



АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ:

Частина 2. СТРУКТУРИ ДАНИХ І АЛГОРИТМИ ЇХ ОБРОБКИ Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>122 Комп'ютерні науки</i>
Освітня програма	<i>Комп'ютерні технології в біології та медицині</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>Очна(денна)</i>
Рік підготовки, семестр	<i>1 курс, весняний семестр; 2 курс, осінній семестр</i>
Обсяг дисципліни	<i>3 кредити ЄКТС / 90 год. (28 год. – лекції, 26 год. – комп'ютерні практикуми, 36 год. – СР)</i>
Семестровий контроль/ контрольні заходи	<i>Освітній компонент 1: залік, модульна контрольна робота (МКР) Освітній компонент 2: залік, МКР</i>
Розклад занять	<i>http://rozklad.kpi.ua</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: к.т.н. Алхімова Світлана Миколаївна, контактний телефон: +380674045083, e-mail: alkhimova.svitlana@iit.kpi.ua</i>
Розміщення курсу	<i>Посилання на дистанційний ресурс Google classroom: https://classroom.google.com/c/NjIwODUwNzIwNzQ0</i>

Програма навчальної дисципліни

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

Дисципліна «Алгоритмізація та програмування» складається з двох освітніх компонентів:

Частина 1 – Організація програм мовою високого рівня C++;

Частина 2 – Структури даних і алгоритми їх обробки.

Предметом навчальної дисципліни є система здатностей та умінь із основ алгоритмізації задач в предметній галузі та створення програмних застосунків мовою програмування C++, які необхідні під час виконання обов'язків, виробничих функцій та типових задач діяльності фахівця.

Метою навчальної дисципліни є засвоєння студентами необхідних знань з програмування мовою високого рівня C++, а також формування у студентів цілісного уявлення про розробку алгоритмів для вирішення задач у предметній галузі та базової підготовки в галузі використання засобів обчислювальної техніки, а також підготовки студентів до виконання науково-дослідної роботи на старших курсах на професійному рівні.

Програмні результати навчання.

Студенти після засвоєння навчальної дисципліни мають набути наступні компетентності.

Загальні компетентності

ЗК 1 Здатність до абстрактного мислення, аналізу та синтезу

ЗК 2 Здатність застосовувати знання у практичних ситуаціях

ЗК 3 Знання та розуміння предметної області та розуміння професійної діяльності

ЗК 6 Здатність вчитися і оволодівати сучасними знаннями

Фахові компетентності

ФК 3 Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

ФК 8 Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

ФК 18 Здатність проектувати та реалізовувати програмне забезпечення для вирішення різноманітних задач в біології та медицині із застосуванням різних мов програмування та сучасних бібліотек крос-платформного програмування, проводити тестування на декількох апаратних платформах, впроваджувати і підтримувати роботу інформаційних систем на сучасних платформах.

Студенти після засвоєння навчальної дисципліни мають продемонструвати наступні програмні результати навчання

ПР 1 Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук

ПР 5 Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.

ПР 9 Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

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

В структурно-логічній схемі програми підготовки фахівця навчальна дисципліна «Алгоритмізація та програмування. Частина 1-2» входить до переліку нормативних дисциплін, циклу професійної підготовки.

Пререквізити. Навчальна дисципліна «Алгоритмізація та програмування» базується на раніше засвоєній навчальній дисципліні «Основи інформатики та програмування».

Постреквізити. Навчальна дисципліна забезпечує наступну дисципліну «Об'єктно-орієнтоване програмування», а також є основою для підготовки дипломних робіт за спеціальністю та в подальшій практичній роботі за фахом.

3. Зміст навчальної дисципліни

Тема 1. Робота з типами даних. Поняття абстрактних типів даних (контейнерів)

Ієрархія типів мови C++. Автоматичне (неявне) та явне перетворення типів; формат та правила використання операторів явного перетворення типів в мові C++. Перелічуваний тип даних (перерахування); створення псевдонімів до типу, використовуючи ключове слово using та використовуючи специфікатор typedef. Абстрактні типи даних; особливості статичних та динамічних абстрактних типів даних.

Тема 2. Статичні контейнери та алгоритми їх обробки

Формат визначення та особливості використання масивів у мові C++. Переваги та недоліки використання масивів. Доступу до елементів масиву під час використання операції індексування масив та під час використання покажчиків. Масиви як аргументи функцій. Рядки як масиви елементів символьного типу. Особливості роботи зі статичними та динамічними масивами; алокація та звільнення пам'яті під час роботи з масивами. Алгоритми пошуку елементів в масиві: лінійний пошук, бінарний (дихотомійний) пошук. Алгоритми

сортування: бульбашкове сортування, коктейльне сортування, сортування гребінцем, швидке сортування, сортування простими вставками, сортування Шелла, сортування злиттям, сортування вибором, пірамідального сортування.

Тема 3. Файлові структури даних

Файл як структура даних; текстові файли; бінарні файли. Засоби керування положенням покажчика файлу; операції відкриття та закриття файлів; основні функції для роботи із файлами. Алгоритми для реалізації базових операцій над файловими структурами даних. Роботи з буфером під час обміну даними з зовнішніми носіями інформації.

Тема 4. Структури як тип користувача

Основні засади створення типів користувача. Структури як тип даних користувача; особливості роботи із структурами. Алгоритми для реалізації базових операцій над структурами; операції доступу до компонентів структури; масиви структур.

Тема 5. Динамічні контейнери та алгоритми їх обробки

Однозв'язні списки; двозв'язні списки; стек; черга; двостороння черга (дек). Визначення дерева; двійкове дерево; двійкове дерево пошуку. Класифікація дерев за максимальною кількістю гілок (бінарні, тернарні, n-гілок) і за висотою піддерева (збалансовані, незбалансовані). Реалізація бінарного дерева пошуку, операції вставки вузла, видалення вузла, пошуку цільового вузла. Визначення графа, різновиди. Реалізації графа як абстрактного типу даних (матриця суміжності, матриця інцидентності, список суміжності, список ребер). Алгоритми обходу графа (пошуку в ширину, пошук в глибину) та пошуку найкоротшого шляху в графі (алгоритм Дейкстри). Визначення хеш-таблиці та хеш-функції; методи реалізації хеш-функції; методи вирішення колізій під час хешування; оцінка якості хеш-функції.

