**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ОПТИКО-МЕХАНІЧНИЙ КОЛЕДЖ**

Циклова комісія програмування та інформаційних технологій

|  |  |
| --- | --- |
|  | **«ЗАТВЕРДЖУЮ»** В.о. директора   \_\_\_\_\_\_\_\_\_\_\_\_Т.Г.Колесник  «\_\_\_\_» \_\_\_\_\_\_\_ 2020 року |

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

***ОСНОВИ АЛГОРИТМІЗАЦІЇ ТА ПРОГРАМУВАННЯ*для студентів**

|  |  |
| --- | --- |
| *галузь знань* | ***07 Управління та адміністрування*** |
| *спеціальність* | ***076 Підприємництво, торгівля та біржова діяльність*** |
| *освітній рівень* | ***Молодший спеціаліст*** |
| *освітня програма* | ***Інформаційна діяльність підприємства*** |
| *спеціалізація* |  |
| *вид дисципліни* | ***Обов’язкова*** |

|  |  |  |
| --- | --- | --- |
|  | Форма навчання | денна |
|  | Навчальний рік | 2019/2020 |
|  | Семестр | 4 |
|  | Кількість кредитів ECTS | 5 |
|  | Мова викладання, навчання та оцінювання | українська |
|  | Форма заключного контролю | екзамен |

Викладач: Лумпова Тетяна Іванівна

Пролонговано: на 20\_\_\_/20\_\_\_ н.р. \_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_) «\_\_\_» \_\_\_\_\_\_ 20\_\_\_\_ р.

на 20\_\_\_/20\_\_\_ н.р. \_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_) «\_\_\_» \_\_\_\_\_\_ 20\_\_\_\_ р.

**Київ – 2020**

Розробник: Лумпова Тетяна Іванівна, к.е.н.

|  |
| --- |
| ЗАТВЕРДЖЕНО |
| Голова циклової комісії  програмування та інформаційних технологій |
| ( Заболотний В.О. ) |
| (підпис) (прізвище та ініціали) |
| Протокол № \_\_\_\_\_ від «\_\_\_»\_\_\_\_\_\_\_20\_\_\_\_ р. |

Схвалено педагогічною радою

Оптико-механічного коледжу КНУ імені Тараса Шевченка

Протокол від «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_\_ року № \_\_\_\_\_

Голова педагогічної ради \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)  
 (підпис) (прізвище та ініціали)

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_\_ року

**1. Мета дисципліни** – забезпечення студентам рівня знань та практичних навиків в області програмування у відповідності до кваліфікаційних вимог.

**2. Попередні вимоги до опанування або вибору навчальної дисципліни:**1. Знати: проблеми і напрями розвитку програмних засобів; етапи розробки програмного продукту; основні алгоритмічні структури, що застосовуються при розробці програм; синтаксис та семантику однієї з універсальних мов програмування високого рівня; структури та типи даних; основні прийоми структурного програмування.   
2. Вміти: аналізувати умову поставленої задачі; застосовувати для розв’язку задачі необхідну методологію розробки програм; описувати алгоритм розв’язку задачі, дотримуючись основних вимог алгоритмічної структури; використовувати конструкції мови програмування та типи даних для рішення задачі; застосовувати прийоми структурного програмування; користуватись стандартними бібліотеками підпрограм; використовувати графічні можливості мови програмування для побудови зображень та графіків математичних функцій.

3. Володіти елементарними навичками розробки та запису алгоритмів розв’язку задач; розробки програмного коду на одній з універсальних мов програмування; тестування програмного продукту та усунення синтаксичних та логічних помилок.

**3. Анотація навчальної дисципліни:** обсяг дисципліни 150 год. – 5,0 кредитів ECTS . Лекцій – 22 год., практичних робіт – 20 год., лабораторних робіт -30 год., самостійна робота – 78 год.  
Форма контролю – диференційований залік.

Предметом навчальної дисципліни є основні поняття та методи алгоритмізації та програмування, навики написання та налагодження програм мовою C++, створення структур даних, оволодіння методологією проектування програмних засобів.

Основні теми дисципліни: основи алгоритмізації; основи програмування на мові C++; структурне, процедурне програмування; рекурсія; організація даних та алгоритми їх обробки; файлові та динамічні структури даних.

Вивчення дисципліни базується на дисциплінах «Математика», «Інформатика», «Фізика», «Вступ до спеціальності» і являється базовою для вивчення дисциплін «Об'єктно-орієнтоване програмування*»,* «Інструментальні засоби візуального програмування», «Бази даних», «Конструювання програмного забезпечення», «Основи програмної інженерії».

**4. Завдання (навчальні цілі):** досягнення студентами таких компетентностей, як здатність до абстрактного мислення (ЗК1), здатність опановувати знання і розуміти предметну сферу та професійну діяльність (ЗК5), здатність використовувати сучасні інформаційні технології для розв’язання різноманітних задач у навчальній та практичній діяльності (ЗК7), здатність працювати як в команді так і самостійно (ЗК9).

