



Структури даних

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>15 Автоматизація та приладобудування¹</i>
Спеціальність	<i>153 Мікро- та наносистемна техніка</i>
Освітня програма	<i>Мікро- та наноелектроніка (для бакалаврів і магістрів), Мікро- та наносистемна техніка (для PhD)</i>
Статус дисципліни	<i>вибіркова</i>
Форма навчання	<i>очна(денна)</i>
Рік підготовки, семестр	<i>3 курс, весняний семестр</i>
Обсяг дисципліни	<i>4 кредити (120 годин)</i>
Семестровий контроль/ контрольні заходи	<i>залік</i>
Розклад занять	
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	Лектор: к.т.н., доц., Татарчук Д.Д., dmitry.tatarchuk@gmail.com , м. 0971521861 Комп'ютерний практикум: асистент, доктор філософії, Шевлякова Г.В., g.shevliakova-me@iit.kpi.ua , м. 0958563996
Розміщення курсу	https://meet.google.com/omh-jupk-zna?authuser=0&hs=179
Код курсу	<i>fnbiw75</i>

Програма навчальної дисципліни

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

Мета курсу:

- дати студентам знання основ функціонування апаратних та програмних засобів обчислювальної техніки, основ розробки прикладного програмного забезпечення сучасними програмними засобами;
- сформувати у студентів здатності до організації автоматизованого робочого місця на базі персональної ЕОМ, використання засобів обчислювальної техніки та програмного забезпечення у професійній діяльності, кваліфікованого використання програмного забезпечення призначеного для розробки програм, розробки програмного забезпечення для розв'язку типових задач професійного спрямування.
- розвинути у студентів уміння обирати та налагоджувати засоби обчислювальної техніки та програмного забезпечення у відповідності до професійних потреб, розробляти прикладне програмне забезпечення для розв'язку інженерних завдань професійної спрямованості, розробляти системне програмне забезпечення для мікропроцесорних систем.
- сформувати у студентів досвід практичного використання апаратних та програмних засобів обчислювальної техніки, програмування з використанням сучасних засобів та технологій.

¹ В полях Галузь знань/Спеціальність/Освітня програма:

Для дисциплін професійно-практичної підготовки зазначається інформація відповідно до навчального плану.
Для соціально-гуманітарних дисциплін вказується перелік галузей, спеціальностей, або «для всіх».

Дисципліна формує:

1. Загальні компетентності:

- ЗК 1. Здатність застосовувати знання у практичних ситуаціях;
- ЗК 2. Знання та розуміння предметної області та розуміння професійної діяльності;
- ЗК 5. Навички використання інформаційних і комунікаційних технологій;
- ЗК 6. Здатність вчитися і оволодівати сучасними знаннями.
-

2. Фахові компетентності:

- ФК 4. Здатність застосовувати відповідні наукові та інженерні методи, сучасні інформаційні технології і комп'ютерне програмне забезпечення, комп'ютерні мережі, бази даних та Інтернет-ресурси для розв'язання професійних задач в галузі мікро- та наносистемної техніки.

В результаті успішного засвоєння дисципліни здобувачі вищої освіти досягають таких програмних результатів навчання:

- ПРН 5. Використовувати інформаційні та комунікаційні технології, прикладні та спеціалізовані програмні продукти для розв'язання задач проектування та налагодження обладнання геліоенергетики, приладів фізичної та біомедичної електроніки;
- ПРН 17. Використовувати знання принципів і методів побудови та застосування сучасних інфокомунікаційних мереж, навички програмування та спеціалізовані програмні продукти для вирішення задач проектування та конструювання мікроелектронних інформаційних систем.

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

Дисципліна базується на курсі «Програмування та алгоритмічні мови»

Зміст навчальної дисципліни

Вступ до курсу «Алгоритми та структури даних»

Поняття алгоритму. Види запису алгоритмів

Розділ 1. Основи об'єктно орієнтованого програмування на C++

Тема 1.1. Основні поняття теорії об'єктно орієнтованого програмування

Тема 1.2. Особливості об'єктної моделі в C++.

