



Структури даних і алгоритми

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>123 Комп'ютерна інженерія</i>
Освітня програма	<i>Комп'ютерні системи та мережі</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>очна</i>
Рік підготовки, семестр	<i>1 рік (1 семестр)</i>
Обсяг дисципліни	<i>5 кредитів (ECTS). Виділення часу -150 годин, включаючи 54 аудиторних години та 98 годин самостійного навчання</i>
Семестровий контроль/ контрольні заходи	<i>1 семестр – залік</i>
Розклад занять	<i>1.5 заняття на тиждень, розклад на http://rozklad.kpi.ua/</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	Лектор: д.т.н., с.н.с., Анатолій Михайлович Сергієнко, email anat.srg@gmail.com , web page http://kanyevsky.kpi.ua Викладач з практичних занять: Анастасія Анатоліївна Молчанова, email an.ser.313kpi@gmail.com
Розміщення курсу	Лекційний матеріал: https://bbb.comsys.kpi.ua/b/ana-gca-2xm Методичні рекомендації: https://kanyevsky.kpi.ua/студентам/алгоритми-і-структури-даних/

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Викладання алгоритмів та структур даних в університеті має такі аспекти:

Академічний аспект. Алгоритми та структури даних - це наука, яка дає фундаментальні знання. Мета - ознайомити студентів з основними поняттями та методами представлення даних та виконання з ними алгоритмів, вимірюючи складність алгоритму.

Професійний аспект. Знання структур даних та алгоритмів є обов'язковими для інженерів та програмістів, які розробляють програмне забезпечення для комп'ютера. Мета цього курсу – сформувати навички використання мови C для програмування базових алгоритмів, які використовуються у більшості програмного забезпечення універсальних та спеціалізованих систем.

Інтелектуальні та освітні аспекти. Вивчення структур даних та алгоритмів сприяє розвитку когнітивних навичок, що є суттєвим інтелектуальним фактором у процесі створення програмного забезпечення.

В результаті вивчення цієї дисципліни досягаються такі результати навчання.

Компетенції:

- здатність до абстрактного мислення, аналізу та синтезу;
- здатність до навчання та набуття сучасних знань;
- вміння застосовувати знання в практичних ситуаціях;
- здатність до алгоритмічного та логічного мислення
- вміння оформляти отримані результати роботи у вигляді презентацій, науково -технічних

звітів.

- здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення
 - володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних
- Знання:

- основні поняття та методи теорії алгоритмів;
- основні структури даних та алгоритми;
- базові знання про архітектуру комп'ютера, операційну систему, компіляцію програм,

обчислювальні процеси в комп'ютерах;

- знати підходи щодо оцінки та забезпечення якості програмного забезпечення
- методи вимірювання складності алгоритму та програми.

Навички:

- програмування базових алгоритмів мовою С з різними структурами даних;
- вирішення задач аналізу складності алгоритмів та програм;
- знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення
- систематично мислити та застосовувати творчі здібності для формування нових ідей.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Передумови, тобто дисципліни, вивчення яких має передувати вивченню цієї дисципліни:

- базовий рівень володіння англійською мовою не нижче А1.

Постреквізити - дисципліни, вивченню яких має передувати вивчення даної дисципліни:

- об'єктно-орієнтоване програмування;
- архітектура комп'ютера;
- комп'ютерна графіка;
- чисельні алгоритми та методи;
- розробка програм;
- системне програмування;
- комп'ютерні мережі;
- основи паралельного програмування;
- системи обробки сигналів;
- програмування мобільних систем;
- сучасні технології програмування.

3. Зміст навчальної дисципліни

Тема 1: Основи мови С.

Предмет і мета предмета «структури даних та алгоритми». Поняття алгоритму. Архітектура з точки зору програміста. Історія мови С. Модель мови С. Об'єкти мови С. Операції та оператори мови С. Перетворення типів у виразах.

Тема 2: Алгоритми та програми.

Конструкції для проектування алгоритму. Блокові конструкції. Конструкції управління: конструкції розгалуження, повторення, перехід. Подання алгоритмів та оператори. Діаграми діяльності. Алгоритмічні блок-схеми. Основи структурного програмування.

