Рівненський державний гуманітарний університет

Факультет математики та інформатики

Кафедра інформаційних технологій та моделювання

АНОТАЦІЯ

Назва дисципліни / освітнього компонента	**Теорія алгоритмів**
Освітня програма	**Інженерія програмного забезпечення**
Компонент освітньої програми	**Вибірковий**
Загальна кількість кредитів та кількість годин для вивчення дисципліни	**3 кредити / 90 годин**
Вид підсумкового контролю з	**залік**
Мова викладання	**українська**
Викладач	**Шевцова Наталія Вікторівна, к.т.н., доцент кафедри інформаційних технологій та моделювання**
CV викладача на сайті кафедри	**https://kitm.rshu.edu.ua/sklad-kafedru/shevcova-natalia-victorivna/**
E-mail викладача	**natalia.shevtsova@rshu.edu.ua**

Мета та завдання навчальної дисципліни

Навчальна дисципліна «Теорія алгоритмів» має фундаментальне значення для підготовки фахівців з інженерії програмного забезпечення, оскільки формує строге алгоритмічне мислення, необхідне для аналізу, проєктування та оцінки ефективності програмних систем.

Мета навчальної дисципліни «Теорія алгоритмів» полягає в ознайомленні здобувачів вищої освіти з математичними основами теорії алгоритмів, які слугують інструментарієм для формалізації обчислювальних задач, та у формуванні практичних навичок аналізу та синтезу алгоритмів для розробки ефективного програмного забезпечення.

Основними завданнями вивчення дисципліни «Теорія алгоритмів» є:

- Закріплення та систематизація знань, отриманих під час вивчення програмування, для їх використання в алгоритмічному моделюванні програмних систем.
- Вивчення основних формальних моделей алгоритмів (машини Тьюринга, Поста, алгоритмів Маркова, рекурсивних функцій) та їхньої ролі в обґрунтуванні обчислюваності та коректності програм.

- Формування вмінь аналізувати складність алгоритмів (часову та просторову), оцінювати їх ефективність та обирати оптимальні підходи для вирішення прикладних задач проєктування ПЗ.
- Опанування основних методів розробки алгоритмів (динамічне програмування, «жадібні» підходи) та їх застосування для створення надійних і продуктивних програмних компонентів.
- Підготовка інженерів ПЗ, здатних формалізувати вимоги до програмної системи, обґрунтовувати вибір алгоритмічних рішень та прогнозувати їхню поведінку з точки зору ресурсів та обчислювальної складності.

В результаті вивчення навчальної дисципліни здобувачі вищої освіти повинні:

знати:

- Основні поняття, формальні моделі та фундаментальні властивості алгоритмів.
- Класичні абстрактні моделі обчислень (машини Тьюринга, Поста, алгоритми Маркова) та їх зв’язок з сучасними парадигмами програмування.
- Принципи алгоритмічної розв'язності та обмеження обчислювальних систем (нерозв'язність масових проблем).
- Методи аналізу алгоритмічної складності (асимптотичну оцінку) та базові класи складності (P, NP), важливі для оцінки продуктивності ПЗ.

вміти:

- Формалізувати обчислювальні задачі, що виникають на етапі проєктування ПЗ, за допомогою абстрактних моделей.
- Аналізувати алгоритми на коректність та оцінювати їх часову й просторову ефективність для обґрунтування архітектурних рішень.
- Застосовувати основні методології розробки алгоритмів для створення оптимальних програмних модулів.
- Ідентифікувати клас складності прикладної задачі та прогнозувати можливості її ефективної програмної реалізації.

набути компетентностей:

загальні:

К01. Здатність до абстрактного мислення, аналізу та синтезу.

К02. Здатність застосовувати знання у практичних ситуаціях.

фахові:

К14. Здатність брати участь у проєктуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

К20. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв’язання завдань інженерії програмного забезпечення.

К26. Здатність до алгоритмічного та логічного мислення.

та **програмних результатів навчання:**

ПР01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.

ПР05. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об’єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.

ПР13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

Зміст навчальної дисципліни

Змістовий модуль 1. Фундаментальні поняття та моделі алгоритмів.

Тема 1. Поняття алгоритму та його властивості. Підходи до визначення алгоритму (інтуїтивне, формальне). Фундаментальні властивості: визначеність (детермінованість), результативність (скінченність), масовість, дискретність. Вимоги до алгоритмів.

Тема 2. Типи та структури алгоритмів. Класифікація алгоритмів за різними ознаками (призначенням, способом реалізації, виконанням). Базові алгоритмічні конструкції (структури): лінійне слідування, розгалуження (умовний перехід), цикл (ітерація). Поняття про структурне програмування.

Тема 3. Формальні моделі алгоритмів. Обчислювальні функції. Поняття формальної моделі. Визначення обчислювальної функції. Примітивно-рекурсивні функції (базові, оператори суперпозиції та примітивної рекурсії). Частково-рекурсивні функції та оператор мінімізації. Теза Чорча та її роль в теорії обчислюваності.

Тема 4. Моделі на основі абстрактних автоматів: машина Поста та Тьюрінга. Машина Поста: будова, програма, принцип роботи. Обчислювальна машина Тьюрінга: конструкція (стрічка, головка, стан), програма та таблиця переходів, приклади обчислень.

Тема 5. Нормальні алгоритми Маркова: алфавіт, схеми підстановок, стратегія застосування. Порівняння з іншими моделями. Машина з довільним доступом (RAM) як модель, близька до реальних комп'ютерів: регістри, набір команд, принцип адресації.

Змістовий модуль 2. Методи розробки та аналізу алгоритмів.

Тема 6. Аналіз складності алгоритмів. Асимптотична оцінка. Поняття часової та просторової складності. Асимптотична оцінка: O (верхня оцінка), Ω (нижня оцінка), Θ (точна оцінка). Приклади аналізу простих алгоритмів.

Тема 7. Класи складності обчислювальних задач. Проблема P та NP. Обчислювальні задачі та їхня складність. Поліноміальна та експоненціальна складність. Класи задач P та NP. Поняття NP-повних задач та гіпотези $P \neq NP$.

Тема 8. Методи проектування алгоритмів. Динамічне програмування.

Тема 9. Методи проектування алгоритмів. «Жадібні» алгоритми.