4. Навчальні матеріали та ресурси

Базова

1. Ришковець, Ю. В., Висоцька, В. А. Алгоритмізація та програмування. Частина 1: навч. посіб. – Львів: «Новий Світ-2000», 2018. – 337 с.
2. Ришковець, Ю. В., Висоцька, В. А. Алгоритмізація та програмування. Частина 2: навч. посіб. – Львів: «Новий Світ-2000», 2020. – 320 с.
3. Веклич, Р. А. Вступ до програмування мовою C++. Структури даних: навч. посіб. / Р. А. Веклич, Т. О. Карнаух, А. Б. Ставровський – К. : ВПЦ "Київський університет", 2018. – 99 с.
4. Алгоритми, дані і структури : навч. посіб. / В.М. Ільман, О.П. Іванов, Л.О. Панік. Дніпропет. нац. ун-т залізн. трансп.ім. акад. В. Лазаряна. – Дніпро, 2019. – 134 с.
5. Алгоритмізація та програмування мовою високого рівня C++ : Комп'ютерний практикум : навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки та інформаційні технології», спеціалізації «Інформаційні технології в біології та медицині» / С. М. Алхімова. – К.: Вид-во «Політехніка», КПІ ім. Ігоря Сікорського, 2018. – 156 с.

Допоміжна

1. Вступ до програмування мовою C++. Організація даних : навч. посіб. / Т. О. Карнаух, Ю. В. Коваль, М. В. Потієнко, А. Б. Ставровський. – К. : Видавничо-поліграфічний центр "Київський університет", 2015. – 151 с.
2. Stroustrup, B. C++ Programming Language / Bjarne Stroustrup. – [3rd ed.]. – Boston, MA : Addison Wesley Professional, 2000. – 1030 p.
3. Stroustrup, B. Programming : principles and practice using C++ / Bjarne Stroustrup. – [2nd ed.]. – Boston, MA : Addison Wesley Professional, 2014. – 1312 p.
4. Davidson, J. Beautiful C++: 30 Core Guidelines for Writing Clean, Safe, and Fast Code / J. Davidson, K. Gregory. – Boston, MA : Addison Wesley Professional, 2021. – 352 p.
5. Meyers, S. Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14 / Scott Meyers. – Sebastopol, California : O'Reilly Media, Incorporated, 2014. – 334 p.
6. Алгоритмізація та програмування. Програмування мовою високого рівня C++ : метод. вказівки до викон. лаб. робіт для напряму підгот. 6.050101 «Комп'ютерні науки» » / Уклад. С. М. Алхімова. – К.: НГУУ «КПІ», 2015. – 108 с.
7. C++. Алгоритмізація та програмування: підручник / О.Г. Трофименко, Ю.В. Прокоп, Н.І. Логінова, О.В. Задерейко. – [2-ге вид. перероб. і доповн]. – Одеса : Фенікс, 2019. – 477 с.
8. Ковалюк, Т. В. Алгоритмізація та програмування: Підручник. – Львів: Магнолія 2006, 2013. – 400 с.

9. Васильєв, О.М. Програмування на С++ в прикладах і задачах : навч. посібник / О. М. Васильєв. – Київ : Ліра-К, 2020. – 382 с.
10. Шаховська, Н. Б. Алгоритми і структури даних : навч. посібник / Н. Б. Шаховська, Р. О. Голошук. - Львів : Магнолія 2006, 2018. – 214 с.
11. Алгоритми та структури даних : навч. посібник / Т. О. Коротєєва. – Львів : Видавництво Львівської політехніки, 2014. – 280 с.
12. Азарян, А. А. Основи алгоритмізації та програмування : навч. посіб. / А.А. Азарян, Н. О. Карабут, Т.П. Козикова, О.Г. Рибальченко, А.А. Трачук, Н.Н. Шаповалова. – Кривий Ріг: Вид-во ОксанПринт, 2014. – 308 с.
13. Грицюк, Ю. І., Рак, Т. Є. Програмування мовою С++ : навч. посіб. – Львів : Вид-во Львівського ДУ БЖД, 2011. – 292с.
14. Щедріна, О. І. Алгоритмізація та програмування процедур обробки інформації : навч. посіб. – К.: КНЕУ, 2001. – 240 с.
15. Ткачук, В. М. Алгоритми і структура даних: навч. посіб / В.М.Ткачук. - Івано-Франківськ : Вид-во Прикарпатського національного університету імені Василя Стефаника, 2016. – 286 с.
16. Алгоритми та структури даних: конспект лекцій. Частина 1. Структури даних / упоряд.: О. Д. Воробйов, Л. В. Глазунов. Одеса : ОНАЗ ім.О.С. Попова, 2017. 48 с.
17. Алгоритми та структури даних: конспект лекцій. Частина 2. Алгоритми пошуку,стиснення даних, внутрішнього та зовнішнього сортування, алгоритми на графах / упоряд.: О. Д. Воробйов, Л. В. Глазунов. Одеса : ОНАЗ ім. О.С. Попова, 2017. 52 с
18. Knuth, D. E. The Art of Computer Programming / Donald E. Knuth / Vol. 1: Fundamental Algorithms. – Boston, MA : Addison Wesley Professional, 1997. – 672 p.
19. Knuth, D. E. The Art of Computer Programming / Donald E. Knuth / Vol. 3: Sorting and Searching. – Boston, MA : Addison Wesley Professional, 1998. – 513 p.
20. Cormen, T. H. Introduction to Algorithms / Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. – Cambridge, Massachusetts : The MIT Press, 2009. – 1292 p.

Інформаційні ресурси

1. Інтернет-портал організаційно-методичного забезпечення дистанційних олімпіад з програмування для обдарованої молоді навчальних закладів України : [Електронний ресурс]. – Режим доступу : <http://www.e-olimp.com.ua/ua>
2. International Standard ISO/IEC 14882:2014(E) – Programming Language C++ : [Електронний ресурс]. – Режим доступу : <https://isocpp.org/std/the-standard>
3. MS Visual Studio : [Електронний ресурс]. – Режим доступу: <http://visualstudio.com>
4. MSDN : [Електронний ресурс]. – Режим доступу: <http://msdn.microsoft.com>

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента) і самостійна робота студента

Лекції

Лекція 1. Вступна

Питання, що розглядаються:

- 1) структура даних: визначення і сутність поняття;
- 2) загальний огляд алгоритмів, які працюють із структурами даних.

Література:

- основна – [1 (С. 19, 383), 3];
 додаткова – [10, 11, 15-17].

Завдання на СРС:

вивчити конспект лекції; ознайомитися із списком рекомендованої літератури; знати визначення і сутність основних понять; ознайомитися із вимогами до виконання комп'ютерних, структурою звіту з комп'ютерного практикуму.

Лекція 2. Особливості роботи з типами даних. Абстрактні типи даних (контейнери)

Питання, що розглядаються:

- 1) приведення типів і ієрархія стандартних типів мови С++;
- 2) випадки неявного приведення типів;
- 3) явне приведення типів;
- 4) перелічуваний тип даних (перерахування);
- 5) створення псевдонімів до типу;

б) абстрактні типи даних (контейнери).

Література:

основна – [1 (С. 19, 383), 3];

додаткова – [10, 11, 15-17].