**5. Результати навчання за дисципліною:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Результат навчання**  **(1.знати; 2.вміти; 3.комунікація; 4.автономність та відповідальність)** | | **Форми (та/або методи і технології) викладання і навчання** | **Методи оцінювання та пороговий критерій оцінювання (за необхідності)** | **Відсоток у підсумковій оцінці з дисципліни** |
| Код | Результат навчання |
| *1.1* | *Знати етапи створення програмного продукту* | *Лекція, самостійна робота* | *Опитування* | *10%* |
| *1.2* | *Знати основні базові структури алгоритмів* | *Лекція, самостійна робота* | *Опитування* | *10%* |
| *1.3* | *Знати основні елементи мови програмування С++ та типи даних* | *Лекція, самостійна робота* | *Контрольна лабораторна робота* | *10%* |
| *1.4* | *Знати основні методи структурного програмування* | *Лекція, самостійна робота* | *Опитування* | *10%* |
| *2.1* | *Вміти аналізувати умову поставленої задачі та застосовувати необхідну методологію розробки програм* | *Лабораторна робота, самостійна робота* | *Контрольна лабораторна робота* | *15%* |
| *2.2* | *Вміти описувати алгоритм розв’язку задачі, дотримуючись основних вимог алгоритмічної структури* | *Лабораторна робота, самостійна робота* | *Контрольна лабораторна робота* | *15%* |
| *2.3* | *Вміти використовувати конструкції мови програмування та типи даних для рішення задачі* | *Лабораторна робота, самостійна робота* | *Контрольна лабораторна робота* | *15%* |
| *2.4* | *Вміти застосовувати прийоми структурного програмування та користуватись стандартними бібліотеками підпрограм* | *Лабораторна робота, самостійна робота* | *Контрольна лабораторна робота* | *15%* |

**6. Співвідношення результатів навчання дисципліни із програмними результатами навчання**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Результат навчання дисципліни  (код)  Програмні результати навчання | **1.1** | **1.2** | **1.3** | **1.4** | **2.1** | **2.2** | **2.3** | **2.4** |
| Оцінювати і вибирати методи і моделі розробки, впровадження, експлуатації програмних засобів та управління ними на всіх етапах життєвого циклу -ПРН1 | + | + | + |  |  |  |  |  |
| Аналізувати, оцінювати і вибирати методи, сучасні програмно-апаратні інструментальні та обчислювальні засоби, технології, алгоритмічні та програмні рішення для ефективного виконання конкретних виробничих задач з програмної інженерії - ПРН3 |  |  |  | + |  |  |  |  |
| Володіти методами та технологіями організації та застосування даних - ПРН7 |  |  |  |  | + | + | + | + |

**7. Схема формування оцінки**

**7.1 Форма оцінювання студентів:   
- семестрове оцінювання:**1.Тематичні опитування та тестові завдання на практичних роботах (10 робіт): РН1.1 –РН1.4 -3 бали /1 бал за кожне виконане завдання

2.Контрольні лабораторні роботи (15 робіт): РН2.1 – РН2.4 – 3 бали / 2 бали за кожну виконану лабораторну роботу  
**3.** Тематичне комплексне опитування (5 опитувань) – РН1.1, РН1.2, РН2.1 РН2.2, РН2.3, РН2.4 – 5 балів / 4 бали.

**- підсумкове оцінювання:**  
Диференційований залік – за результатами роботи студента впродовж семестру  
 100 балів/60 балів

**7.2 Організація оцінювання**

Оцінювання за формами контролю у четвертому семестрі:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Практичні роботи*** | ***Лабораторні роботи*** | ***Тематичне комплексне опитування*** | ***Загальна кількість балів*** |
| ***Мінімум*** | 10 х 1 б = 10 | 15 х 2 б = 30 | 5 х 4 б = 20 | 60 |
| ***Максимум*** | 10 х 3 б = 30 | 15 х 3 б = 45 | 5 х 5 б = 25 | 100 |

Усні відповіді можуть надавати додаткові бали до загальної кількості балів.

Результати навчальної діяльності студентів оцінюються за 100 бальною шкалою.

Рекомендований мінімум для отримання заліку – 60 ***балів***.

**7.3 Шкала відповідності оцінок**

|  |  |
| --- | --- |
| **Відмінно**/ Excellent | 90-100 |
| **Добре**/ Good | 75-89 |
| **Задовільно**/Satisfactory | 60-74 |
| **Незадовільно**/ Fail | 0-59 |
| **Зараховано**/ Passed | 60-100 |
| **Не зараховано**/ Fail | 0-59 |

**8. Структура навчальної дисципліни**

**ТЕМАТИЧНИЙ ПЛАН ЛЕКЦІЙ І ПРАКТИЧНИХ РОБІТ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Назва лекції** | **Кількість годин** | | |
| **Лекції** | **Прак-тичні** | **Само-стійна робота** |
| **Тема 1. Основи алгоритмізації обчислювальних процесів** | | | | |
| 1 | Алгоритми і алгоритмізація. Принципи побудови алгоритму вирішення задач. Інформаційні системи й технології | 2 |  | 4 |
| 2 | Архітектура сучасних ПК. Подання цілих і дробових чисел в двійковій системі числення. Алгоритми і структури даних Побудова і аналіз алгоритмів. Методи розробки алгоритмів. *Тематичне опитування.* | 2 |  | 4 |
| **Тема 2. Основні поняття та типи даних мови програмування С++** | | | | |
| 3 | Загальні поняття. Елементи мови С++ - константи, змінні, операції, перетворення типів. | 2 | 2 | 6 |
| 4 | Поняття вхідного та вихідного потоку, найпростіші математичні функції. | 2 | 2 | 6 |
| 5 | Структура програми. Операції порівняння, логічні операції. *Тематичне опитування.* | 2 | 2 | 4 |
| **Тема 3. Програми лінійної, розгалуженої та циклічної структури.** | | | | |
| 6 | Програми лінійної, розгалуженої та циклічної структури. Загальний огляд. | 2 | 2 | 4 |
| 7 | Блок. Область дії оголошення імені. Практичні поради щодо написання програм | 2 | 2 | 4 |
| 8 | Принципи структурного програмування. *Тематичне опитування.* | 2 |  | 4 |
| **Тема 4. Символьні та рядкові величини. Складні типи даних – масиви, структури.** | | | | |
| 9 | Символьні та рядкові величини. | 2 | 2 | 4 |
| 10 | Складні типи даних – масиви, структури. *Тематичне опитування.* | 2 | 2 | 4 |
| **Тема 5. Процедури і функції. Файли.** | | | | |
| 11 | Процедури і функції. | 2 | 2 | 4 |
| 12 | Файли. *Тематичне опитування.* | 2 | 2 | 4 |
| **Тема 6. Сучасний підхід до програмування. Тестування програм.** | | | | |
| 13 | Сучасний підхід до програмування. Тестування програм. *Підсумкове опитування.* | 2 |  | 4 |
|  | **Всього по предмету:** | **26** | **22** | **56** |

