; накладні витрати алгоритмів за часом і пам'яттю; рекурентні співвідношення та аналіз рекурсивних алгоритмів.

Порівняння алгоритмів, вплив структур даних і особливостей реалізації на ефективність алгоритмів.

7.2. Алгоритмічні стратегії

Методи розробки алгоритмів: структурне програмування, рекурсія, обходи дерев, «поділяй і пануй», балансування, динамічне програмування, програмування з відходом назад, метод «гілок і меж», евристичні та наближені алгоритми.

Тема 8. Основи теорії обчислюваності. Класи складності P і NP

8.1. Поняття обчислюваною функції.

Властивість покрокового виконання алгоритму. Розв'язані множини і їх властивості. Перераховувані множини і їх властивості. Перераховувана множина, як множина визначення обчислюваної функції. Перераховувана множина, як множина значень обчислюваної функції. Теорема Поста. Теорема про графік обчислюваної функції.

8.2. Поняття складності обчислення. Функція складності обчислень (за часом). Аксиоми Блюма. Теорема про прискорення. Класи складності. Опис класів P і NP. Приклади завдань, що належать цим класам. Ототожнення класу P з класом реально обчислювальних функцій. Поліноміальне зведення NP-повні задачі. Теорема Кука. Приклади NP-повних задач. Проблема перебору ($P = NP?$). Застосування теорії NP-повноти для аналізу складності завдань.

Модуль 4. Фундаментальні алгоритми та їх побудова Тема

9. Алгоритми сортування, злиття та пошуку

9.1. Прості алгоритми сортування.

Значення сортувань при реалізації алгоритмів. Класифікація сортувань. Характеристики сортувань. Прості сортування як спосіб швидкої реалізації алгоритму. Приклади простих сортувань – метод простого включення, метод простого обміну (бульбашкове сортування), шейкерне сортування, сортування вставками, сортування підрахунком, цифрове сортування. Переваги і недоліки простих сортувань.

9.2. Оптимізовані алгоритми сортування.

Складні сортування як спосіб створення ефективних алгоритмів. Приклади складних сортувань – сортування Шелла, сортування Хоара (швидке сортування), сортування злиттям. Переваги і недоліки складних сортувань. Порівняння простих та складних сортувань.

Тема 10. Фундаментальні алгоритми на графах і деревах

10.1 Основні поняття теорії графів. Матричне подання графів. Матриця зв'язності та матриця відстаней на графі. Пошук найкоротших шляхів та оптимальних маршрутів у графах. Алгоритм Дейкстри. Метод Беллмона. Знаходження мінімального остовного дерева графа за алгоритмом Прима-Краскала. Перевірка зв'язності графів. Алгоритм Тар'яна знаходження найменшого спільного пращура. Задача про найменше вершинне покриття.

10.2 Задача про гамільтонові шляхи на графі. Пошук у ширину на графах. Пошук у глибину на графах.

Тема 11. Алгоритми пошуку в рядках та їх обробки

11.1 Алгоритми пошуку в рядках: бінарний пошук, алгоритм Бойера–Мура, алгоритм Кнута–Морріса–Пратта, алгоритм Карпа–Рабіна, наближений пошук.

11.2 Прості алгоритми побудови дерева суфіксів. Алгоритм Укконена.

Масиви суфіксів. Задача про найбільший спільний підрядок двох рядків.

11.3 Основні алгоритми обробки рядків – розбиття рядків, об'єднання рядків, алгоритми вставки, видалення, заміни.

Тема 12. Геометричні алгоритми

12.1 Основні формули обчислювальної геометрії. Знаходження довжини відрізка в n -вимірному просторі. Відстань від точки до прямої.

Координати точок перетину відрізків і прямих. Рівняння прямої, кола, площини. Знаходження площі багатокутника. Метод триангуляції. Метод трапецій. Перевірка опуклості багатокутника.

12.2 Векторна геометрія. Колінеарність векторів. Перевірка належності точок прямій. Ліві та праві трійки векторів. Знаходження порядку обходу вершин опуклого багатокутника. Задачі мінімізації в геометричній інтерпретації.

Тема 13. Евристичні та криптографічні алгоритми

13.1 Поняття «жадібного» алгоритму. Теоретичні основи «жадібних» алгоритмів. Переваги та недоліки «жадібних» алгоритмів.

13.2 Класичні приклади «жадібних» алгоритмів. Задача про вкладання рюкзака. Розв'язання задач із застосуванням «жадібних» алгоритмів. Геометричні, транспортні, економічні задачі.

Поняття про динамічне програмування. Основні підходи до розв'язання задач методом динамічного програмування. Матричне числення.

