

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕПАРТАМЕНТ ОСВІТИ І НАУКИ
ДОНЕЦЬКОЇ ОБЛАСНОЇ ДЕРЖАВНОЇ АДМІНІСТРАЦІЇ
КОМУНАЛЬНИЙ ПОЗАШКІЛЬНИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ДОНЕЦЬКА ОБЛАСНА МАЛА АКАДЕМІЯ НАУК
УЧНІВСЬКОЇ МОЛОДІ»**

ПОГОДЖЕНО

Протокол засідання науково-методичної (експертної) ради
Донецького обласного інституту
післядипломної педагогічної освіти
№ _____ від _____ 20 р.

ЗАТВЕРДЖЕНО

Наказ директора департаменту
освіти і науки Донецької обласної
державної адміністрації
№ _____ від _____ 20 р.

**Навчальна програма з позашкільної освіти
дослідницько-експериментального напрямку
"ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ"**

**Основний рівень
1 рік навчання**

Укладач:

Величко В.Є., кандидат фізико-математичних наук, доцент, доцент кафедри методики навчання математики та методики навчання інформатики, ДВНЗ «Донбаський державний педагогічний університет»

Рецензенти:

Кадубовський О.А., кандидат фізико-математичних наук, доцент, декан фізико-математичного факультету ДВНЗ «Донбаський державний педагогічний університет»

Стьопкін А.В., кандидат фізико-математичних наук, доцент, доцент кафедри математики та інформатики ДВНЗ «Донбаський державний педагогічний університет»

СХВАЛЕНО

Протокол засідання педагогічної ради
комунального позашкільного навчального
закладу «Донецька обласна Мала академія
наук учнівської молоді»

№ _____

Пояснювальна записка

Актуальність навчальної програми пов'язана з широким проникненням інформаційних технологій практично в усі сфери сучасного життя та зростанням попиту на ІТ-фахівців. Учнівська молодь активно цікавиться вивченням інформаційних технологій, однією із складових частин якої є програмування.

Мета навчальної дисципліни – викладення основних понять алгоритмізації і техніки застосування у програмуванні базових алгоритмічних структур (організація програм) і базових структур даних (організація даних).

Для досягнення мети поставлені такі **основні завдання**:

вивчення основних етапів процесу проектування програмного забезпечення і визначення принципів процедурного програмування щодо розробки програм мовою С++;

вивчення типових підходів до розробки і аналізу найбільш розповсюджених алгоритмів рішення економіко-математичних задач;

здійснення аналізу можливостей сучасних інструментальних середовищ розробки програм (на прикладі середовища Visual С++.NET);

визначення концепцій і вивчення основних принципів організації програм.

Предметом навчальної дисципліни є теорія і практика застосування у програмуванні базових алгоритмічних структур і базових структур даних на базі сучасних технологій розробки програмного забезпечення.

Навчання групи 12–15 вихованців передбачене для дітей віком 15–17 років протягом 1 року (основний рівень - 216 годин, 6 годин на тиждень).

Основною формою проведення занять є групова. Практична робота виконується вихованцями як індивідуально, так і у групах, відповідно до обраної керівником гуртка методики, і передбачає виконання конкретного однакового для всіх завдання.

Навчально-тематичний план програми є орієнтовним.

Розділи розподілені на теми із зазначенням обсягу годин, необхідних для виконання кожного виду навчальних занять та самостійної роботи. Керівник гуртка може змінювати кількість теоретичних і практичних занять (залежно від того, як швидко та якісно вихованці засвоюють теоретичні знання та набувають практичних навичок), враховуючи рівень інтелектуального розвитку гуртківців, математичної підготовки. Різниця розподілу навчальних годин вказується в навчально-тематичному плані. Учні обирають теми творчих проектів самостійно за особистим уподобанням та інтересом, але відповідно до навчальної тематики, рівня складності, віку, а також за порадою керівника гуртка. Є можливість

виконувати індивідуальні, парні та колективні проекти. Основний принцип роботи педагога полягає в доступному викладенні учням матеріалу навчального курсу. Тому глибина вивчення тем залежить від рівня математичної підготовки учнів. Психологічний клімат у групі формується на основі творчості та взаємоповаги між учнями та викладачем.