Загальний обсяг 150*год.,* в тому числі:

Лекції **– 22** *год.*Лабораторні роботи **– 30** *год.*Самостійна робота **- 78** *год.*

**9. Рекомендовані джерела інформації.**.

**Основні**

1. Ковалюк Т. В. Алгоритмізація та програмування: Підручник. — Львів: «Магнолія 2006», 2013. — 400 с., ил.
2. Єжова Л. Ф. Алгоритмізація і програмування процедур обробки інформації: Навч.-метод. посіб. для самост. вивч. дисц. К.: КНЕУ, 2000.
3. Вступ до програмування мовою С++. Організація обчислень: навч. посіб. / Ю. А. Бєлов, Т. О. Карнаух, Ю. В. Коваль, А. Б. Ставовський. – К. : Видавничо-поліграфічний центр "Київський університет", 2012. – 175 с. с.: іл. ISBN (укр.) . URL: <http://csc.knu.ua/uk/library/books/belov-24.pdf>
4. Джейс Либерти Освой самостоятельно С++ за 21 день: 3-е изд. пер. с англ.: Уч. пос. – М.: Издательский дом „Вильямс”, 2001. – 816 с.: ил..
5. Павловская Т.А. С/С++. Программирование на языке высокого уровня. СПб.: Питер, 2003. – 461 с. URL: <http://www.ph4s.ru/bookprogramir_1.html>
6. Страуструп Б. Язык программирования С++. — СПб.; М.: Невский диалект — ЗАО “Изд-во “Бином”, 1999.
7. Бондарев В. М. Программирование на С++: Учеб. пособие. — Харьков: СМИТ, 2004г. — 294 с. URL: <https://www.rulit.me/author/bondarev-v-m/programmirovanie-na-c-get-161082.html>
8. Глушаков С.В., Коваль А. В., Смирнов С.В. Язык программирования С++. Учебный курс. Харьков: Фолио. М.: «Издательство АСТ», 2001. – 500 с. URL: <http://www.ph4s.ru/bookprogramir_1.html>
9. Динман М.И. С++. Освой на примерах. – СПб.: БХВ – Петербург, 2006. – 384 с. URL: <http://www.ph4s.ru/bookprogramir_1.html>
10. Зелковиц М., Шоу А., Геннон Дж. Принципы разработки программного обеспечения. — М.: Мир, 1982. — 368 с. URL: <http://computersbooks.net/index.php?id1=4&category=teoriyaprogramirovaniya&author=zelkovic-m&book=1982>
11. Липпман С. Б., Лажойе Ж. Язык программирования С++: Вводный курс. — М.: ДМК, 2001. URL: <http://www.insycom.ru/html/metodmat/inf/Lipman.pdf>
12. Дейтел Х., Дейтел П. Основы программирования на С++. – М.: Бином, 1999. – 1024 с. URL: [http://ijevanlib.ysu.am/wp-content Холингвэрт Д., Баттерфилд Д., Сворт Б., Оллсоп Д. C++ Builder 5. Руководство разработчика. Том 1. ­ М.: И.Д. "Вильямс", 2001. ­ 880 с. URL: <http://mexalib.com/view/58962>/uploads/2018/03/deytel.pdf](http://ijevanlib.ysu.am/wp-content/uploads/2018/03/deytel.pdf)
13. Холингвэрт Д., Баттерфилд Д., Сворт Б., Оллсоп Д. C++ Builder 5. Руководство разработчика. Том 2. Сложные вопросы программирования. ­ М.: И.Д. "Вильямс", 2001. ­ 817 с. URL: <http://mexalib.com/view/58965>
14. Бадд Т. Объектно-ориентированное программирование в действии. [2-е изд.] – СПб.: Изд-во "Питер". 1997. URL: http://khizha.dp.ua/library/Timothy\_Budd\_-\_Introduction\_to\_OOP\_(ru).pdf.

**Додаткові**

1. Н.Вирт. Алгоритмы и структуры данных. Москва, Мир, 1989 г. 420 с.
2. Куприянова Л.М. Программирование, алгоритмические языки и вычислительная математика: Учеб. пособие. — М.: Финансы и статистика, 1985. —223 с.
3. Шаховська, Н. Б. **Алгоритм**и і структури даних [Текст]: посібник / Н.Б. Шаховська, Р.О. Голощук; За ред. В.В. Пасічника. - Львів: Магнолія, 2010. - 215 с.
4. А. Я. Архангельский Программирование в С++ Builder 6. – М.: ЗАО «Издательство БИНОМ», 2004 г. – 1152 с.: ил.