Основи оцінки алгоритму. Часова складність. Просторова складність. Оцінка порядку складності. Ітеративні алгоритми. Типи алгоритмів: поліноміальні комплексні та NP-повні алгоритми. Вимірювання складності в програмі. Оптимізація алгоритму. Найпростіші алгоритмічні прийоми. Метод динамічного програмування.

Підпрограми. Функції та процедури. Контекст підпрограми, інтерфейс та реалізація. Реалізація підпрограми в архітектурі. Методи передачі параметрів. Структура програми С. Програмування функцій у С. Методи передачі параметрів у підпрограму.

Тема 3: Структури даних.

Концепція абстрактного типу даних. Функції над структурами даних. Класифікації структури даних. Статичні та динамічні структури даних. Скалярні структури даних. Комбіновані структури даних. Рівномірні та неоднорідні структури даних. Масиви.

Алгоритми зсуву масиву. Одно- та багатовимірне оголошення та зберігання масиву. Програмування дисплея консолі. Алгоритми переміщення 2-D масиву. Особливості роботи з квадратними матрицями.

Тема 4: Алгоритми з масивами.

Алгоритми пошуку. Алгоритм лінійного пошуку та його програмування. Алгоритми сортування та їх використання. Оцінка алгоритмів сортування. Сортування шляхом прямої вставки. Алгоритми сортування шляхом прямого вибору. Алгоритм швидкого сортування. Сортування злиттям. Сортування з підрахунком. Апаратна сортування. Алгоритми з рядками. Функції зі рядками. Структури. Множини. Юніони.

4. Навчальні матеріали

Базова література

1. Б. Керніган, Д. Рітчі. Мова програмування Сі. 229 с.
2. Дж. Макконелл. Основи сучасних алгоритмів. 2004. 368 с.

Додаткова література

1. Н. Н. Непийвода, І. Н. Скопін. Основи програмування. 2002. 919 с.
2. Н. Вірт, Алгоритми та структури даних. 358 с. ДМК. 2010. 272 с.

Слайди лекцій та керівництва до лабораторних робіт <https://kanyevsky.kpi.ua/en/for-students/>

Обладнання, що необхідне для проведення занять

Лекційні заняття проводяться в аудиторії, яку обладнано проектором, практичні заняття – в комп'ютерному класі.

5. Методика опанування навчальної дисципліни (освітнього компонента)

Навчальний зміст дисципліни складається з лекцій та лабораторних робіт.

Лекції.

Семестр 1.

Тема 1: Основи мови C.

Лекція 1. Вступ.

Предмет і мета курсу «структур даних та алгоритмів». Сучасний рівень обчислювальної техніки. Навчальні матеріали. Програмне забезпечення для використання. Система оцінок. Історія алгоритмів та розрахунків. Концепція алгоритму. Обчислювальний процес. Граф алгоритму. Архітектура комп'ютера та його історія. Ієрархія архітектури. Архітектура з точки зору програміста.

Лекція 2. Модель мови C.

Модель фон Неймана. Історія мови C. Стандарти мови C. Мова C для комп'ютера PDP-11. Модель мови C. Об'єкти мови C: коментарі, символи, ключові слова, ідентифікатори, літерали. Дані та їх види. Змінні. Локальні та глобальні об'єкти.

Лекція 3. Операції з мовою C

Операції та оператори. Арифметичні та логічні операції. Операції призначення. Пріоритети операцій. Перетворення типів у виразах.

Тема 2: Алгоритми та програми.

Лекція 4. Конструкції для проектування алгоритму

Блокові конструкції. Конструкції управління: конструкції розгалуження, повторення, перехід. Умовні, безумовні та селективні конструкції. Однорівневі конструкції переходів.

Лекція 5. Подання алгоритмів та оператори.

Діаграми діяльності. Блок -схеми алгоритмів. Оператори умовного та безумовного переходу. Функції вводу-виводу. Теорія схем програм. Основи структурного програмування.

Лекція 6. Основи оцінки алгоритмів.

Часова складність. Просторова складність. Стандартні моделі для оцінки часової складності. Розрахунок часової складності в програмах C. Оцінка порядку складності. Ітеративні алгоритми. Приклад алгоритму та розрахунок його складності.

Лекція 7. Основи оцінки алгоритму (продовження)

Спрощена оцінка складності. Типи алгоритмів: поліноміальна складність та NP-повні алгоритми. Вимірювання складності в програмі. Приклад алгоритму та розрахунок його складності.