Знання та вміння контролюються поточним оцінюванням виконання практичних і самостійних завдань, запланованих відповідно до програми, а також участю учнів у конкурсах, змаганнях і олімпіадах. Основними способами контролю ефективності навчання є тестування, співбесіда, система заліків, контрольні завдання, захист власних проектів, а також іспит.

При формуванні групи враховуються вікові особливості дітей та зміст програми.

До навчання в гуртку (секції, творчому об'єднанні тощо) зараховуються діти за результатами тестування, оскільки для опанування програми потрібно мати певний рівень знань і вмінь користувача-початківця.

У процесі навчання учні отримують необхідні знання під час проведення аудиторних занять: лекційних, практичних та лабораторних. Велике значення в процесі вивчення та закріплення знань має самостійна робота учнів і виконання ІНДЗ.

У результаті вивчення навчальної дисципліни учні повинні:

знати:

- основні етапи процесу проектування програмного забезпечення; типові алгоритмічні конструкції;
- принципи процедурного і структурованого програмування; особливості застосування сучасних базових інструментальних програмних засобів, призначених для вирішення економічних задач; базові типи даних;
- похідні типи даних: переліки, покажчики, посилання, масиви, структури, об'єднання;
- оператори управління програмою; команди передпроцесорної обробки; правила роботи з функціями; систему уведення-виведення C++; основні принципи роботи з файлами; правила роботи із шаблонами;
- принципи розробки Windows-додатків;
- основу побудови програм на керованому C++;

вміти:

- складати програми мовою C++, забезпечуючи: рішення задач з курсу вищої математики (чисельне диференціювання і інтеграція, рішення рівнянь і т.д.); створення і обробку структур, масивів структур; найпростішу обробку файлів; використання функцій; використання основних елементів призначеного для користувача інтерфейсу ОС; використання

сучасного інструментального програмного забезпечення;

○ користуватися раніше складеними програмами і здійснювати супровід програм, вносити зміни в програму, виконувати відладку програм за допомогою вбудованих інструментальних засобів.

1. Тематичний план навчальної дисципліни

З самого початку вивчення дисципліни кожен сучень має бути ознайомлений як з програмою дисципліни і формами організації навчання, так і зі структурою, змістом та обсягом кожного з її навчальних модулів, а також з усіма видами контролю та методикою оцінювання знань. Тематичний план дисципліни складається з двох змістових модулів, кожен з яких об'єднує в собі відносно окремих самостійний блок дисципліни, який логічно пов'язує кілька навчальних елементів дисципліни за змістом та взаємозв'язками (табл.1).

Таблиця 1

Структура навчальної дисципліни

Тема	Кількість годин		
	Теоретичних	Практичних	Разом
Розділ 1. Введення в розробку і кодування алгоритмів			
Тема 1. Алгоритм як основне поняття програмування. Лексичні основи мов високого рівня	8	12	20
Тема 2. Алгоритмічна мова C++. Основні типи даних	12	10	22
Тема 3. Програмування обчислювальних процесів. Оператори управління програмою	8	10	18
Тема 4. Функції	8	10	18
Тема 5. Похідні типи даних. Рядки	10	8	18

Тема 6. Введення в систему вводу-виводу C++	9	10	19
Усього за модулем	55	60	115
Розділ 2. Принципи розробки додатків			
Тема 7. Передпроцесорна обробка	4	10	14
Тема 8. Структури та об'єднання	10	10	20
Тема 9. Шаблони	10	10	20
Тема 10 Програмування в середовищі	12	10	22
Тема 11. Програмування на керованому C++	12	13	25
Усього за модулем	48	53	101
Усього	103	113	216

2.Зміст навчальної дисципліни за модулями та темами

Розділ 1. Введення в розробку і кодування алгоритмів

Тема 1. Алгоритм як основне поняття програмування.

Лексичні основи мов високого рівня

Поняття алгоритму. Властивості алгоритму. Типові алгоритмічні конструкції. Розробка алгоритму методом покрокового уточнення. Алгоритмічні конструкції: послідовність, вибір, повторення. Способи завдання алгоритму. Критерії оцінки алгоритмів.