**10. Інформаційні ресурси**

1. Основи програмування на мовах Сі та Cі++ для початківців. URL: <http://cppstudio.com/uk/>
2. Керівництво по продуктам Microsoft. URL: <https://msdn.microsoft.com/ru-ru/>
3. Сайт з довідкою по С++ . URL: <http://www.cplusplus.com/>
4. Алгоритми та методи програмування. URL: <http://algolist.ru/>

**11.Методичне забезпечення**

1. Витяг з навчального плану.
2. Навчальна (типова) програма.
3. Робоча навчальна програма.
4. Плани занять.
5. Конспект лекцій з дисципліни.
6. Завдання для обов‘язкової контрольної роботи.
7. Інструкційно-методичні матеріали до лабораторних занять.
8. Інструкційно-методичні матеріали до самостійної роботи.
9. Питання до заліків.
10. Контрольні тестові завдання до заліків.
11. Питання до екзамену.
12. Екзаменаційні білети.
13. Навчальний посібник.
14. Роздавальний матеріал.
15. Презентації до тем.

**Індивідуальні завдання**

| №пп | Тема | Вид завдання |
| --- | --- | --- |
|  | Парадігми і мови програмування. Історія створення С++. | Реферат |
|  | Програмне середовище розробника С++ Builder. Інтегроване середовище розробки програм. Елементи інтерфейсу, налагодження параметрів. | Реферат |
|  | Програмне середовище розробника С++ Builder. Налагодження програм, режими покрокового виконання, перегляд локальних змінних. | Реферат |
|  | Функції в мові C++ (з прикладами використання) | Реферат |
|  | Історія розвитку ЕОМ та програмування як науки. Розвиток програмування в Україні. | Реферат |
|  | Методи тестування програмного продукту. Помилки, які виявляються в процесі налагодження програми. | Реферат |
|  | Методології розробки програм: низхідне, модульне та структурне проектування. | Реферат |
|  | Правила оформлення коду в С++ | Реферат |
|  | Поняття програми, мови програмування. Типи мов програмування. | Реферат |
|  | Еволюція мов програмування та напрями розвитку мов програмування. | Реферат |

**ПИТАННЯ НА ЗАЛІК**

1. Історія розвитку ЕОМ та програмування як науки. Розвиток програмування в Україні.
2. Етапи розв’язування задач на ЕОМ.
3. Алгоритм, властивості алгоритму.
4. Процес алгоритмізації та його етапи.
5. Правила побудови алгоритму.
6. Поняття декомпозиції та інтеграції.
7. Типи та базові структури алгоритмів.
8. Базова структура алгоритму «слідування».
9. Базові структури алгоритмів «розгалуження» та «обхід».
10. Базова структура алгоритму «багатоваріантний вибір».
11. Базова структура алгоритму «цикл до».
12. Базова структура алгоритму «цикл доки».
13. Методології розробки програм.
14. Загальні принципи роботи з файлами в мові С++; операції читання та запису інформації для файлів різних типів.
15. Сутність і види інформаційних систем
16. Історія створення мови С++.
17. Складові частини інтегрованого середовища програмування; поняття вхідного, об'єктного та виконуваного кодів програми..
18. Визначення інформації та даних, зв’язок між ними.
19. Структура програми, написаної на мові С++.
20. Основні елементи мови С++.
21. Типи операторів мови С++.
22. Типи даних, що використовуються в мові С++ (у вигляді схеми).
23. Які основні та складені типи даних має мова С++
24. Що таке «область дії ідентифікатора» та «клас пам’яті»?
25. Тривалість життя об’єктів. Коли і як застосовується автоматичне, статичне і динамічне виділення пам’яті?
26. Що таке пріоритет виконання операцій? Перерахуйте відомі Вам операції мови С++ в порядку зростання старшинства?
27. Способи представлення дійсних чисел.
28. Арифметичні вирази, пріоритет виконання арифметичних операцій.
29. Логічні вирази, пріоритет виконання логічних операцій.
30. Поняття лінійної програми.
31. Процедури введення даних з клавіатури та виведення даних на екран.
32. Поняття розгалуженої програми. Дати характеристику умовному оператору IF.
33. Поняття розгалуженої програми. Дати характеристику оператору вибору CASE.
34. Поняття циклічної програми. Дати характеристику оператору циклу FOR.
35. Оператор циклу WHILE.
36. Охарактеризуйте типи циклічних операторів? В яких випадках застосовується той чи інший тип оператора?
37. Методології розробки програм.
38. Визначення алгоритмізації, її процеси та етапи.
39. Правила побудови алгоритмів
40. Види алгоритму
41. Способи записів алгоритмів
42. Що таке транслятор, компілятор, інтерпретатор?
43. Що таке директиви препроцесора? Яке їх призначення?
44. Що таке системи програмування?
45. Символьні величини, **виконання дій з елементами символьного масиву.**
46. Рядкові величини, стандартні функції для роботи з ними.
47. Поняття масиву; типи масивів; операції над масивами.
48. Поняття структури; типи структур; операції над елементами структур.
49. Як у С++ реалізується введення-виведення елементів масиву та структури?
50. Мови програмування
51. Поняття підпрограми; типи підпрограм; використання локальних та глобальних змінних, формальних та фактичних параметрів.
52. Класифікація стандартних процедур та функцій мови С++.
53. Основні задачі інформаційної системи
54. Основні функції інформаційних систем та їх зміст.
55. Задачі оброблення інформації.
56. Класифікація структур даних. Базові одиниці структур даних.
57. Лінійні та нелінійні структури даних. Приклади.
58. Принципи побудови і аналізу алгоритмів.
59. Тестування програм: визначення та призначення.
60. Види тестування.
61. Поняття файлу, операції з файлами.
62. Організація роботи з файлами.