Лекція 8. Оптимізація алгоритму.

Найпростіші алгоритмічні прийоми. Використання множника зі змінним знаком. Перевірка парних номерів. Використання прапорця знака. Розгортання циклів. Заміна множення на додавання. Метод динамічного програмування. Мінімізація умовних гілок.

Лекція 9. Підпрограми.

Функції та процедури. Контекст, інтерфейс та реалізація підпрограми. Виклик підпрограми. Реалізація підпрограми в архітектурі. Методи передачі параметрів. Функції на мові C. Структура програми C. Програмування функцій у C. Передача параметрів за значенням. Параметр, що передається за посиланням. Параметр типу const. Ключове слово static. Конвенція про виклик підпрограм у мові C.

Тема 3: Структури даних.

Лекція 10. Структури даних.

Структури даних. Концепція абстрактного типу даних. Функції над структурами даних. Класифікація структур даних. Статичні та динамічні структури даних. Скалярні структури даних. Комбіновані структури даних. Рівномірні та неоднорідні структури даних. Масиви. Одновимірний масив, його зберігання та доступ.

Лекція 11. Масиви та їх зміна.

Алгоритми зсуву масиву. Циклічний зсув масиву в одну позицію. Циклічний зсув масиву на k позицій. Обмін блоками. Циклічний зсув, що зберігає середню частину масиву. Циклічний зсув із переворотами.

Лекція 12. Двовимірні масиви

Оголошення та зберігання багатовимірних масиву. Ініціалізація двовимірних масиву. Програмування дисплея консолі. Бібліотека Windows та її функції для програмування прямого доступу до буфера консолі. Алгоритми обходу двовимірних масиву. Особливості роботи з квадратними матрицями.

Тема 4: Алгоритми з масивами.

Лекція 13. Алгоритми пошуку.

Задача пошуку. Алгоритм лінійного пошуку та його програмування. Класичний алгоритм лінійного пошуку. Лінійний пошук з бар'єром. Двійковий пошук та його дві варіації. Прямий пошук рядків. КМП-пошук. Алгоритм Рабіна-Карпа. Алгоритми пошуку з хеш-функцією.

Лекція 14. Алгоритми сортування.

Алгоритми сортування та їх використання. Оцінка алгоритмів сортування. Сортування прямою вставкою. Алгоритм з лінійним пошуком за вставленим елементом. Алгоритм з лінійним пошуком від елемента, який вставляється. Алгоритм з лінійним пошуком від елемента, що вставляється з бар'єром. Алгоритм з двійковим пошуком. Аналіз алгоритмів прямої вставки.

Лекція 15. Алгоритми сортування (продовження)

Алгоритми сортування шляхом прямого вибору. Алгоритм прямого обміну. Алгоритм бульбашкового сортування. Алгоритм шейкерного сортування. Алгоритм Шела.

Лекція 16. Алгоритми сортування (закінчення)

Алгоритм швидкої сортування. Сортування злиттям. Сортування з підрахунком. Апаратне сортування. Мережа бітонного сортування. Мережа бульбашкового сортування. Мережа сортування парно-непарним злиттям.

Лекція 17. Алгоритми з рядками

Рядки. Представлення рядків в комп'ютерах. Функції зі рядками. Структури. Множини. Юніти.

Лекція 18.

Модульна контрольна робота.

Лабораторні роботи

Семестр 1.

Лабораторна робота 1.

Розгалужені алгоритми.

Лабораторна робота 2.

Циклічні алгоритми та динамічне програмування.

Лабораторна робота 3.
Алгоритми лінійного пошуку.
Лабораторна робота 4.
Алгоритми обходу двовимірних матриць.
Лабораторна робота 5.
Двійкові алгоритми пошуку.
Лабораторна робота 6.
Алгоритми сортування.

6. Самостійна робота студента

Самонавчання включає самостійну роботу студентів і полягає в наступному:

- підготовка до лекцій шляхом вивчення попереднього лекційного матеріалу, а також літературних джерел, на яких ґрунтується матеріал попередніх лекцій (перелік джерел та перелік розділів подаються разом із матеріалом лекції);
 - підготовка до лабораторної вправи шляхом ознайомлення із завданням та методичними вказівками до лабораторних робіт, включаючи вивчення теоретичного матеріалу, необхідного для відповіді на контрольні запитання до лабораторних робіт;
- Кожна лабораторна робота вимагає для підготовки та виконання двох тижнів.