Тема 1.3. Ініціалізація і знищення об'єктів. Конструктори та деструктори. Динамічний розподіл пам'яті.

Тема 1.4. Доступ до полів та методів класу. Статичні члени класу.

Тема 1.5. Особливості реалізації механізмів наслідування та поліморфізму в C++. Доступ до членів базових класів

Тема 1.6. Друзі класу

Тема 1.7. Шаблони класів

Тема 1.8. Перевантаження методів та операторів класів

Тема 1.9. Поток в C++, особливості роботи з файлами в C++

Разом за розділом 1

Розділ 2. Типові структури даних та алгоритми їх обробки

Тема 2.1. Статичні та динамічні масиви. Методи сортування масивів. Методи пошуку у масивах

Тема 2.2. Списки з одинарними та подвійними зв'язками, сортування списків, пошук у списках

Тема 2.3. Стеки та Деки

Тема 2.4. Черги прості та циклічні

Тема 2.5. Дерева та графи. Бінарні дерева. Сортування бінарних дерев. Пошук у бінарних деревах. AVL – дерева.

Розділ 3. Методи налагодження програм

Тема 3.1. Методи налагодження програм

2. Навчальні матеріали та ресурси

Базова література:

1. Татарчук, Д. Д. Інформатика [Електронний ресурс] : навчальний посібник / Д. Д. Татарчук, Ю. В. Діденко ; НТУУ «КПІ». – Електронні текстові дані (1 файл:). – Київ : НТУУ «КПІ» Вид-во «Політехніка», 2016. – 215 с.

2. «Алгоритми та структури даних» [Електронний ресурс] : методичні вказівки до виконання лабораторних робіт студентів факультету електроніки всіх форм навчання / НТУУ «КПІ» ; уклад. О. В. Мачулянський, Д. Д. Татарчук. – Електронні текстові дані (1 файл: 315 Кбайт). – Київ : «Аверс», 2008. – 40 с.

3. Моделювання засобами C++ [Електронний ресурс] : навчальний посібник / НТУУ «КПІ» ; уклад.: О. В. Мачулянський, Д. Д. Татарчук. – Електронні текстові дані (1 файл: 582,5 Кбайт). – Київ : АБЕРС, 2010.– 83 с.

Додаткові матеріали та ресурси:

1. Matthew H. Austern *Generic Programming and the STL Using and Extending the C++ Standard Template Library* Addison-Wesley, Reading, MA, 1998

2. Margaret A. Ellis, Bjarne Stroustrup *The Annotated C++ Reference Manual (ARM)* Addison-Wesley, Reading, MA, 1990

3. Scott Meyers *More Effective C++ 35 New Ways to Improve Your Programs and Designs* Addison-Wesley, Reading, MA, 1996

4. David R. Musser, Atul Saini *STL Tutorial and Reference Guide C++ Programming with the Standard Template Library* Addison-Wesley, Reading, MA, 1996.

5. Sedgewick, Robert; Wayne, Kevin (2011). *Algorithms (4th ed.)*. Addison-Wesley Professional. ISBN 978-0-321-57351-3.

6. Sedgewick, Robert (1998). *Algorithms, 3rd Edition, in C, Parts 1-4: Fundamentals, Data Structures, Sorting, and Searching*. Reading, MA: Addison-Wesley. ISBN 978-0201314526..

Навчальний контент

3. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття:

Лекція №1

Вступ до курсу «Алгоритми та структури даних»

Поняття алгоритму. Види запису алгоритмів

Лекція №2

Розділ 1. Основи об'єктно орієнтованого програмування на C++

Лекція №3

Тема 1.1. Основні поняття теорії об'єктно орієнтованого програмування

Лекція №4

Тема 1.2. Особливості об'єктної моделі в C++.

Лекція №5

Тема 1.3. Ініціалізація і знищення об'єктів. Конструктори та деструктори. Динамічний розподіл пам'яті.

Лекція №6

Тема 1.4. Доступ до полів та методів класу. Статичні члени класу.