**Програма навчальної дисципліни**

**Тема 1.** Основи алгоритмізації обчислювальних процесів (8 годин)

**Лекція 1.** Алгоритми і алгоритмізація.

Змістовне поняття алгоритму. Основні властивості алгоритмів. Приклади. Виконавець алгоритмів та його система команд. Правила побудови алгоритмів. Класифікація алгоритмів. **Способи записів алгоритмів.** Програми, компілятори, інтерпретатори. Практичні аспекти застосування алгоритмів. Поняття алгоритмізації. Процеси і етапи алгоритмізації. Декомпозиція та інтеграція. Приклади алгоритмізації на практиці. Об'єкт алгоритмізації.

*Для самостійного вивчення (3 години)*: Архітектура сучасних ПК. Розвиток мов програмування. Подання цілих і дробових чисел в двійковій системі числення.

*Література* [1; 2; 16]

*Контрольні запитання*.

1. Дайте визначення алгоритму та перелічите його властивості.
2. Як можна визначити виконавця алгоритму та систему його команд?
3. Перелічите правила побудови алгоритмів.
4. Яки види алгоритмів ви можете визначити?
5. В який спосіб можна записати алгоритм?
6. Які позначки використовуються при запису блок-схем?
7. Яким чином програма перетворюється в машинні коди?
8. Дайте визначення алгоритмізації.
9. Які процеси та етапи алгоритмізації?
10. В чому полягає декомпозиція?
11. Коли застосовується інтеграція?
12. Наведіть приклади алгоритму та алгоритмізації в повсякденному житті.

*Запитання для самоконтролю по самостійно вивченому матеріалу*:

1. Що таке система числення? Які типи систем числення ви знаєте?
2. Яка система числення використовується для подання чисел у пам’яті комп’ютера? Чому?
3. Охарактеризуйте архітектуру сучасних ПК.
4. Назвіть основні етапи розвитку мов програмування.

**Лекція 2.** Принципи побудови алгоритму вирішення задач. Інформаційні системи й технології

Основні етапи розв’язування задач за допомогою комп’ютера. Сутність і види інформаційних систем (ІС)**.** Властивості системи. Місія та основні задачі ІС. Компоненти та основні функції ІС. Етапи розвитку інформаційних систем. Поняття системного підходу. Етапи розвитку й класифікація інформаційних технологій. Класифікація автоматизованих інформаційних систем. Класифікація задач обробки інформації..

*Для самостійного вивчення (3 години)*: Класифікація програмного забезпечення. Системні програми. Операційні системи та оболонки.

*Література* [1; 2; 16]

*Контрольні запитання*.

1. Назвіть основні етапи розв’язування задач за допомогою комп’ютера.
2. Що таке «припустима елементарна операція»?
3. Як Ви можете визначити інформаційну систему, чим вона характеризується?
4. Які основні характеристики інформаційної системи?
5. В чому полягає місія та основні задачі інформаційної системи.?
6. Які компоненти ІС і як вони приймають участь в обробці інформації?
7. Які основні функції ІС?
8. Назвіть етапи розвитку інформаційних систем.
9. В чому полягає системний підхід?
10. Назвіть етапи розвитку інформаційних технологій?
11. Які типи ІС Ви знаєте?

*Запитання для самоконтролю по самостійно вивченому матеріалу*:

1. Що таке програмне забезпечення?
2. За якими ознаками можна класифікувати програмне забезпечення?
3. Що забезпечують прикладні програми?
4. Які функції виконують системні програми?
5. Для чого використовуються сервісні програми?
6. Що таке операційна система?
7. Які сучасні операційні системи Ви знаєте?
8. Що входить до функцій операційної системи?

**Лекція 3.** Алгоритми і структури даних

Інформація і дані. **Структура даних.** Рівні описування/подання структур даних. Логічне структурування економічної інформації. Одиниці фізичної структури даних. Лінійні структури даних. Нелінійні структури даних. Найпростіші стандартні та складені типи даних.

*Для самостійного вивчення (2 години)*: Вивчення лекційного матеріалу та додаткових джерел. Розгляд запитань і виконання завдань для самостійної роботи, запропонованих на лекції.

*Література* [1; 2; 15, 16]

*Контрольні запитання*.

1. Надайте визначення інформації та даних.
2. Як пов’язані інформація і дані?
3. Поняття структури даних. Як вона характеризується?
4. Класифікація структур даних у програмах користувача.
5. Визначте рівні подання структур даних.
6. Назвіть елементи логічної структури економічної інформації.
7. Надайте визначення інформаційного масиву даних та інформаційного потоку.
8. Окресліть взаємозв'язок між елементами логічної структури економічної інформації.
9. Визначте одиниці фізичної структури даних.
10. Наведіть приклади лінійних та нелінійних структур даних.
11. Чим відрізняється стек від черги?
12. Назвіть складені типи даних.