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Система вимог до студентів:

- студент зобов'язаний відвідувати лекції та лабораторні заняття, активно працювати над засвоєнням викладеного на них матеріалу;
- на лекції лектор використовує власний презентаційний матеріал;
- лабораторні роботи захищаються у два етапи: перший етап - студенти виконують завдання, складають електронний звіт та надсилають викладачу; другий етап - захист лабораторних робіт у лабораторії. Контроль знань на лабораторних роботах здійснюється шляхом перевірки звіту про лабораторні роботи, а також питань з матеріалів робіт.
- модульна контрольна робота пишеться як тест на лекції з використанням усіх наявних матеріалів, але без використання комп'ютерних засобів.

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

На першому занятті студенти знайомляться з політикою оцінювання, яка ґрунтується на Положенні про систему оцінювання результатів навчання https://document.kpi.ua/files/2020_1-273.pdf. Рейтинг студента на курсі складається з балів, які він/вона отримує за захищені лабораторні роботи (R1) та модульну контрольну роботу (R2).

$$R_s = R_1 + R_2 = 100 \text{ балів}$$

В результаті максимальний середній бал дорівнює:

$$6 \text{ лабораторних робіт} \times 10 \text{ балів} = 60 \text{ балів}$$

$$\text{модульна контрольна робота} = 40 \text{ балів}$$

Згідно з університетськими правилами щодо моніторингу успішності студента (https://kpi.ua/document_control), існує два тижні оцінювання (атестація), зазвичай протягом 7/8 та 14/15 тижнів семестру, коли оцінюється прогрес у навчанні студентів та відповідність до критеріїв політики оцінювання курсу.

Умовою першої атестації є отримання не менше 10 балів (на момент атестації). Умова другої атестації - набрати не менше 30 балів (на момент атестації).

Критеріями оцінки є:

- Виконання лабораторних робіт:

- бездоганна робота оцінюється у 9 балів;

- є певні недоліки в оформленій роботі - 8-7 балів;

- є деякі недоліки у виконанні робочої програми - 6-5 балів;

Робота не виконана або не захищена - 0 балів.

За роботу, подану вчасно, діє заохочення - 1 бал (сумарно не більше - 6 балів).

- Модульна контрольна робота оцінюється 40 балами. Контрольна робота складається з 16 тестових питань, а також практичного завдання (скласти програму) зі списку, наведеного в додатку до робочої програми.

За кожну правильну відповідь на тестове питання нараховується 2 бали. Відповідь на практичне завдання оцінюється 8 балами за такими критеріями:

- "відмінно" - правильний текст програми з коментарями - 8 - 7 балів;

- "добре" - текст програми, загалом правильний, але не використаний прийом оптимізації - 6-5 балів;

- "задовільно" - є деякі принципові помилки в тексті програми - 4 - 3 бали;

- "незадовільно" - незадовільна відповідь - 0 балів.

Студенти, які нарешті наберуть необхідну кількість балів (≥ 60), можуть:

- отримати остаточну залікову оцінку відповідно до рейтингового балу;

- виконати залікову контрольну роботу, щоб підвищити оцінку.

Студенти можуть отримати до 6 заохочувальних балів за виконання творчих робіт з кредитного модуля (складання тез, участь у конкурсах, у дослідженнях тощо).

Студенти, чий остаточний бал за успішність Rs нижче 60 балів, але більше за 30, повинні виконати залікову контрольну роботу. Якщо оцінка за тестування нижча від оцінки, яку студент отримує за свою семестрову діяльність, застосовується суворіша вимога - попередній рейтинг студента анулюється, і він отримує оцінку за результатами залікової контрольної роботи. Студенти, чий бал нижче 30, не допускаються до заліку.