Завдання на СРС:

вивчити конспект лекції; вивчити ієрархія типів операндів мови С++; визначити випадки, коли виконується автоматичне (неявне) перетворення типів; вивчити формат та правила використання операторів явного перетворення типів в мові С++; форма об'явлення та особливості використання перелічуваного типу даних (перерахування); особливості створення псевдонімів до типу, використовуючи ключове слово using та використовуючи специфікатор typedef; проаналізувати призначення використання абстрактних типів даних; знати особливості статичних та динамічних абстрактних типів даних.

Лекція 3. Масиви

Питання, що розглядаються:

- 1) масиви як абстрактний тип даних;
- 2) ініціалізація масивів;
- 3) доступ до елементів масиву;
- 4) функції з параметрами у вигляді масивів.

Література:

основна – [1 (С. 91-104), 2];

додаткова – [1-5, 8, 9, 13].

Завдання на СРС:

вивчити конспект лекції; вивчити формат визначення масиву мовою С++, поняття розміру та розмірності масиву; проаналізувати переваги та недоліки використання масивів; проаналізувати особливості доступу до елементів масиву під час використання операції індексування масив та під час використання покажчиків; визначити особливості роботи з функціями, що мають параметри у вигляді масивів.

Лекція 4. Особливості роботи з масивами

Питання, що розглядаються:

- 1) управління пам'яттю в мові С++;
- 2) організація розподілення пам'яті виконуваної програми;
- 3) масиви із статично і динамічно оголошеними розмірами;
- 4) приклади алокації та звільнення пам'яті під час роботи з динамічними масивами.

Література:

основна – [1 (С. 170-189, 123-137) - 3];

додаткова – [1-5, 8, 9, 13].

Завдання на СРС:

вивчити конспект лекції; знати оператори для алокації та звільнення пам'яті в мові С++; знати організацію розподілення пам'яті виконуваної програми, особливості різних сегментів пам'яті; визначити особливості роботи зі статичними та динамічними масивами; знати як виконати алокацію та звільнення пам'яті під час роботи з одновимірними, двовимірними, тривимірними динамічними масивами; вивчити алгоритми пошуку елементів в масиві: лінійний пошук, бінарний (дихотомічний) пошук.

Лекція 5. Рядки

Питання, що розглядаються:

- 1) поняття нуль-термінованого рядка;
- 2) однобайтові, розширені, багатобайтові нкль-терміновані рядки;
- 3) функції для класифікації символів із стандартної бібліотеки;
- 4) функції для маніпуляцій із символами із стандартної бібліотеки;
- 5) вбудовані функції обробки нуль-термінованих рядків.

Література:

основна – [1 (С. 91-104), 2];

додаткова – [1-5, 8, 9, 13].

Завдання на СРС:

вивчити конспект лекції; визначити особливості роботи з рядками, як масивами елементів символічного типу; знати різницю між однобайтовими, розширеними та багатобайтовими нуль-термінованими рядками; знати функції для класифікації символів із стандартної бібліотеки; знати функції для маніпуляцій із символами із стандартної бібліотеки; знати вбудовані функції обробки нуль-термінованих рядків.

Лекція 6. Пошук

Питання, що розглядаються:

- 1) алгоритм лінійного пошуку елементів в масиві;
- 2) алгоритм бінарного пошуку елементів в масиві.

Література:

основна – [1 (С. 170-189, 123-137) - 3];

додаткова – [1-5, 8, 9, 13].

Завдання на СРС:

вивчити конспект лекції; вивчити алгоритми пошуку елементів в масиві: лінійний пошук, бінарний (дихотомічний) пошук.

Лекція 7. Алгоритми сортування масивів

Питання, що розглядаються:

- 1) огляд алгоритмів сортування;
- 2) бульбашкове сортування;
- 3) коктейльне сортування;
- 4) сортування гребінцем;
- 5) швидке сортування;
- 6) сортування простими вставками;
- 7) сортування Шелла;
- 8) сортування злиттям;
- 9) сортування вибором;
- 10) пірамідальне сортування.

Література:

основна – [3];
додаткова – [1-5, 8, 9, 13].

Завдання на СРС:

вивчити конспект лекції; вивчити основи сортування даних; вивчити типові алгоритми з сортування даних масивів різної розмірності; знати алгоритми бульбашкового, коктейльного сортування, сортування гребінцем, швидкого сортування, сортування простими вставками, сортування Шелла, злиттям, вибором та пірамідального сортування мовою програмування C++.

Лекція 8. Файлові структури даних

Питання, що розглядаються:

- 1) поняття файлу;
- 2) класифікація файлів;
- 3) особливості роботи з файлами;
- 4) поняття буферизації.

Література:

основна – [1 (С. 347-354), 4];
додаткова – [1- 5].

Завдання на СРС:

вивчити конспект лекції; проаналізувати призначення використання файлових структур; вивчити засоби керування положенням покажчика файлу; визначити особливості операцій відкриття та закриття файлів мовою програмування високого рівня C++; ознайомитися з основними функціями для роботи із файловими структурами текстового та бінарного типів; визначити та вивчити алгоритми для реалізації базових операцій над файловими структурами даних; вивчити особливості роботи з буфером під час обміну даними з зовнішніми носіями інформації.

Лекція 9. Структури як тип користувача

Питання, що розглядаються:

- 1) поняття типу користувача;
- 2) визначення структури в програмному коді;
- 3) доступ до компонентів структури;
- 4) масив структур.

Література:

основна – [1 (С. 137-152), 4];
додаткова – [1-5, 8, 9, 13].

Завдання на СРС:

вивчити конспект лекції; розглянути основні засади створення типів користувача; проаналізувати призначення використання структур у програмному коді; визначити та вивчити алгоритми для реалізації базових операцій над структурами; знати специфіку роботи із операціями доступу до компонентів структури; знати особливості роботи із масивами структур.

Лекція 10. Списки

Питання, що розглядаються:

- 1) зв'язані списки;
- 2) однозв'язні списки;
- 3) двозв'язні списки;
- 4) стек;
- 5) черга;
- 6) двостороння черга (дек).

Література:

основна – [1 (С. 267-287), 2];
додаткова – [6, 7, 12, 14, 18-20].

Завдання на СРС:

вивчити конспект лекції; проаналізувати призначення використання стеку, черги, списку; визначити та вивчити алгоритми для реалізації базових операцій над стеком, чергою, списком.

Лекція 11. Дерева

Питання, що розглядаються:

- 1) визначення дерева;
- 2) двійкове дерево;
- 3) двійкове дерево пошуку;
- 4) додавання елементів у двійкове дерево пошуку;
- 5) видалення елементів у двійковому дереві пошуку;
- 6) найпростіша функція виведення елементів дерева на екран.

Література:

основна – [1 (С. 287-296); 2];
 додаткова – [6, 7, 12, 14, 18-20].