**Практична робота №1.** Алгоритми і алгоритмізація. Принципи побудови алгоритму вирішення задач.

Контрольне завдання. Накреслити блок-схему системи ведення звітності головного офісу мережі супермаркетів, яка щоденно збирає звіти про продаж товарів в своїх магазинах, виконує їх перевірку та накопичення, а також формує зведений звіт про продажі.

Розгляд укрупненої блок-схеми, аналіз помилок контрольного завдання та проведення декомпозиції «блоку» (підсистеми) введення даних.

*Контрольні запитання*

1. Поняття інформаційної моделі.
2. Назвіть етапи вирішення прикладних задач з використання ПК.
3. Що ми розуміємо під поняттям “алгоритм”?
4. Які є способи опису алгоритмів?
5. Поняття схеми алгоритму та правила її побудови.
6. Які властивості повинен мати алгоритм?
7. Які є три головні алгоритмічні конструкції?
8. З чого складаються прості (лінійні) алгоритми?
9. Який процес називається розгалуженим?
10. Який процес називається циклічним?
11. Види циклічних обчислювальних процесів, їх відмінність.

**Лекція №4. Побудова і аналіз алгоритмів. Методи розробки алгоритмів.**

Формалізація алгоритмів. Покрокове проектування алгоритмів. Методи розробки алгоритмів. Низхідне проектування. Висхідний метод проектування. Аналіз алгоритму. Поняття складності та трудомісткості алгоритму. Процес створення алгоритму. Структурні принципи алгоритмізації. Цілі структурного програмування. Тестування алгоритму. Економічна інформація та її структура. Вимоги до економічної інформації. Формалізація, алгоритмізація та автоматизована обробка економічної інформації.

*Для самостійного вивчення (2 години)*: Вивчення лекційного матеріалу та додаткових джерел. Розгляд запитань і виконання завдань для самостійної роботи, запропонованих на лекції.

*Література* [1; 2; 10]

*Контрольні запитання*.

1. Дати визначення етапів формалізації алгоритмів.
2. Назвіть та охарактеризуйте найпоширеніші методи розробки алгоритмів.
3. В чому полягає аналіз алгоритмів.
4. Цілі структурного програмування.
5. В чому полягає тестування алгоритму.
6. Що розуміють під економічною інформацією і які особливості її оброблення?
7. Чи потрібен змістовний опис для задач оброблення економічної інформації?

**Практична робота №2. Основи алгоритмізації обчислювальних процесів. Тематичний контроль. Висновки та узагальнення.**

**Тема 2. Основні поняття та типи даних мови програмування С++**

**Лекція № 5. Загальні поняття. Елементи мови С++ - константи, змінні, операції, перетворення типів.**

# Короткі відомості про виникнення мови С++. Алфавіт, лексеми, синтаксис мови С++. Рекомендації щодо вибору ідентифікатора. Спеціальні символи. Структура програми. Рекомендації щодо використання коментарів. Основні вимоги при створенні програм. Типи даних. Змінні. Константи. Послідовності. Операції. Перетворення типів.

*Для самостійного вивчення (2 години)*: Історія створення та розвитку мови С++. Сумісність та перетворення типів [3, с.53-56].

*Література* [3-9; 11-12]

*Контрольні запитання*.

1. Що включає в себе алфавіт мови С++?
2. Що таке лексеми, як вони складаються, що до них відносять?
3. Дайте визначення ідентифікатору, яким чином він утворюється?
4. Що називають ключовими словами?
5. Які символи можуть бути роздільниками лексем?
6. Які коментарі використовує мова C++? Надайте поради до їх створення, наведіть приклади.
7. Що являє собою структура програми на C++? Наведіть основні вимоги, які слід ураховувати при створенні програм мовою C++.
8. Які основні та складені типи даних має мова С++?
9. Що таке змінна і як здійснюється її опис та визначення?
10. Що таке «область дії ідентифікатора» та «клас пам’яті»?
11. Які константи налічує С++? Охарактеризуйте їх застосування.
12. Що таке пріоритет операції? Наведіть приклади арифметичних та логічних операцій.
13. Які операції присвоювання та операції відношення налічує С++?
14. Що реалізують логічні операції та операції обробки окремих бітів?
15. Які операції над покажчиками та додаткові операції має С++?

**Практична робота №3. Елементи мови С++**

**Завдання.** Намалювати блок – схему рішення задач, наданих викладачем, та на мові С++ записати оголошення змінних, які будуть використовуватися в програмах. Записати мовою С++ вирази надані в завданнях.

*Контрольні запитання*

1. Що таке ідентифікатор? Які правила запису ідентифікаторів?
2. Що таке змінна? Для чого використовуються змінні? Яка відмінність змінної від константи?
3. Які є форми запису дійсних чисел у мові С++?
4. Які прості типи даних мови С++ Ви знаєте? Які кваліфікатори можна до них застосовувати?
5. Чому в програмах доцільно використовувати коментарі? Які є типи коментарів в С++?
6. Опишіть коротко структуру програми мовою С++.
7. Як здійснюється виведення на екран в мові С++?
8. Що таке керуюча послідовність? Наведіть приклади цих послідовностей.
9. Які операції в мові С++ Ви знаєте?
10. Які є форми запису операцій інкременту та декременту? Яка між ними різниця?
11. Що таке пріоритет виконання операцій? Перерахуйте відомі Вам операції мови С++ в порядку зростання старшинства?
12. Що таке бібліотека стандартних функцій? Для чого вони використовуються?

