Затверджено Вченою Радою ОНУ
імені І.І. Мечникова
від "20"грудня 20 16 р. № 4

Одеський національний університет імені І. І. Мечникова
Факультет математики, фізики та інформаційних технологій
Кафедра теоретичної механіки

«ЗАТВЕРДЖУЮ»
Проректор з науково-педагогічної роботи

_____

"____" _____ 20___ р.

НАВЧАЛЬНА ПРОГРАМА ДИСЦИПЛІНИ

ОК9 Структури даних Алгоритми
(назва навчальної дисципліни)

Рівень вищої освіти Бакалавр

Спеціальність 126 – Інформаційні системи і технології
(код і назва спеціальності (тей)

2019

Розробники:
Волков В.Е., д-р техн. наук, професор, зав. кафедри ТМ
Рачинська А.Л., к. ф-м н., доцент
Царенко О.П., ст. викладач

Навчальна програма затверджена на засіданні кафедри теоретичної механіки

Протокол № 1 від. "2" вересня 2019 р.

Завідувач кафедри _____ (Волков В.Е.)
(підпис) (прізвище та ініціали)

Обговорено та рекомендовано до затвердження навчально-методичною комісією (НМК) з ІТ факультету МФІТ

Протокол № 1 від. "23" 09 2019 р.

Голова НМК _____ (Савастру О.В.)
(підпис) (прізвище та ініціали)

ВСТУП

Навчальна програма дисципліни «Структури даних та алгоритми» складена відповідно до освітньо-професійної програми підготовки бакалаврів спеціальності 126 – Інформаційні системи та технології (версія 2019 р.).

Дисципліна «Структури даних та алгоритми» є теоретичним та практичним фундаментом для фахівців з інформаційних систем та програмування.
Цей курс є другим в циклі курсів з основ програмування та полягає у наданні теоретичних знань та практичних навиків застосування загальних законів та правил проведення математичних обчислень із використанням сучасної обчислювальної техніки та об'єктно-орієнтованих мов програмування. Даний курс дає можливість студентам отримати знання, які допоможуть використовувати сучасні технології програмування для вивчення інших дисциплін.

Предметом вивчення навчальної дисципліни є структури даних та алгоритми, які реалізовані на мові програмування високого рівня MS C#.

Місце навчальної дисципліни в структурі освітнього процесу. Дисципліна «Структури даних та алгоритми» належить до обов'язкових дисциплін у підготовці бакалаврів.
Програма навчальної дисципліни складається з таких змістових модулів:

1. Прості чисельні алгоритми.
2. Обчислювальні алгоритми задач лінійної алгебри.
3. Структури даних.

1. Мета та завдання навчальної дисципліни

Вивчення дисципліни базується на знанні математичних дисциплін і загального курсу інформатики.
Основною метою курсу є придбання базових знань і практичних навичок розробки та налагодження програм на мові програмування високого рівня, вивчення теоретичних основ алгоритмізації і проектування програм, оволодіння базовими алгоритмами та структурами даних, навичками алгоритмічного мислення та структуризації даних, засвоєння методів тестування програм.

Цілі і завдання курсу

– поглибити та розширити знання студентів завдяки застосуванню їх математичного апарату при розв'язуванні практичних питань щодо побудування обчислювальних алгоритмів та застосуванню певних структур даних;
– вивчення структур даних, які застосовуються у програмуванні та інформатиці, їх специфікацій та реалізації, алгоритмів обробки даних та аналізу цих алгоритмів, взаємозв'язок алгоритмів та структур даних;
– оволодіння навичками розв'язування простіших обчислювальних задач науково-технічного характеру та змісту, та оформлення відповідних звітів;
– набуття студентами теоретичних знань та практичних навичок, опанування базових теоретичних понять, які лежать в основі розробки алгоритмів та структур даних;