Лекція №7

Тема 1.5. Особливості реалізації механізмів наслідування та поліморфізму в C++. Доступ до членів базових класів

Лекція №8

Тема 1.6. Друзі класу

Лекція №9

Тема 1.7. Шаблони класів

Лекція №10

Тема 1.8. Перевантаження методів та операторів класів

Лекція №11

Тема 1.9. Потоки в C++, особливості роботи з файлами в C++

Лекція №12

Тема 2.1. Статичні та динамічні масиви. Методи сортування масивів. Методи пошуку у масивах

Лекція №13

Тема 2.2. Списки з одинарними та подвійними зв'язками, сортування списків, пошук у списках

Лекція №14

Тема 2.3. Стеки та Деки

Лекція №15

Тема 2.4. Черги прості та циклічні

Лекція №16

Модульна контрольна робота.

Лекція №17

Тема 2.5. Дерева та графи. Бінарні дерева. Сортування бінарних дерев. Пошук у бінарних деревах. AVL – дерева.

Лекція №18

Тема 3.1. Методи налагодження програм

Комп'ютерний практикум:

Заняття №1

Робота з файлами в C++.

Заняття №2

Робота з масивами та списками. Методи сортування та пошуку.

Заняття №3

Робота з масивами та списками. Методи сортування та пошуку (продовження).

Заняття №4

Робота зі списками. Динамічна зміна розмірів списків.

Заняття №5

Використання списків для збереження та відображення графічної інформації.

Заняття №6

Використання списків для збереження та відображення графічної інформації (продовження).

Заняття №7

Робота з бінарними деревами.

Заняття №8

Захист робіт та РГР.

Заняття №9

Залік.

4. Самостійна робота студента/аспіранта

Для стимуляції самостійної роботи студентів, заохочення їх до самовдосконалення та знайомства з новітніми інформаційними технологіями в кредитному модулі передбачено в

якості індивідуального завдання розрахунково-графічна робота за тематикою кредитного модуля. В роботі необхідно засобами об'єктно-орієнтованого програмування мови C++ реалізувати за індивідуальним завданням просту базу даних з реалізацією функцій створення бази даних, додавання записів, вилучення записів, пошуку інформації за заданим критерієм. На виконання завдання передбачено 4 години самостійної роботи.

Також на самостійну роботу винесено вивчення наступного теоретичного матеріалу:
Повторне використання коду – 4 години.

Конструктор за замовчуванням – 4 години.

Доступ до полів та методів предків – 6 години.

Класи друзі класів – 6 години.

Файлові потоки. Класи *ifstream* та *ofstream* – 14 годин.

Метод Шелла – 4 години.

Підготовка до заліку – 18 годин.

Політика та контроль

5. Політика навчальної дисципліни (освітнього компонента)

Виконання всіх завдань є обов'язковою умовою допуску до заліку.

Перед комп'ютерним практикумом необхідно попередньо ознайомитись з завданням на дане заняття. Завдання повинне бути виконане і показане викладачеві згідно розкладу. В противному випадку знімаються рейтингові бали згідно вимог PCO.

Захист практичних робіт може здійснюватися на протязі всього семестру. На захист практичної роботи повинен бути оформлений протокол виконаної роботи. Протокол оформлюється за загальними правилами університету і повинен містити титульний аркуш, завдання і код програми. Процедура захисту складається з відповідей на запитання викладача за темою роботи. За неправильні відповіді або неправильне оформлення протоколу оцінка знижується згідно вимог PCO. За умови неправильної відповіді більш ніж на третину запитань захист не зараховується.

Розрахункова робота також повинна бути захищена. На захист повинна бути пред'явлена виконана робота в електронному та друкованому варіантах. Друкований варіант повинен бути оформлений згідно вимог університету і повинен містити титульний аркуш, завдання, програмний код, опис всіх змінних, функцій і об'єктів, які використано у коді. Процедура захисту складається з відповідей на запитання викладача за темою роботи. За неправильні відповіді або неправильне оформлення роботи оцінка знижується згідно вимог PCO. За умови неправильної відповіді більш ніж на третину запитань захист не зараховується.