*Для самостійного вивчення (2 години)*: Вивчення лекційного матеріалу та додаткових джерел. Розгляд запитань і виконання завдань для самостійної роботи, запропонованих на лекції.

**Лабораторна робота №1. Вступ до С++.**

**Мета**: навчитися працювати в середовищі програмування **С++ Builder** в **консольному режимі**

**Лекція № 6. Поняття вхідного та вихідного потоку, найпростіші математичні функції.**

Основи введення-виведення інформації. Введення та виведення даних в потоці. Форматоване введення / виведення. Операції, операнди, вирази. Старшинство операторів і порядок виконання операцій. Бібліотечні математичні функції та константи.

*Для самостійного вивчення (2 години)*: Різновиди констант; складені присвоювання [3, с.30-53] . Вивчення лекційного матеріалу та додаткових джерел. Розгляд запитань і виконання завдань для самостійної роботи, запропонованих на лекції.

*Література* [3-9; 11-12]

*Контрольні запитання*.

1. Що представляють собою операції введення-виведення інформації?
2. Як визначаються стандартні файли введення та виведення даних в потоці?
3. Як здійснюється потокове введення / виведення даних?
4. Яких правил потрібно дотримуватися при організації потокового введення даних?
5. Що таке буфер і який його об‘єм?
6. Які функції використовуються для форматованого введення / виведення даних?
7. Які операції називаються арифметичними?
8. Які дії виконує операція sizeof**?**
9. Поясніть властивість лівобічного зв'язування.
10. Який модуль необхідно підключити при виконанні математичних функцій?

**Практична робота №4. Використання математичних функцій, введення – виведення даних.**

**Завдання.**

1. Написати на мові С++ програму введення даних для розрахунку формул, наданих викладачем, а також виведення результатів. Використати потокове введення та виведення даних.
2. Після ознайомлення з теоретичною частиною написати оператори форматованого введення та виведення даних за завданням наданим викладачем.

*Контрольні запитання*

1. Для чого слугують заголовні файли?
2. Як компілятор шукає розташування заголовних файлів?
3. Які маніпулятори використовуються для завдання довжини поля, дробової частини числа?
4. Який заголовний файл використовується для маніпуляторів завдання довжини поля, дробової частини числа?

**Лабораторна робота №2.** **Використання математичних функцій, введення даних для розрахунку та виведення результатів на консоль.**

**Лекція № 7. Структура програми. Операції порівняння, логічні операції.**

Структура програми. Макровизначення. Директиви умовної компіляції. Зовнішні оголошення. Поняття функції. Операції порівняння. Логічні операції. Правила виконання логічних операцій.Логічні вирази. Тернарна операція.

*Для самостійного вивчення (2 години)*: Вивчення лекційного матеріалу та додаткових джерел. Розгляд запитань і виконання завдань для самостійної роботи, запропонованих на лекції.

*Література* [1-4]

*Контрольні запитання*.

1. Визначте основні частини типової структури програми на С++.
2. Що таке макровизначення? Наведіть приклади.
3. Для чого використовується умовна компіляція?
4. Що може міститися в зовнішніх оголошеннях?
5. Надайте визначення функції.
6. Які значення може містити умова, що перевіряється?
7. Як визначається результат операція **not?**
8. Як визначається результат операція **and?**
9. Як визначається результат операція **or?**
10. Поясніть, що представляє собою логічний вираз.
11. Як визначається тернарна операція?

**Практична робота №5. Тематичне опитування**

**Лабораторна робота №3.** **Використання математичних функцій, операції порівняння, логічні операції, форматоване виведення результатів на консоль.**

**Лекція № 8. Програми лінійної, розгалуженої та циклічної структури**

Виведення значень виразів. Блок. Область дії оголошення імені. Інструкції розгалуження (Оператор умовного переходу **if, оператор-перемикач** switch). Прості інструкції повторення обчислень (Інструкція циклу з передумовою **while**, Інструкція циклу з післяумовою **do while**, Інструкція циклу for). Переривання break та продовження циклу continue. Збільшення та зменшення Принципи структурного програмування.

*Для самостійного вивчення (2 години)*: Вивчення лекційного матеріалу та додаткових джерел. Розгляд запитань і виконання завдань для самостійної роботи, запропонованих на лекції.

*Література* [1-4]

*Контрольні запитання*

1. Які оператори реалізують розгалуження у програмі?
2. Як діє умовний оператор if?
3. Як працює оператор-перемикач switch?
4. Які оператори циклу використовуються у C++?
5. Як працює оператор циклу for?
6. Пояснити на прикладах використання циклу з передумовою і циклу з післяумовою.
7. Які оператори з розглянутих є операторами керування у мові C++?

*Контрольні запитання для надання письмових відповідей*.

1. Наведіть власні приклади умовного оператору **if** у двох формах.
2. Наведіть власний приклад оператор-перемикача **switch**.
3. Наведіть власні приклади всіх форм оператору циклу.
4. Наведіть власні приклади використання операторів переривання **break** та продовження циклу **continue.**

**Практична робота №6. Розгляд прикладів програм лінійної, розгалуженої та циклічної структури, побудова блок-схем**

**Лабораторна робота №4.** **Написання програм лінійної, розгалуженої та циклічної структури з використанням математичних функцій та виконанням операцій введення- виведення**

**Лабораторна робота №5. Написання програм лінійної, розгалуженої та циклічної структури з вкладеними циклами, використанням математичних функцій та виконанням операцій введення- виведення**

**Лабораторна робота №6. Написання програм з вкладеними циклами та розгалуженнями, використанням математичних функцій та виконанням операцій введення- виведення**

**Лекція № 9. Тематичне опитування у вигляді тестування. Символьні та рядкові величини. Складні типи даних – масиви, структури.**

Використання масивів. Дані символьного типу. **Виконання дій з елементами символьного масиву.** Використання рядків типу string. Функції для обробки рядків типу string. Приклади використання рядків типу string. Визначення та використання структур

*Для самостійного вивчення (10 годин)*: Вивчення лекційного матеріалу та додаткових джерел. Розгляд запитань і виконання завдань для самостійної роботи, запропонованих на лекції.