Завдання на СРС:

вивчити конспект лекції; знати дерева, як тип контейнера; знати, що дерево являє собою набір вузлів з одним коренем і гілками, які також є деревами, пояснювати на прикладах основні означення; проаналізувати призначення використання двійкових дерев та двійкових дерев пошуку; знати класифікацію дерев: за максимальною кількістю гілок (бінарні, тернарні, n-гілок), за висотою піддерева (збалансовані, незбалансовані); знати визначення бінарних дерев пошуку (правила бінарних дерев пошуку і визначення того, чи задовольняє дерево цим правилам); знати реалізацію бінарного дерева пошуку, операції вставки вузла, видалення вузла.

Лекція 12. Алгоритми обробки елементів двійкового дерева пошуку

Питання, що розглядаються:

- 1) пошук вузлів у двійковому дереві пошуку;
- 2) горизонтальний і вертикальний обходи двійкового дерева пошуку;
- 3) прямий, зворотний, симетричний порядки вертикального обходу двійкового дерева пошуку;
- 4) збалансовані двійкові дерева пошуку.

Література:

основна – [1 (С. 287-296); 2];
 додаткова – [6, 7, 12, 14, 18-20].

Завдання на СРС:

вивчити конспект лекції; знати реалізацію операції пошуку цільового вузла у двійковому дереві пошуку; знати правила виконання горизонтального та вертикального обходів двійкового дерева пошуку; вміти виконати реалізацію прямого, зворотного, симетричного порядків вертикального обходу двійкового дерева пошуку; знати, що таке збалансовані двійкові дерева пошуку, алгоритми балансування.

Лекція 13. Графи

Питання, що розглядаються:

- 1) визначення графа, різновиди;
- 2) реалізації графа як матриці суміжності;
- 3) реалізації графа як матриці інцидентності;
- 4) реалізації графа як списку суміжності;
- 5) реалізації графа як списку ребер;
- 6) алгоритми обходу графа (пошуку в ширину, пошук в глибину);
- 7) пошуку найкоротшого шляху в графі (алгоритм Дейкстри).

Література:

основна – [1 (С. 297-354); 2];
 додаткова – [6, 7, 12, 14, 18-20].

Завдання на СРС:

вивчити конспект лекції; знати основні поняття теорії графів; вміти реалізувати граф як абстрактного типу даних через матрицю суміжності, матрицю інцидентності, список суміжності, список ребер; знати алгоритми обходу графа, реалізацію алгоритмів пошуку в ширину та пошук в глибину; знати алгоритми пошуку найкоротшого шляху в графі, реалізацію класичного алгоритму Дейкстри.

Лекція 14. Хеш-таблиці

Питання, що розглядаються:

- 1) поняття асоціативного масиву;
- 2) визначення хеш-таблиці та хеш-функції;
- 3) методи реалізації хеш-функції.
- 4) методи вирішення колізій під час хешування;
- 5) метод відкритої адресації (закрите хешування);
- 6) метод закритої адресації (відкрите хешування).

Література:

основна – [1 (С. 467-476), 3];
 додаткова – [6, 7, 12, 14, 18-20].

Завдання на СРС:

вивчити конспект лекції; знати поняття асоціативного масиву, можливі способи його реалізації; знати призначення використання хеш-таблиць, що таке хеш-функція, досконала хеш-функція; знати методи реалізації хеш-функцій; знати, що таке колізії під час хешування, чому вони відбуваються; знати методи вирішення колізій під час проведення хешування; знати алгоритми вирішення колізій методом відкритої адресації; знати недоліки використання метода відкритої адресації, як виконують опрацювання проблем під час використання цього методу; знати, як вирішуються колізії під час хешування методом закритої адресації.

Комп'ютерні практикуми

Основна ціль комп'ютерних практикумів полягає в оволодінні студентами мовою програмування високого рівня C++; вивченні та оволодінні засадами програмування для розробки та аналізу алгоритмів вирішення типових задач мовою програмування C++; отриманні навичок роботи (поглиблений рівень) в інтегрованому середовищі розробки програмного забезпечення Microsoft Visual Studio.

Комп'ютерний практикум 1. Відлагодження в Microsoft Visual Studio

Питання, що розглядаються:

- 1) функціональні можливості відладчика Microsoft Visual Studio під час розробки програм;
- 2) призначення властивостей налаштування програмного рішення середовища Microsoft Visual Studio;
- 3) налагоджування середовища для потреб обробки та відлагодження програми під час її виконання.

Література:

основна – [5];
додаткова – [1- 4, 18-20].

Завдання на СРС:

вивчити конспект лекції та ознайомитися з навчально-методичними матеріалами за темою заняття, щоб відповісти на питання: що таке відлагодження та для чого воно призначене; які різновиди помилок існують; які засоби відлагодження надає інтегроване середовище розробки програм Microsoft Visual Studio; які є гарячі клавіші для роботи з відладчиком надає інтегроване середовище розробки програм Microsoft Visual Studio, як їх налаштувати; перелічити можливості використання точок переривання як інструмента для відлагодження програми; які можливості існують для відстеження значень змінних під час відлагодження програми; чим команда виконання по кроках відрізняється від команди трасування; як задати додаткові умови для точок переривання; які саме додаткові умови можна встановлювати для точок переривання; як змінити значення змінної в процесі відлагодження програми?

Комп'ютерний практикум 2. Перелічуваний тип даних

Питання, що розглядаються:

- 1) ключове слово enum;
- 2) конфлікти імен під час використання перелічуваного типу даних;
- 3) використання просторів імен;
- 4) особливості програмної реалізації перерахувань; особливості програмної реалізації просторів імен.

Література:

основна – [3-5];
додаткова – [1- 4, 18-20].

Завдання на СРС:

вивчити конспект лекції та ознайомитися з навчально-методичними матеріалами за темою заняття, щоб відповісти на питання: які значення може приймати нумератор перерахування; як організувати код, щоб значення кожного наступного нумератора перерахування зменшувалося на одиницю; чи можуть бути всередині одного перерахування два нумератори з однаковими іменами; за яких умов можуть виникати конфлікти імен під час використання перелічуваного типу даних; як організувати код, щоб знати яка кількість нумераторів зберігається у конкретному перерахуванні; що є ідентифікаторами з глобального контексту програми; якими трьома способами можна організувати доступ до ідентифікаторів конкретного простору імен із коду, що розташований за його межами; що таке вкладений простір імен; за якої умови члени зовнішнього простору імен мають доступу до членів вкладеного простору; чи впливає використання простору імен на час існування змінної; вміти проводити програму реалізацію перерахувань і просторів імен

Комп'ютерний практикум 3. Одновимірні масиви

Питання, що розглядаються:

- 1) структурна організація одновимірних масивів;
- 2) способи доступу до елементів одновимірних масивів;
- 3) базові операції для роботи з одновимірними масивами;
- 4) алгоритмізація процесів з обробки одновимірних масивів;
- 5) особливості програмної реалізації визначення та використання одновимірних масивів;
- 6) особливості програмної реалізації доступу до елементів одновимірних масивів;
- 7) особливості програмної реалізації базових операцій для роботи з одновимірними масивами;
- 8) особливості програмної реалізації обробки одновимірних масивів.