Початкові відомості про технологію програмування: процедурне, структуроване і об'єктно-орієнтоване програмування. Мови програмування: процедурні, апликативні, системи правил, об'єктно-орієнтовні. Стандартизація мов та середовища проектування. Транслятори. Редактори. Компонувальники. Відладчики. Керуючі структури: оператори, вирази та підпрограми. Огляд сучасних інтегрованих систем програмування. Інтегроване середовище системи програмування Visual C++.NET. Платформа DOT.NET. Етапи розробки та впровадження програм. Вимоги до програмного коду.

Тема 2. Алгоритмічна мова C++. Основні типи даних

Стандарт ANSI. Структура C++ програми.

Лексичні елементи мови C++: алфавіт, коментарі, ідентифікатори, службові слова, дані, вираз, операнд, змінна, операція. Домовленості про імена.

Поняття типу даних. Класифікація і представлення даних. Базові типи даних: логічний, символний, цілий, речовинний. Перетворення типів: неявні перетворення, явні перетворення.

Пріоритети операцій. Зведена таблиця пріоритетності і асоціативності операцій.

Операції. Унарні операції: унарний мінус, унарний плюс, порозрядне інвертування, логічне заперечення, інкремент, декремент, операція обчислення розміру (sizeof). Бінарні операції: адитивні, мультиплікативні, зсувів, порозрядні, операції відносин, логічні, привласнення, операція "кома". Пріоритети операцій. Зведена таблиця пріоритетності і асоціативності операцій.

Стандартні математичні функції.

Константні величини: цілі, речовинні, перелічувальні, символні (літерні), рядкові (рядки або літерні рядки). Правила визначення компілятором констант. Визначення констант за допомогою ключового слова const. Константи

переліків.

Тема 3. Програмування обчислювальних процесів. Оператори управління програмою

Загальні відомості про систему вводу-виводу даних.

Вирази, символи пропусків, блоки і комплексні вирази. Операнд, змінна. Оператор привласнення. Оголошення та ініціалізація змінних.

Типи операторів. Найпростіший оператор, оператор-оголошення, оператор-визначення, оператор-вираз. Управляючі оператори: оператори проходження; оператори вибору (єдиний вибір – if, подвійний вибір – if / else, множинний вибір – switch, умовна операція); оператори повторення (оператор while, оператор do-while, оператор for). Вкладені цикли. Управляючі оператори в циклах: оператор break, оператор continue, оператор goto. Рекомендації по вибору циклів.

Тема 4. Функції

Загальні відомості про функції. Структура функції. Значення, параметри і аргументи, що повертаються. Оголошення функції. Прототипи функцій. Визначення функції. Виконання функції.

Локальні і глобальні змінні. Правило видимості змінних. Приведення типів аргументів функцій. Правила автоматичного (неявного) приведення типів. Явні перетворення типів. Правила роботи з функціями. Класи пам'яті. Список параметрів функції. Параметри за умовчанням. Способи передачі параметрів. Способи повернення значення. Функції, що підставляються.

Створення власних заголовних файлів. Перевантаження функцій. Рекурсія.

Робота функцій. Розбиття пам'яті. Стек і функції. Модифікатори функцій.

Тема 5. Похідні типи даних. Рядки

Масиви. Оголошення масивів. Ініціалізація масивів. Обробка одновимірних масивів даних економічного характеру. Алгоритми сортування масивів. Багатовимірні масиви. Ініціалізація багатовимірного масиву. Типові приклади обробки матриць.

Масиви як параметри функцій.

Рядки як масиви символів. Операції з рядками. Тип даних string.
Ввід-вивід рядків.

Поняття покажчика, посилання. Покажчики і масиви. Адресна арифметика. Посилання. Приклади використання покажчиків і посилань.

Показчики на функції. Посилання. Параметри функцій як посилання. Організація пам'яті в сучасних процесорах і покажчики мови C++. Моделі пам'яті. Статичні і динамічні змінні. Оператори new і delete. Динамічні масиви. Динамічні масиви як параметри функцій. Зв'язні списки.

Створення однозв'язного списку. Прохід однозв'язного списку. Включення нового елемента в існуючий список. Виключення елемента із списку.