*Література [1-4, 11]*

*Контрольні запитання для самоперевірки*.

1. Що таке масив та які існують різновиди масивів?
2. Як здійснюється звернення до елементів масивів?
3. Як у С++ реалізується введення-виведення елементів масиву?
4. Що таке рядки та значення елементів символьного типу?
5. Що являє собою масив символьного типу?
6. Як здійснюється введення символьних даних?
7. Як виконується порівняння даних символьного типу?
8. Як визначити кількість символів у рядку?
9. Які функції мови С++ необхідні для виділення підрядка з рядка?
10. Як описуються дані типу структура?
11. Які типи полів може містити структура?

*Контрольні запитання для надання письмових відповідей*.

1. Наведіть приклад використання операції конкатенації.
2. Охарактеризуйте функції пошуку підрядка в рядку.
3. Наведіть приклади використання структур.

**Практична робота №7. Розгляд прикладів програм з символьними та рядковими величинами, масивами та структурами**

**Лабораторна робота №8.** **Написання програм з використанням символьних та рядкових величин**

**Лабораторна робота №9. Написання програм з використанням рядкових величин та структур**

**Лабораторна робота №10. Написання програм з використанням символьних та рядкових величин, масивів та структур. Тематичне опитування у вигляді тестування.**

**Лекція № 10.** **Процедури і функції. Файли.**

**Визначення функції.** Структура функції. Огляд процесу виконання виклику функції. Автоматична пам'ять або програмний стек. Механізм передачі параметрів. **Виклик функції з передачею значень.** Передача даних за замовчуванням. Поняття файлу. Використання файлів даних у програмі. Конструктори для роботи з файлами. Розгляд прикладу.

*Для самостійного вивчення (10 годин)*: Вивчення лекційного матеріалу та додаткових джерел. Розгляд запитань і виконання завдань для самостійної роботи, запропонованих на лекції.

*Література [1-4, 11]*

*Контрольні запитання для самоперевірки*.

1. Які змінні називаються локальними?
2. Розкажіть про порядок дій, які виконуються під час виклику функції.
3. Які параметри називаються формальними, а які фактичними?
4. Чи можна у викликаємій функції змінювати параметри при **виклику функції з передачею значень? Чи вплине така зміна значень параметрів на дії в головній функції, якщо ці параметри потім там використовуються?**
5. **Які правила встановленні для формування списку параметрів, коли** передача даних виконується за замовчуванням?

*Контрольні запитання для надання письмових відповідей*.

1. .Що потрібно зробити, щоб зберегти значення в середині функції між її викликами? Наведіть приклад у вигляді фрагменту коду.
2. Розгляньте останній приклад , виберіть всі мовні конструкції щодо оброблення файлів, які ми детально не розглядали в лекції.

**Практична робота № 8 Розроблення програм з процедурами і функціями**

**Лабораторна робота №11. Написання програм з використанням процедур і функцій**

**Практична робота №9. Розроблення програм з використанням файлів**

**Лабораторна робота №12.** **Написання програм з використанням файлів**

**Лабораторна робота №13. Написання програм з використанням функцій та файлів**

**Лабораторна робота №14. Написання програм з використанням функцій та файлів Тематичне опитування у вигляді тестування.**

**Лекція № 11.** **Сучасний підхід до програмування. Тестування програм.**

Покоління мов програмування. Парадигми програмування. Види діяльності зі створення програми**.** Поняття тестування. Основні методи тестування. Тестування і налагодження програми. Види тестування. Налагодження програми.

*Для самостійного вивчення (10 годин)*: Вивчення лекційного матеріалу та додаткових джерел. Розгляд запитань і виконання завдань для самостійної роботи, запропонованих на лекції.

*Література [2, 3, 10, 16]*

*Контрольні запитання для самоперевірки*.

1. Визначте та охарактеризуйте покоління мов програмування.
2. До якого покоління мов програмування належить С++?
3. Визначте та охарактеризуйте основні парадигми мов програмування.
4. Назвіть і опишіть основні види роботи зі створення програми.
5. Чому, створюючи програму, не можна відразу починати писати її текст?
6. Чим відрізняється компіляція від інтерпретації?
7. Надайте визначення тестування та сформулюйте його мету?
8. З якими видами діяльності тісно пов‘язане тестування?
9. В чому полягає функціональне тестування?
10. В чому полягає структурне тестування?
11. Чим різниться Альфа-тестування та Бета-тестування?
12. В чому сутність процесу налагодження програми?

*Контрольні запитання для надання письмових відповідей*.

1. Визначте види прояву помилок, які у вас зустрічалися? Яка була причина їх появи. Надайте фрагмент програмного коду.
2. Який вид та метод тестування Ви використовуєте при налагодженні своїх програм. Відповідь поясніть.