Література:

основна – [5];
додаткова – [1-5].

Завдання на СРС:

вивчити конспект лекції та ознайомитися з навчально-методичними матеріалами за темою заняття, щоб відповісти на питання: що саме може бути використане у якості елементів масиву; що визначає індекс елемента масиву; які обмеження накладені на індекс масивів; чи може масив містити один елемент, взагалі не містити жодного; пояснити, чи можуть числа 0, 1.1, 2, 2.0 бути елементами одного масиву; на яку фактичну величину змінюється значення покажчика під час застосування до нього операції інкременту; який формат визначення одновимірного масиву в мові C++; пояснити механізм визначення розмірів

багатовимірному масиву всередині функції, що передається до неї у якості параметра; пояснити, чому у разі визначення масиву з ім'ям `array` можна звернутися до його другого елемента, використавши запис `*(arrayID + 1)`, а запис `*(arrayID++)` призведе до помилки; перелічити та пояснити, у чому полягають позитивні та негативні сторони використання масивів.

Завдання на СРС:

вивчити конспект лекції та ознайомитися з навчально-методичними матеріалами за темою заняття, щоб вміти проводити програму реалізацію визначення, використання та обробки одновимірних масивів.

Комп'ютерний практикум 4. Двовимірні масиви

Питання, що розглядаються:

- 1) особливості визначення та використання двовимірних масивів;
- 2) структурна організація двовимірних масивів;
- 3) способи доступу до елементів двовимірних масивів;
- 4) базові операції для роботи з двовимірними масивами та алгоритмами лінійного пошуку за умовою;
- 5) алгоритмізація процесів з обробки двовимірних масивів;
- 6) особливості програмної реалізації визначення та використання двовимірних масивів;
- 7) особливості програмної реалізації доступу до елементів двовимірних масивів;
- 8) особливості програмної реалізації базових операцій для роботи з двовимірними масивами та алгоритмами лінійного пошуку за умовою;
- 9) особливості програмної реалізації обробки двовимірних масивів.

Література:

основна – [4, 5];
додаткова – [1-5].

Завдання на СРС:

вивчити конспект лекції та ознайомитися з навчально-методичними матеріалами за темою заняття, щоб відповісти на питання: якими способами задаються двовимірні масиви в мові C++ і коли їх доцільно використовувати; в якій послідовності виконується обробка елементів багатовимірних масивів в мові програмування C++; поясніть із наведенням відповідного програмного коду, скільки необхідно операторів циклів для виведення на екран значень елементів двовимірного масиву; які значення будуть приймати елементи двовимірного статичного масиву за умови відсутності їх ініціалізації під час оголошення цього масиву; яка умова визначення елементів на головній (другорядній) діагоналі двовимірного масиву, навести відповідні приклади; яка умова визначення всіх елементів k-го стовпчика (рядка) двовимірного масиву, навести приклад; як організувати код для відображення всіх елементів двовимірного масиву на екрані у вигляді вирівняної таблиці; як можна під час виконання програми змінити розмір двовимірного масиву (кількість елементів в ньому), навести відповідний приклад; перелічити відмінності в роботі зі статичними та динамічними масивами; на прикладі пояснити, як змінити значення двовимірного масиву з використанням покажчиків для доступу до його елементів; вміти проводити програму реалізацію визначення, використання та обробки двовимірних масивів.

Комп'ютерний практикум 5. Рядки

Питання, що розглядаються:

- 1) структурна організація нуль-термінованих рядків;
- 2) способи доступу до елементів рядків;
- 3) базові операції для роботи з рядками;
- 4) особливості використання функцій обробки рядків мовою C++;
- 5) алгоритмізація процесів з обробки рядків;
- 6) особливості програмної реалізації визначення та використання рядків;
- 7) особливості програмної реалізації доступу до елементів рядків;
- 8) особливості програмної реалізації базових операцій для роботи з рядками;
- 9) особливості програмної реалізації обробки рядків.

Література:

основна – [4, 5];
додаткова – [1- 5].

Завдання на СРС:

вивчити конспект лекції та ознайомитися з навчально-методичними матеріалами за темою заняття, щоб відповісти на питання: дати визначення і пояснити, чому C-рядок по суті є структурою даних; для чого використовують наступні символні типи `char`, `wchar_t`, `char8_t`, `char16_t`, `char32_t`; чим відрізняється ініціалізація символу від ініціалізації рядка; наведіть можливі способи початкової ініціалізації рядка; пояснити, який символ використовують для визначення кінця рядка і чи може цей символ бути розміщений у середині символного масиву; навести приклади як використовуючи в кодї символної літери можна безпосередньо або за допомогою різних видів `escape`-послідовності (звичайної, вісімкової, шіснадцяткової) задати символ у рядку; пояснити, що таке універсальні імена символів, та навести приклади як і з якою метою вони можуть бути використані в кодї програми; яку функцію можна використовувати для зчитування з клавіатури рядка, що містить пробіли; як в кодї програми задати необроблений рядковий літерал і пояснити навіщо його використовують; пояснити, як будуть зберігатися у пам'яті декілька рядкових літералів, що розташовані в кодї програми один за одним і розділені лише символами пробілу або коментарями до коду; вміти проводити програму реалізацію визначення, використання та обробки рядків.

Комп'ютерний практикум 6. Алгоритми сортування

Питання, що розглядаються:

- 1) типові алгоритми з обробки даних масивів;
- 2) алгоритмізація сортування простим обміном, змішуванням, методом Шелла, простими вставками, вибором, злиттям, бінарного сортування, швидкого сортування та пірамідального сортування;
- 3) особливості програмної реалізації алгоритмів з обробки даних масивів;
- 4) особливості програмної реалізації алгоритмів сортування простим обміном, змішуванням, методом Шелла, простими вставками, вибором, злиттям, бінарного сортування, швидкого сортування та пірамідального сортування;
- 5) часова складність алгоритмів сортування (найгірша, середня, краща);
- 6) просторова складність алгоритмів сортування;
- 7) особливості програмної реалізації визначення часу проведення сортування масиву.

Література:

основна – [4, 5];
додаткова – [1- 5].