Оцінюється, що залікова контрольна робота складає до 60 балів. Контрольне завдання цієї роботи складається з трьох питань. Кожне з 3 питань оцінюється 20 балами відповідно до таких критеріїв:

- "відмінно" - повна відповідь (принаймні 90% необхідної інформації), за умови відповідного обґрунтування та особистої думки - 20 - 18 балів;

- "добре" - досить повна відповідь (щонайменше 75% необхідної інформації), виконана відповідно до вимог щодо рівня "кваліфікації", або незначні неточності) - 17... 15 балів;

- "задовільно" - неповна відповідь (не менше 60% необхідної інформації, яка виконується відповідно до вимог до "стереотипного" рівня та деяких помилок) - 14... 12 балів;

- "незадовільно" - незадовільна відповідь - 0 балів.

Таблиця. Остаточна оцінка RS приймається за системою оцінювання університету

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

9. Додаткова інформація про курс

Питання в модульній контрольній роботі виглядають як наступні.

1. Обчислювальний алгоритм – це а) послідовність дій, виконання якої приводить до очікуваного результату, б) обчислювальний процес, що заданий на чітко описаній моделі обчислювача, в) граф або програма, обхід частин яких відповідає деякому обчислювальному процесу.
2. В ЕОМ використовуються 6-бітові літери. Алфавіт цієї ЕОМ вміщує: а) 32 цифри і знаки, б) 55 цифр, букв і знаків, в) 65 цифр, букв і знаків.
3. Що означає у мові Сі запис: \0 — а) ділення на нуль без переривання, б), символ NULL в) ознака кінця рядка?
4. Змінна типу int займає в машині об'єм – а) 16 біт, б) 32 біт, в) 16 або 32 біт .
5. Вираз: $a \& b / c \& d$; еквівалентний виразу — а) $(a \& (b / c)) \& d$; б) $a \& ((b / c) \& d)$; в) $(a \& b) / (c \& d)$;
6. Результатом операції $12 \mid \mid 6 \epsilon - a$ б , б) 1 , в) 14 .
7. Результатом операції `switch (3 >> 1){case 0: a = 1; case 1: a = 2; break; default: a=3;}` ϵ : а) 1, б) 2, в) 3.
8. Блок — це: а) виділена частина програми, змінні якої мають локальну дію, б) виділена частина програми, яка має інтерфейс для вхідних і вихідних змінних, в) частина структури комп'ютера.
9. цикл з передумовою – це цикл, який а) перестає виконуватись за умовою, б) виконується за умовою, в) це цикл, який починає виконуватись за умовою .
10. Щоб цикл з лічильником не зациквився, слід забезпечити – а) умови обов'язкового виходу з циклу, б) щоб справджувалась умова оператора for хоча б в одній з ітерацій, в) щоб змінна лічильника була цілого типу.
11. Оператор continue призначений для – а) продовження роботи програмного модуля після переривання, б) виходу з циклу, в) переходу на наступну ітерацію циклу.
12. Оператор `for (i = 1; ; i++){...}` — виконує цикл, – а) оператор є неправильним, б) якщо у його блоці є оператор виходу з циклу, в) якщо у його блоці є оператор виходу з циклу, інакше — нескінченний цикл.
13. При виконанні фрагменту: `int A[3] = {1,2,3}; printf(“%d”,A[3]);` буде виводитись – а) 3 , б) 0 , в) невідоме число.
14. Дані статичної структури – це дані, – а) які складаються з констант, б) взаєморозташування та взаємозв'язки яких є постійними, в) які мають незмінне розташування під час виконання програми.

15. У програмі на Сі масив задано як `int a[2][3] = { 1, 2, 3, 4, 5, 6 }`; Чому дорівнює елемент `a[2][3]` а) NULL, б) невідоме число, в) 6 ?.

16. Сортування з підрахунком ґрунтується на тому, що підраховується – а) кількість елементів у масиві однакової величини, б) дистанція, на яку слід перемістити черговий елемент на черговому кроці впорядкування масиву, в) довжина частини масиву, яка потребує сортування.

II Розробити алгоритм і програму, що вирішує задачу. Є число з плаваючою комою X. Розрахувати

масив $X[n]$, у якому $X[0] = X$; $X[i] = \sum_{k=1}^i (-X)^k (k+1)$; Число операцій має бути мінімізоване.

Робочу програму навчальної дисципліни (силабус):

Складено д.т.н., с.н.с, А. М. Сергієнко

Ухвалено кафедрою обчислювальної техніки (протокол № 10 , від 25.05.2022)

Погоджено Методичною комісією ФІОТ (протокол № 10 , 13.06. 2022)