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

З метою контролю процесу засвоєння учбового матеріалу у курсі передбачено модульну контрольну роботу за матеріалом першого та другого розділів. Оцінювання контрольної роботи здійснюється згідно рейтингової системи. За неправильні відповіді бали не зараховуються, за неточні або не повні відповіді бали знижуються.

Календарний контроль: провадиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу. Для успішного проходження першого календарного контролю: студент повинен набрати не менше 20% балів від максимального сумарного рейтингу протягом семестру. Для успішного проходження другого календарного контролю студент повинен набрати не менше 40% балів від максимального рейтингу.

Семестровий контроль здійснюється у вигляді заліку.

Студенти, які набрали протягом семестру необхідну для позитивної оцінки кількість балів мають можливість:

- не складати залік, а отримати оцінку „автоматом” відповідно до набраного рейтингу з дисципліни;

- складати залік з метою підвищення оцінки.

У разі отримання на заліку оцінки нижчої, ніж за рейтингом, за студентом не зберігається оцінка отримана „автоматом”.

Студенти, семестровий рейтинг яких відповідає оцінці „незадовільно”, зобов'язані скласти залік.

Студенти, які за семестровим рейтингом не допущені до заліку з цієї дисципліни, зобов'язані підвищити його до рівня не менше 60%.

Оцінка визначається за сумою набраних рейтингових балів відповідно до системи розрахунку шкали рейтингу.

Рейтинговий бал студента нараховується за наступними правилами;

1. Виконання практичних робіт

Ваговий бал 3. Максимальна кількість балів $3 \times 5 = 15$. Бали нараховуються в разі правильного виконання завдань передбачених практичними роботами.

2. Захист практичних робіт

Ваговий бал 5. Максимальна кількість балів $5 \times 5 = 25$. Бали нараховуються за результатами захисту робіт. Захист полягає у відповіді на 5-запитань викладача. За правильну відповідь на запитання нараховується 1 бал. За неправильну відповідь бали не нараховуються. Якщо не зараховано три або більше запитань захист не зараховується. В цьому випадку студент повинен підготуватися і прийти на повторний захист.

3. Модульна контрольна робота – максимальний бал 30.

4. Розрахунково-графічна робота – максимальний бал 30.

Система рейтингових (вагових) балів

№ п/п	Заняття, що підлягають рейтинговій оцінці	Загальна кількість	Макс. бал	Число балів на відмінно
1.	Комп'ютерний практикум: своєчасне виконання захист	5	3	15
		5	5	25
2.	Модульна контрольна робота	1	30	30
3.	РГР	1	30	30
	Рейтинг за курс, R			100

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

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

7. Додаткова інформація з дисципліни (освітнього компонента)

Перелік запитань для семестрового контролю.

1. З яких етапів складається обчислювальний експеримент?
2. Які вимоги висуваються до моделей?
3. Що таке абстрактний тип даних?
4. В чому полягає основна ідея ООП?
5. Які основні принципи ООП?
6. Що таке абстракція в контексті ООП?