Завдання на СРС:

вивчити конспект лекції та ознайомитися з навчально-методичними матеріалами за темою заняття, щоб відповісти на питання: що таке сортування, яка його основна мета; чи впливає тип даних, що сортуються, на час виконання алгоритму сортування; чому не існує універсального алгоритму сортування; чим можна пояснити різноманітність алгоритмів сортування; як і що треба враховувати під час вибору алгоритму сортування; які на сьогоднішній день найефективніші методи сортування; чому алгоритми швидкого сортування не дають великого виграшу при малих розмірах масивів; за рахунок чого в алгоритмі швидкого сортування та його модифікаціях можна покращити їх продуктивність; що таке просторова складність і як її оцінити для алгоритмів сортування; чим відрізняється внутрішнє сортування від зовнішнього; вміти проводити програму реалізацію алгоритмів сортування простим обміном, змішуванням, методом Шелла, простими вставками, вибором, злиттям, бінарного сортування, швидкого сортування та пірамідального сортування; вміти проводити оцінку часової та просторової складності алгоритмів сортування.

Комп'ютерний практикум 7. Бінарні файли даних

Питання, що розглядаються:

- 1) 1) структурна організація бінарних файлів даних;
- 2) способи доступу до елементів бінарних файлів даних;
- 3) базові операції для роботи з бінарними файлами даних;
- 4) алгоритмізація процесів з обробки покажчика на файл;
- 5) особливості програмної реалізації визначення та використання бінарних файлів даних в мові C++;
- 6) особливості програмної реалізації доступу до елементів бінарних файлів даних;
- 7) особливості програмної реалізації обробки покажчика на файл.

Література:

основна – [4, 5];
додаткова – [1- 5].

Завдання на СРС:

вивчити конспект лекції та ознайомитися з навчально-методичними матеріалами за темою заняття, щоб відповісти на питання: що називається файлом і коли доцільно його використовувати; у чому різниця між текстовим та бінарним файлами; що таке логічний запис, як він використовується під час роботи із файлом і чому його розмір відрізняється від розміру блоку (найменшої одиниці даних, за допомогою якої виконується обмін між зовнішнім пристроєм та оперативною пам'яттю); з якою метою передбачені різні режими відкриття файлу і які з них забезпечують збереження його вмісту; що таке буфер і чому його використання підвищує швидкість передачі даних; коли змінюється вміст буфера; які відмінності в організації символного, строкового, блокового та форматowanego введення/виведення інформації у файл; що саме і за яких умов може повернути функція `forep`; чи може вміст текстового файлу після його занесення в область пам'яті виконуваної програми бути збережений до іншого файлу, який є бінарним; пояснити, навіщо необхідно закрити файл і чи можна відкрити ще раз той самий файл, якщо він не був попередньо закритий; вміти проводити програму реалізацію визначення, використання та обробки бінарних файлів даних.

Комп'ютерний практикум 8. Структури

Питання, що розглядаються:

- 1) структури як тип користувача;
- 2) способи доступу до елементів структур;
- 3) базові операції для роботи зі структурами;
- 4) алгоритмізація процесів з обробки елементів структур та елементів масиву структур;
- 5) особливості програмної реалізації визначення та використання структур мовою C++;
- 6) особливості програмної реалізації доступу до елементів структур;
- 7) особливості програмної реалізації обробки елементів структур та елементів масиву структур.

Література:

основна – [4, 5];
додаткова – [1- 4, 18-20].

Завдання на СРС:

вивчити конспект лекції та ознайомитися з навчально-методичними матеріалами за темою заняття, щоб відповісти на питання: що таке структура, для чого вона призначена; що таке вкладена структура; пояснити, чим відрізняється оголошення змінної структури окремо в кодї програми від оголошення, що слїдує одразу за оголошенням самої структури після фігурних дужок; на прикладї пояснити, як ініціалізувати поле структури, що є масивом; як оголосити покажчик на структуру; як ініціалізувати покажчик на структуру; які операції використовуються для доступу до елементів структури, в чому їх різниця; яким чином передаються структури до функцій; чи можна передавати в функцію масив структур; які проблеми можуть виникнути під час копіювання змінних структурного типу одна в одну; вміти проводити програму реалізацію визначення, використання та обробки структур.

Комп'ютерний практикум 9. Однозв'язні списки

Питання, що розглядаються:

- 1) структурна організація однозв'язних списків;
- 2) доступ до елементів однозв'язних списків;
- 3) базові операції для роботи із однозв'язними списками;
- 4) алгоритмізація процесів з обробки елементів однозв'язних списків;
- 5) особливості програмної реалізації визначення та використання однозв'язних списків в мові C++;
- 6) особливості програмної реалізації доступу до елементів однозв'язних списків;
- 7) особливості програмної реалізації обробки елементів однозв'язних списків.

Література:

основна – [4, 5];
додаткова – [1-5].

Завдання на СРС:

вивчити конспект лекції та ознайомитися з навчально-методичними матеріалами за темою заняття, щоб відповісти на питання: які переваги і недоліки використання однозв'язних списків у порівнянні із масивами; чим відрізняється пошук в масиві від пошуку в списку; у чому полягають особливості роботи з кільцевими однозв'язними списками; пояснити, який тип має мати вузол однозв'язного списку; у чому полягають принципові відмінності додавання елемента на першу і у будь-яку іншу позицію в однозв'язному списку; з якою метою необхідно виконувати перевірку на порожність під час обробки однозв'язного списку; пояснити, чи необхідно під час роботи із однозв'язними списками у окремій змінній зберігати їх розмір; чому при роботі з однозв'язним списком його обробку необхідно виконувати, починаючи з першого елемента списку; чим відрізняються перший і останній елементи від інших у однозв'язному списку; до яких змін у роботі програми призведе ситуація, в якій не виконується видалення зв'язного списку після закінчення роботи з ним; вміти проводити програму реалізацію визначення, використання та обробки однозв'язних списків.

Комп'ютерний практикум 10. Двозв'язні списки

Питання, що розглядаються:

- 1) структурна організація двозв'язних списків;
- 2) доступ до елементів двозв'язних списків;
- 3) базові операції для роботи із двозв'язними списками;
- 4) алгоритмізація процесів з обробки елементів двозв'язних списків;
- 5) особливості програмної реалізації визначення та використання двозв'язних списків мовою C++;
- 6) особливості програмної реалізації доступу до елементів двозв'язних списків;
- 7) особливості програмної реалізації обробки елементів двозв'язних списків.

Література:

основна – [4, 5];
додаткова – [1-5].

Завдання на СРС:

вивчити конспект лекції та ознайомитися з навчально-методичними матеріалами за темою заняття, щоб відповісти на питання: у чому полягають переваги і недоліки використання двозв'язних списків у порівнянні із однозв'язними; чому при роботі з двозв'язним списком його обробку можна виконувати, починаючи з будь-якого елемента списку; визначте набір допустимих операцій для роботи із лінійними двозв'язними списками; визначте операції, що за швидкістю виконання кращі у двозв'язних списках ніж у однозв'язних; у чому полягають принципові відмінності для роботи з однозв'язними та двозв'язними списками; чим відрізняється операція пошуку у однозв'язному і двозв'язному списках; які методи сортування використовують для роботи із двозв'язними списками; чи залежить час, витрачений на додавання елемента до двозв'язного списку, від кількості елементів у ньому; у чому полягає різниця між лінійними та кільцевими зв'язними списками; за рахунок чого робота зі зв'язними списками уповільнює виконання програми; вміти проводити програму реалізацію визначення, використання та обробки двозв'язних списків.