– закладення в основу конструювання та використання складних (динамічних) структур даних модель (парадигму) абстрактного типу даних (специфікація + уявлення + програмна реалізація);
– навчитися визначати основні вимоги до алгоритмів (методів), які плануються до використання під час розв’язування певних фізико-математичних проблем;
– навчитися визначати особливості в програмних реалізаціях тих чи інших структур даних одного класу;
– навчитися будувати або формувати тестові завдання (задачі), що призначені для перевірки коректності розроблених алгоритмів та відповідних структур даних та відповідних програмних компонентів;
– сформувати уявлення та знання про основні класи алгоритмів (виключаючи пошук, швидкий пошук, сортування, алгоритми на графах тощо), та структурах даних, які на них використовуються, та загальних схемах розв’язування задач на їх основі;
– вивчити основні способи реалізації типових алгоритмів та структур даних та їх модифікацій на обраній мові програмування, наприклад, C#;
– вивчити особливості компонування комп’ютерних програм, де використовуються розроблені алгоритми та структури даних;
– сформувати уявлення та знання про аналіз складності алгоритмів та програм.

Змістовне наповнення дисципліни обумовлене загальними завданнями в підготовці фахівців з розробки комп'ютерних програм.

Процес вивчення дисципліни спрямований на формування елементів наступних **компетентностей** (згідно ОПП «Інформаційні системи та технології від **2019** р.):

Спеціальні (фахові, предметні) компетентності: КС.4, КС.11, КС.15
КС 4. Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші).
КС 11. Здатність до аналізу, синтезу і оптимізації інформаційних систем та технологій з використанням математичних моделей і методів.
КС 15. Здатність до математичного та логічного мислення, а також до використання математичного апарату фундаментальної та прикладної математики під час розв’язання прикладних і наукових завдань в області інформаційних систем і технологій.

Програмні результати навчання:
ПР 1. Знати лінійну та векторну алгебру, диференціальне та інтегральне числення, теорію функцій багатьох змінних, теорію рядів, диференціальні рівняння для функції однієї та багатьох змінних, операційне числення, теорію ймовірностей та математичну статистику в обсязі, необхідному для розробки та використання інформаційних систем, технологій та інфокомунікацій, сервісів та інфраструктури організації.
ПР 2. Застосовувати знання фундаментальних і природничих наук, системного аналізу та технологій моделювання, стандартних алгоритмів та дискретного аналізу при розв’язанні задач проектування і використання інформаційних систем та технологій.
ПР 3. Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп’ютерних програм мовами високого рівня із застосуванням об’єктно-орієнтованого програмування для розв’язання задач проектування і використання інформаційних систем та технологій.

ПР 4. Проводити системний аналіз об’єктів проектування та обґрунтовувати вибір структури, алгоритмів та способів передачі інформації в інформаційних системах та технологіях.
ПР 12. Розробляти та застосовувати моделі представлення знань, стратегії логічного виведення, технологій інженерії знань, технологій і інструментальних засобів побудови інтелектуальних систем і систем штучного інтелекту.

Очікувані результати навчання. У результаті успішного засвоєння навчальної дисципліни студент зможе продемонструвати такі результати навчання:

знання:

- основні методи розробки комп’ютерних алгоритмів та програм, структури даних, які використовуються для уявлення типових інформаційних об’єктів, основні задачі аналізу алгоритмів;
- основні комп’ютерні алгоритми та характеристики їх складності для типових задач, які часто зустрічаються на практиці та вже стали «класичними» в області інформатики та програмування;

уміння:

- застосовувати сучасні мови програмування та середовища розробки програмних компонентів до розв’язування певних або загальних фізико-математичних проблем;
- будувати стійки алгоритми, використовуючи викладені в курсі загальні схеми, методи та прийоми будування алгоритмів, обираючи необхідні структури даних для представлення інформаційних об’єктів;
- доводити коректність складеного алгоритму та оцінювати основні характеристики його складності;
- реалізовувати алгоритми та структури даних, що використовуються, засобами сучасних мов програмування;
- експериментальним шляхом (із допомогою комп’ютера) досліджувати ефективність алгоритму або програми;
- мати уяву про деякі математичні методи аналізу алгоритмів;
- мати уяву про класифікацію алгоритмічних задач за їх складністю, зведеністю алгоритмічних задач до відомих задач певного класу складності.