Перемноження декількох матриць. Знаходження найбільшої спільної підпоследовності множин. Визначення оптимальної триангуляції

багатокутника. Задачі лінійного програмування. Симплекс-метод розв'язання задач економічного планування. Криптографічні алгоритми та їх застосування для захисту інформації.

4. Структура навчальної дисципліни

Назви змістових модулів та тем	Кількість годин
--------------------------------	-----------------

	усього	у тому числі			
		лекцій	пз	сем	сам. р.
Змістовий модуль 1. Структури даних					
Структури та об'єднання	8	3			5
Введення в систему вводу-виводу C++. Файлові структури даних	8	3			5
Динамічні структури даних	9	3	1		5
<i>Разом за змістовим модулем 1</i>	25	9	1		15
Змістовий модуль 2. Бібліотека шаблонів					
Стандартна бібліотека шаблонів. Шаблони функцій	9	3	1		5
Стандартний клас String	9	3	1		5
Обробка виключень. Особливості стандартів C, C++	9	3	1		5
<i>Разом за змістовим модулем 2</i>	27	9	3		15
Змістовий модуль 3. Аналіз алгоритмів та алгоритмічні стратегії					
Математичні основи аналізу алгоритмів	9	3	1		5
Основи теорії обчислюваності. Класи складності P і NP	9	3	1		5
<i>Разом за змістовим модулем 3</i>	18	6	2		10
Змістовий модуль 4. Фундаментальні алгоритми та їх побудова					
Алгоритми сортування, злиття та пошуку	9	3	1		5
Фундаментальні алгоритми на графах і деревах	10	3	1		6
Алгоритми пошуку в рядках та їх обробки	10	3	1		6
Геометричні алгоритми	10	3	1		6
Евристичні та криптографічні алгоритми	11	4	1		6
<i>Разом за змістовим модулем 3</i>	50	16	5		29
Усього годин	120	40	11	-	69

5. Теми семінарських занять

Не передбачено

6. Теми практичних занять

№	Тема практичного заняття	Кількість годин
1.	Розгалужені алгоритми	2
2.	Циклічні алгоритми	2

№	Тема практичного заняття	Кількість годин
3.	Алгоритми пошуку	2
4	Алгоритми обробки масивів	2
5	Алгоритми сортування масивів	3

7. Теми лабораторних занять
Не передбачено

8. Самостійна робота

№ теми	Тема самостійної роботи	Кількість годин
1	Базові поняття теорії алгоритмізації	10
2	Поняття структури даних	10
3	Хешування даних	10
4	Нелінійні структури даних	10
5	Алгоритми пошуку	10
6	Алгоритми сортування	10
7	Підготовка індивідуального завдання	9
Разом		69

9. Індивідуальні завдання
Не передбачено

10. Методи навчання

Методи навчання, що використовуються у процесі лекційних занять:

- лекція з елементами пояснення;
- лекція-бесіда;
- лекція-дискусія;
- ілюстрація наочних матеріалів;
- пояснення.

Методи навчання, що використовуються під час практичних занять:

- виконання вправ та завдань;
- застосування комп'ютерної техніки та прикладних програм для вирішення задач;
- розробка власних прикладних програм;
- сумісна робота над проектом;
- самостійна робота.

11. Методи контролю

Підсумковий контроль. Для контролю засвоєння дисципліни навчальним планом передбачений екзамен. Проведення підсумкової атестації і отримання на ній позитивної оцінки включає:

- а) оцінку проміжної атестації (результати модуля),

- б) оцінку відвідуваності занять і активність в аудиторії;
в) оцінку виконання усіх завдань самостійної роботи.

Поточний контроль. Для поточного контролю використовуються результати практичних занять.

12. Розподіл балів, які отримують студенти

Поточне тестування та самостійна робота													Е к з а м е н	С у м а
Заліковий модуль 1					Заліковий модуль 2		Заліковий модуль 3							
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	20	100
6	7	6	7	6	6	6	6	6	6	6	6	6		

T1, T2, T12 – теми змістових модулів.

Шкала оцінювання: національна та ECTS

Бали	Оцінка		
	Шкала	Національна шкала	
		Іспит	Залік
90–100	A	Відмінно	Зараховано
82–89	B	Добре	
74–81	C		
67–73	D	Задовільно	
60–66	E		
35–59	FX	Незадовільно з можливістю повторного складання	Не зараховано з можливістю повторного складання
1–34	F	Незадовільно з обов'язковим повторним курсом	Не зараховано з обов'язковим повторним курсом

13. Методичне забезпечення

№ з/п	Найменування методичних матеріалів	Рік вида ння	наявність в бібл., примірн	Ел. варі- ант	Код
1. Підручники					
				+	