Комп'ютерний практикум 11. Стек та черга

Питання, що розглядаються:

- 1) структурна організація стека та черги;
- 2) доступ до елементів стеку та черги;
- 3) базові операції для роботи зі стеком та чергою;
- 4) алгоритмізація процесів з обробки елементів стеку та черги;
- 5) особливості програмної реалізації визначення та використання стеку та черги мовою C++;

- 6) особливості програмної реалізації доступу до елементів стеку та черги;
- 7) особливості програмної реалізації обробки елементів стеку та черги.

Література:

основна – [4, 5];
додаткова – [1-5].

Завдання на СРС:

вивчити конспект лекції та ознайомитися з навчально-методичними матеріалами за темою заняття, щоб відповісти на питання: у чому переваги і недоліки стеку по відношенню до черги; чим відрізняється звичайна черга від черги з пріоритетом; пояснити принцип функціонування дека, його обмежених різновидів; яке значення зберігає покажчик на стек, а яке покажчик на чергу; наведіть детальний опис операцій, які є обов'язковими для реалізації контейнера стек; наведіть детальний опис операцій, які є обов'язковими для реалізації контейнера черга; які операції окрім обов'язкових допустимі для контейнерів стек та черга; зазначити особливості реалізації стека та черги, що базуються на звичайному масиві; які змінні необхідні для реалізації операцій із стеком або чергою, що базуються на зв'язному списку; для моделювання яких реальних завдань зручно використовувати стек, а для яких чергу; вміти проводити програму реалізацію визначення, використання та обробки стека та черги.

Комп'ютерний практикум 12. Древа

Питання, що розглядаються:

- 1) структурна організація дерев;
- 2) способи доступу до елементів дерев;
- 3) базові операції для роботи з деревами;
- 4) алгоритмізація процесів з обробки дерев;
- 5) особливості програмної реалізації визначення та використання бінарних дерев пошуку мовою C++;
- 6) особливості програмної реалізації операції додавання елемента до бінарних дерев пошуку мовою C++;
- 7) особливості програмної реалізації обходів бінарних дерев пошуку мовою C++;
- 8) особливості програмної реалізації операції пошуку елемента та виведення всіх елементів бінарних дерев пошуку мовою C++;
- 9) особливості програмної реалізації видалення елементів бінарних дерев пошуку мовою C++;
- 10) особливості програмної реалізації операції видалення всього бінарного дерева пошуку мовою C++.

Література:

основна – [4, 5];
додаткова – [1-5].

Завдання на СРС:

вивчити конспект лекції та ознайомитися з навчально-методичними матеріалами за темою заняття, щоб відповісти на питання: що таке дерево, двійкове дерево, двійкове дерево пошуку; як і який саме допоміжний контейнер найкраще використати для нерекурсивної реалізації обходу дерева; поясніть на прикладах, який тип інформації зручно зберігати за допомогою дерев; пояснити, чому для дерев існують декілька схем виконання їх обходу і чому реалізація обходу через рекурсію є більш зручною; навести схему довільного двійкового дерева пошуку та вказати порядок прямого, зворотного та симетричного обходу; які алгоритми пошуку елементів зручно застосовувати під час роботи з деревами; чи можна побудувати повне двійкове дерево з елементів 5, 1, 2, 8, 6, 10, 3, 9, 4, 7, відповідь пояснити; що таке досконале дерево, чим воно відрізняється від повного; пояснити, у чому полягає перевага ідеально збалансованого дерева з огляду на організацію пошуку; поясніть принцип видалення елемента двійкового дерева у випадку, коли цей елемент не має нащадків, має одного нащадка, має двох нащадків; вміти проводити програму реалізацію визначення бінарних дерев пошуку в мові C++; вміти проводити програму реалізацію різних способів обходів бінарних дерев пошуку в мові C++, а також операції пошуку елемента та виведення всіх елементів бінарних дерев пошуку на екран; вміти проводити програму реалізацію видалення елементів бінарних дерев пошуку в мові C++.

Комп'ютерний практикум 13. Проведення МКР

Питання, що виносяться на МКР пов'язані із засвоєнням студентами наступних тем:

Тема 1. Робота з типами даних. Поняття абстрактних типів даних (контейнерів)

Тема 2. Статичні контейнери та алгоритми їх обробки

Тема 3. Файлові структури даних

Тема 4. Структури як тип користувача

Тема 5. Динамічні контейнери та алгоритми їх обробки

Література:

основна – [1-5];
додаткова – [1-20].

Завдання на СРС:

вивчити конспекти лекції; вивчити теоретичні відомості щодо структур даних та алгоритмів їх обробки.

6. Політика навчальної дисципліни (освітнього компонента)

Форми організації освітнього процесу, види навчальних занять і оцінювання результатів навчання регламентуються Положенням про організацію освітнього процесу в Національному технічному університеті України «Київському політехнічному інституті імені Ігоря Сікорського».

Правила відвідування занять. Відвідування є обов'язковим (за винятком випадків, коли існує поважна причина, наприклад, хвороба чи дозвіл працівників деканату). Якщо студент не може бути присутнім на заняттях, він все одно несе відповідальність за виконання завдань, що проводились в комп'ютерному класі.

Правила поведінки на заняттях. Під час виконання комп'ютерних практикумів студент може користуватися ноутбуком, мобільним телефоном або іншими пристроями для пошуку інформації на гугл-диску викладача чи в інтернеті. Проте під час лекційних занять та обговорення завдань комп'ютерних практикумів зазначеними раніше пристроями користуватися неможна. Це відволікає викладача і студентів групи та перешкоджає навчальному процесу. Якщо мається намір використовувати ноутбук або інший пристрій для аудіо- чи відеозапису, необхідно заздалегідь отримати дозвіл викладача. Під час лекційних занять заборонено відволікати викладача, усі питання, уточнення та ін. студенти задають в кінці лекції або відведений для цього час.

Виконання завдань контрольних заходів

Плагіат та інші форми нечесної роботи неприпустимі. Всі комп'ютерні студенти мають виконувати самостійно із використанням рекомендованої літератури й отриманих знань та навичок. Цитування в письмових роботах допускається тільки із відповідним посиланням на авторський текст. Недопустимі підказки і списування у ході захисту комп'ютерних практикумів (лабораторних робіт), на модульній контрольній роботі, на екзамені.

Комп'ютерні практикуми захищаються особисто з попередньою перевіркою теоретичних знань, які необхідні для виконання завдань за темою комп'ютерного практикуму. Перевірка практичних результатів включає перевірку коду та виконання програмного застосунку.