Дисципліна «Структури даних та алгоритми» викладається в другому семестрі для студентів 1 курсу, які навчаються за напрямом бакалаврської підготовки за спеціальністю 126 «Інформаційні системи і технології».
На вивчення навчальної дисципліни відводиться **165** годин **/ 5,5** кредитів ЄКТС.

2. **Зміст навчальної дисципліни**

Змістовий модуль 1. «Прості чисельні алгоритми»

Тема 1. «Введення у чисельні алгоритми»

Задачі обчислювальної математики. Основні етапи розв'язування фізико-математичної проблеми. Моделі, методи, алгоритмі та програмна реалізація.
Стійки та нестійкі алгоритми. Особливості реалізації математичних обчислень у вигляді програмних модулів та компонентів.
Особливості комп'ютерної обчислювальної системи. Література:[1–3]

Тема 2. «Прості чисельні алгоритми»

Алгоритми обчислення чисел Фібоначчі. Тестування алгоритмів.
Ділення двох цілих чисел із заданою кількістю цифр у результату. Тестування алгоритму та його програмної реалізації.
Алгоритм обчислення послідовності простих чисел.
Розкладання натурального числа на прості множники.
Алгоритм Евкліда знаходження найбільшого загального дільника двох чисел. Тестування алгоритму та його програмної реалізації.
Обчислення факторіалу. Література:[1–3]

Тема 3. «Біноміальні коефіцієнти»

Загальна постановка проблеми. Методи обчислення біноміальних коефіцієнтів. Тестування алгоритмів та відповідних програмних модулів. Література:[1–3]

Тема 4. «Алгоритми обчислення сум числових та степеневих рядів»

Загальна постановка проблеми. Абсолютна та відносна похибки при обчисленні сум числових рядів.

Степеневі ряди. Спеціальні функції, які мають представлення у вигляді степеневих рядів. Тестування алгоритмів та відповідних програмних модулів. Література:[1–3]

Змістовий модуль 2. «Обчислювальні алгоритми задач лінійної алгебри.»

Тема 5. «Операції з матрицями та векторами»

Загальні означення з теорії матриць. Операції з матрицями.

Методи вводу та виводу матриць та векторів. Методи генерації та елементарних перетворень з матрицями. Арифметичні операції з матрицями.
Алгоритми методів та їх програмні реалізації. Тестування алгоритмів та відповідних програмних модулів. Література:[4–6]

Тема 6. «Матричні обчислення»

LU-розкладання матриці. Обчислення визначника квадратної матриці. Алгоритми методів та їх програмні реалізації. Тестування алгоритмів та відповідних програмних модулів. Література:[4–6]

Змістовий модуль 3. «Структури даних»

Тема 7. «Лінійні структури даних»

Списки. Масиви. Зв'язані списки. Асоційовані масиви. Стеки. Черги. Графи. Дерева. Хеш-таблиці. Опис. Однозв'язні та двузвязні списки. Програмні реалізації. Література:[4–6]

Тема 8. «Динамічні масиви»

Опис динамічних масивів. Створення простого динамічного масиву. Клас **`MovingParticle`**. Клас **`My_Array_List<T>`**. Методи генерації динамічних масивів. Метод додавання елементів. Метод **`Insert`**. Методи **`RemoveAt`** та **`Remove`**. Метод індексатора. Методика виконання контрольного завдання. Література:[4–6]

Тема 9. «Стеки»

Стек. Описання. Клас **`My_Static_Array_Stack<T>`**. Метод **`Push`**. Метод **`Peek`**. Метод **`Pop`**.

Клас **`My_Singly_Linked_Stack_Item<T>`**.

Клас **`My_Singly_Linked_Stack<T>`**. Методи **`Push, Peek, Pop`**.

Клас **`My_Doubly_Linked_Stack<T>`**.

Застосування стеків до аналізу виразів. Інфіксний запис арифметичних виразів. Префіксна та постфіксна нотація. Література:[4–6]

Тема 10. «Черги»

Черги. Опис. Програмна реалізація. Дії із елементами черги.