7. Що таке інкапсуляція?
8. Що таке успадкування?
9. Що таке поліморфізм?
10. Що означає повторне використання коду в контексті ООП?
11. Які переваги ООП?
12. Що таке клас?
13. Що таке поле класу?
14. Що таке метод класу?
15. Як визначити клас у програмі?
16. Як описати структуру класу?
17. З яких розділів складається структура класу?
18. Що означає заголовок `public` ?
19. Що означає заголовок `private` ?
20. Що означає заголовок `protected` ?
21. Як описати методи класу?
22. Що означає директива `inline` при визначенні методів класу?
23. Що таке об'єкт?
24. Як об'явити об'єкт у програмі?
25. Що таке конструктор?
26. Навіщо потрібен конструктор?
27. Як формується ім'я конструктора?
28. Що таке деструктор?
29. Як формується ім'я деструктора?
30. Навіщо потрібен деструктор?
31. Що таке конструктор за замовчуванням?
32. Скільки конструкторів може мати клас?
33. Скільки деструкторів може мати клас?
34. Для чого використовується оператор `new`?
35. Для чого використовується оператор `delete`?
36. Яким чином здійснюється доступ до відкритих полів та методів класу?
37. Яким чином здійснюється доступ до відкритих полів та методів класу?
38. Яким чином здійснюється доступ до закритих полів та методів класу?
39. Яким чином здійснюється доступ до захищених полів та методів класу?
40. Що представляє собою оператор прямого вибору?
41. Що представляє собою оператор непрямого вибору?
42. Що таке статичне поле класу, які його властивості?
43. Що таке множинне успадкування?
44. Для чого використовуються друзі класів?
45. Що може бути другом класу?
46. Як оголосити функцію чи клас другом класу?
47. Чому при оголошенні класу другом іншого класу використовують випереджуючий неповний опис?
48. Що таке шаблони класів?
49. У яких випадках використовуються шаблони класів?
50. Навіщо використовують перевантаження операторів для класів?
51. У які способи можна реалізовувати перевантаження операторів для класів?
52. Що таке масив?
53. Які бувають масиви?
54. Чим статичний масив відрізняється від динамічного?
55. Які основні операції виконуються над масивами?
56. Що таке сортування?
57. Як класифікуються методи сортування?

58. Які методи називають внутрішніми?
59. Які методи називають зовнішніми?
60. Який алгоритм сортування методом обміну?
61. Який алгоритм сортування методом вибору?
62. Який алгоритм сортування методом включення?
63. Який алгоритм методу прямого пошуку?
64. Який алгоритм методу бінарного пошуку?
65. Що таке список?
66. Які бувають списки?
67. Які основні операції, що виконуються над списками?
68. Які методи сортування використовуються для списків?
69. Які методи пошуку використовуються для списків?
70. Що таке стек?
71. Для чого використовуються стеки?
72. Які можливі варіанти програмної реалізації стеку?
73. Що таке черга?
74. Для чого використовують черги?
75. Які бувають черги?
76. В чому полягає відмінність між простими та циклічними чергами?
77. Що таке бінарне дерево?
78. Що таке вузол (лист) дерева?
79. Що таке корінь дерева?
80. Що таке вітка (піддерево) дерева?
81. Що таке рівень?
82. Який вузол називається кінцевим(термінальним)?
83. Що таке висота дерева?
84. Що таке проходження дерева?
85. Які є методи проходження дерева?
86. Який порядок проходження дерева при послідовному методі?
87. Що таке збалансоване дерево?
88. Що таке AVL-дерево?
89. Що таке коефіцієнт збалансованості?
90. Які значення може приймати коефіцієнт збалансованості для AVL-дерева?
91. Що таке висота вузла?
92. Яким чином може здійснюватись балансування AVL-дерева?
93. Які види обертань використовують для балансування дерев?
94. Що таке граф?
95. Який граф називається напрямленим?
96. Що таке шлях? Чим він відрізняється від ланцюга?
97. Який граф називається зв'язним?
98. Який граф називається k-зв'язним?
99. Що таке зважений граф?
100. Що таке мережа?
101. Що таке цикл?
102. Який граф називається повним?
103. Що таке матриця суміжності?
104. Що таке вагова матриця?
105. Які характерні риси матриці суміжності або вагової матриці для не напрямленого графа?
106. З яких етапів складається процес налагодження програми?
107. Які основні методи використовують для налагодження програми?
108. В чому полягає суть методу журналювання?

109. В чому полягає суть методу трансляції коду?

За наявності сертифікатів проходження дистанційних чи онлайн курсів за відповідною тематикою можливе зарахування курсу повністю або частково за результатами співбесіди викладача зі студентом за темою курсу.

Робочу програму навчальної дисципліни (силабус):

Складено доцентом кафедри мікроелектроніки, д.т.н., доц., Татарчуком Д.Д.

Ухвалено кафедрою мікроелектроніки (протокол № 22 від 23.06.2023)

Погоджено Методичною комісією факультету² (протокол № 06/23 від 29.06.2023)

² Методичною радою університету – для загальноуніверситетських дисциплін.