Модульна контрольна робота проводиться письмово без застосування допоміжних засобів (мобільні телефони, планшети, література та ін.) за принципом хронометражу часу виконання: доступ до контрольного завдання МКР відкривається викладачем у заздалегідь оголошений момент на визначений період часу. Результати модульної контрольної роботи оголошуються студентам на наступному занятті.

Порядок зарахування пропущених занять. Відпрацювання пропущеного заняття з лекційного курсу здійснюється шляхом підготовки і захисту реферату за відповідною темою. Захист реферату відбувається відповідно до графіку консультацій викладача, з яким можна ознайомитись на кафедрі. Відпрацювання пропущеної комп'ютерного практикуму здійснюється шляхом самостійного виконання завдання і його захисту відповідно до графіку консультацій викладача.

Політика щодо академічної доброчесності. Політика та принципи академічної доброчесності визначені у розділі 3 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.

Норми етичної поведінки. Норми етичної поведінки студентів і працівників визначені у розділі 2 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.

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

Рейтинг студента за освітній компонент складається з балів, що студент отримує за:

– виконання та захист **комп'ютерних практикумів** із 12 тем:

1. Відлагодження в інтегрованому середовищі розробки програм Microsoft Visual Studio (Тема 1. Робота з типами даних. Поняття абстрактних типів даних (контейнерів));

2. Перелічуваний тип даних (Тема 1. Робота з типами даних. Поняття абстрактних типів даних (контейнерів));
3. Одновимірні масиви (Тема 2. Статичні контейнери та алгоритми їх обробки);
4. Двовимірні масиви (Тема 2. Статичні контейнери та алгоритми їх обробки);
5. Рядки (Тема 2. Статичні контейнери та алгоритми їх обробки);
6. Алгоритми сортування (Тема 2. Статичні контейнери та алгоритми їх обробки);
7. Бінарні файли даних (Тема 3. Файлові структури даних);
8. Структури (Тема 4. Структури як тип користувача);
9. Однозв'язні списки (Тема 5. Динамічні контейнери та алгоритми їх обробки);
10. Двозв'язні списки (Тема 5. Динамічні контейнери та алгоритми їх обробки);
11. Списки (стек, черга) (Тема 5. Динамічні контейнери та алгоритми їх обробки);
12. Деревя (Тема 5. Динамічні контейнери та алгоритми їх обробки)

– одну модульну контрольну роботу.

Максимальна сума балів за всі контрольні заходи протягом семестру складає:

12 комп'ютерних практикумів 7 балів + 4 питання МКР · 4 бали = 100 балів.

Поточний контроль

Комп'ютерні практикуми (лабораторні роботи)

Ваговий бал – 7.

Максимальна кількість балів за всі роботи дорівнює 7 балів x 12 = 84 балів.

Критерії оцінювання:

- 7 балів** – вірно виконані реалізація алгоритмів, розрахункова частина робота та досконало проведений аналіз отриманих результатів;
- 5.6...6.9 балів** – вірно виконані реалізація алгоритмів та розрахункова робота, але з недостатньо проведеним аналізом отриманих результатів та теоретичним обґрунтуванням;
- 4.2...5.5 балів** – робота виконана з грубими помилками;
- 0 балів** – невірно виконано.

Модульна контрольна робота

Ваговий бал – 4.

Максимальна кількість балів за контрольну роботу дорівнює 4 бали x 4 питання x 1 МКР = 16 балів

Критерії оцінювання:

- 2.4 ... 4 бали** – повна вірна відповідь на питання (відповідь містить не менше ніж 60 % необхідної інформації);
- 0 балів** – відсутність відповіді або невірна відповідь на питання (менше 60% необхідної інформації).

Календарний контроль

Проміжна атестація студентів (далі – атестація) є календарним контролем. Метою проведення атестації є підвищення якості навчання студентів та моніторинг виконання графіка освітнього процесу студентами.

Умовою першої атестації є отримання не менше 27 балів та захист всіх комп'ютерних практикумів.

Умовою другої атестації є отримання не менше 45 балів та захист всіх комп'ютерних практикумів.

Семестровий контроль

Умовами допуску до заліку є:

- зарахування всіх комп'ютерних практикумів;
- написана МКР;

Студенти, які виконали всі умови допуску до заліку та мають рейтингову оцінку 60 і більше балів, отримують відповідну до набраного рейтингу оцінку без додаткових випробувань.

Студенти, які виконали всі умови допуску до заліку та мають рейтингову оцінку менше 60 балів, виконують залікову контрольну роботу.

Студенти, які у семестрі отримали більше 60 балів, можуть взяти участь у заліковій контрольній роботі. У цьому разі бали, отримані ними на заліковій контрольній роботі, є остаточними.

Залікова контрольна робота оцінюється зі 100 балів. Кількість запитань у кожному білеті – 3.

Ваговий бал першого запитання – 25, другого – 25, третього – 50.

Критерії оцінювання першого та другого запитань на екзамені:

- 23...25 балів** – змістовна відповідь на теоретичне питання білету;
- 17...22 балів** – добра відповідь на питання, але з невеликими зауваженнями;
- 15...16 балів** – задовільна відповідь на питання (є декілька грубих помилок у відповіді);
- 0 бали** – більше двох грубих помилок/незнання питання

Критерії оцінювання третього запитання на екзамені:

- 45...50 балів** – правильна програмно-алгоритмічна реалізація розв'язку задачі білету;
- 35...44 балів** – допущено несуттєві неточності в програмній реалізації;

- 30...34 балів** – помилки в рахунках, не викладено основні етапи алгоритмічної частини;
0 балів – незнання розв'язку задачі, грубі помилки.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою

Сума балів, отриманих студентом, переводиться до оцінки згідно з таблицею:

Бали	Оцінка
100...95	Відмінно
94...85	Дуже добре
84...75	Добре
74...65	Задовільно
64...60	Достатньо
Менше 60	Незадовільно
Є не зараховані комп'ютерні практикуми або не написана МКР	Не допущено

8. Додаткова інформація з дисципліни (освітнього компонента)

Дистанційне навчання. В умовах дистанційного режиму організація освітнього процесу здійснюється з використанням технологій дистанційного навчання: платформи дистанційного навчання «Сікорський» та «Електронний кампус». Навчальний процес у дистанційному режимі здійснюється відповідно до затвердженого розкладу навчальних занять. Заняття проходять з використанням сучасних ресурсів проведення онлайн-зустрічей (організація відео-конференцій).

Робочу програму навчальної дисципліни (силабус):

Складено доцентом кафедри БМК, к.т.н., Алхімовою Світланою Миколаївною

Ухвалено кафедрою біомедичної кібернетики (протокол №1 від 31 серпня 2023 року)

Погоджено Методичною комісією факультету (протокол №1 від 1 вересня 2023 року)