Черги з пріоритетами.
Алгоритми методів та їх програмні реалізації. Тестування алгоритмів та відповідних програмних модулів. Література:[4–6]

Тема 11. «Бінарні дерева пошуку»

Словники. Асоційовані масиви. Операції з елементами словників. Програмні реалізації словників через дерева пошуку, хеш-таблиці, списки з пропусками, масиви, зв'язані списки. Операції із елементами. Література:[4–6]

Тема 12. «Хеш-таблиці. Дерева»

Хеш-таблиці. AVL–дерева. Splay–дерева. B– дерева. Префіксні дерева. Суфіксні дерева. Література:[4–6]

Тема 13. «Алгоритми текстового пошуку» Література:[4–6]

Алгоритми шаблонного співставлення. Алгоритм Бойера-Мура. Алгоритм Кнута-Моріса-Пратта. Trie. Компресія текста. Алгоритм кодування Хаффмана.

Тема 14. «Графи» Література:[4–6]

Структурі даних для графа. Проходження графів. Пошук в глибину та в ширину. Орієнтовані графи. Транзитивне замикання. Алгоритм Дейкстри. Алгоритм Крускала.

3. Рекомендована література

3.1 Основна література

1. Стратієнко Н.К., М.Д.Голдевський, І.О.Бородіна Алгоритми і структури даних: практикум: навчальний посібник / – Харків: НТУ «ХПІ», 2017. – 224 с.
2. Коротєєва Т.О. Алгоритми та структури даних : навч. посібник / Коротєєва. – Львів: Видавництво Львівської політехніки, 2014. – 280 с.
3. Robert Sedgewick and Kevin Wayne Algorithms Fourth Edition Part I Addison-Wesley, 2014 - 531p.
4. Robert Sedgewick and Kevin Wayne Algorithms Fourth Edition Part I Addison-Wesley, 2014 - 407p.
5. Michael T. Goodrich Roberto Tamassia Data Michael H. Goldwasser Structures and Algorithms in Java Sixth Edition, Wiley 2010 - 738p.

3.2 Допоміжна література

1. Коноваленко І.В. Програмування мовою C# 7.0 : навчальний посібник / Коноваленко І.В., Марущак П.О., Савків В.Б. – Тернопіль : Тернопільський національний технічний унверситет імені Івана Пулюя 2017 – 300 с.
2. Computer Science handbook Second Edition Editor-In-Chief Allen B. Tucker, CHAPMAN & HALL/CRC, 2004. - 2624p.
3. Кушниренко А.Г., Лебедев Г.В. Программирование для математиков: Учеб. пособие для вузов – М.: Наука, Гл. ред. физ.-мат. лит., 1988. – 384 с.
4. Кнут Д. Искусство программирования, том 1. М.: – «Вильямс», 2000 г.
5. Кнут Д. Искусство программирования, том 3. Сортировка и поиск. М.: – «Вильямс», 2000. – 832 с.

4. Електронні інформаційні ресурси

1. https://metanit.com/sharp/tutorial
2. https://docs.microsoft.com/en-us/
3. Електронний варіант лекцій.
4. Методичні вказівки до виконання лабораторних робіт та контрольних завдань в електронній формі.

5. Форма підсумкового контролю успішності навчання:

Іспит.

6. Методи діагностики успішності навчання

Діагностика знань студентів здійснюється з допомогою:
- усного поточного опитування;
- лабораторних робіт;
- індивідуальних контрольних завдань;

- екзаменаційних завдань.

Примітки:

1. Програма навчальної дисципліни розробляється вищим навчальним закладом. Програма навчальної дисципліни визначає місце і значення навчальної дисципліни, її загальний зміст та вимоги до знань і вмінь.
2. Програма навчальної дисципліни розробляється на основі освітньо-професійної програми.
3. Форма призначена для складання робочої програми навчальної дисципліни.
4. Вищими навчальними закладами можуть вноситися зміни до форми та змістового наповнення «Програми навчальної дисципліни» залежно від специфіки та профілю вищого навчального закладу.
5. Формат бланка – А4 (210×297 мм).