Тема 6. Введення в систему вводу-виводу C++

Базові положення системи вводу-виводу C++. Потоки і буфери.

Стандартні об'єкти вводу-виводу.

Ввід даних за допомогою глобального об'єкта cin.

Вивід рядків. Введення одного символу. Використовування функції get(): без параметрів, з параметрами. Ввід рядків із стандартного пристрою введення. Використовування функції getline(). Вивід даних за допомогою глобального об'єкта cout. Очищення буфера виводу – flush(). Використовування функцій put() і write(). Ввід-вивід даних, що форматується. Маніпулятори вводу-виводу. Функції width(), precision(), fill(). Маніпулятори, що визначаються користувачем.

Використовування файлів для вводу-виводу даних. Створення файла. Створення потоку. Відкриття потоку. "Приєднання" файла до потоку. Обміни з файлом за допомогою потоку. "Від'єднання" потоку від файла. Закриття файла. Знищення файла.

Модуль 2. Принципи розробки додатків

Тема 7. Передпроцесорна обробка

Основи апарату макросів. Директиви препроцесора.

Директива препроцесора #include і файли, що включаються. Директива препроцесора #define: оголошення констант і макросів. Умовна компіляція. Використовування ключового слова typedef із структурами. Відмінність директиви #define від оператора typedef. Оператор defined.

Тема 8. Структури та об'єднання

Структури. Структури з бітовими полями. Вкладені структури. Доступ до елементів структур. Операції з структурами. Структури як параметри функцій.

Масиви структур. Показчики на структури. Передача з посилання масивів структур.

Об'єднання. Операції з об'єднаннями. Переліки. Функції роботи з датою та часом.

Тема 9. Шаблони

Основи апарату шаблонів. Шаблони функцій. Перевантаження шаблонів функцій. Шаблони функцій сортування.

Стандартна бібліотека шаблонів (STL). Призначення та склад STL. Контейнери. Робота з векторами, списками, стеками, чергами.

Тема 10. Програмування в середовищі

Компоненти та завдання операційної системи (ОС). Допоміжні модулі операційної системи. Характеристики ОС. Інтерфейс прикладного програмування (API). Утиліти, системні програми, які оброблюють бібліотеки процедур.

Особливості ОС. Графічні елементи вікна. Елементи призначеного для користувача інтерфейсу. Поняття повідомлення (формат повідомлення). Джерела отримання повідомлень в програмах. Цикл обробки повідомлень. Графічна схема типової програми. Каркасний додаток. Клас вікон. Головна функція. Створення вікна. Створення вікон з використанням існуючих класів. Реєстрація класу вікон. Обробка повідомлень у віконній функції. Функції підтримки вікон. Приєднання даних класу вікна до вікна. Зміна зовнішнього вигляду вікна.

Головна функція додатка. Типи даних, які використовуються. Цикл обробки повідомлень. Джерела повідомлень. Рентабельні функції. Функції перехвату повідомлень. Черги повідомлень.

Органи управління. Редактори тексту. Смуги прокрутки.

Ресурси Windows-додатків. Структура файлів ресурсу. Підключення ресурсів до виконавчого файлу. Таблиці рядків. Ресурси, які визначаються користувачем. Створення меню. Головне меню додатка. Спливаюче (контекстне) меню додатка. Повідомлення меню. Системне меню додатка. Меню без використання ресурсів. Додавання меню до вікна. Повідомлення меню.

Акселератори. Опис акселераторів у файлах ресурсів. Підключення таблиць акселераторів.

Діалогові вікна та їх елементи. Модальні та немодальні діалогові вікна.

Клавіатурний інтерфейс діалогового вікна. Динамічні діалогові вікна. Шаблони діалогового вікна.

Елементи управління діалогових вікон. Створення діалогових вікон. Класи передвизначених вікон. Списки прості та комбіновані. Нотифікаційні повідомлення та їх обробка. Стандартні діалогові вікна. Стандартні діалогові панелі. Панелі для відкриття або збереження файлів. Панелі вибору кольору. Панелі для вибору шрифтів.

Структура багатовіконного інтерфейсу. Головне вікно. Складання багатовіконного додатка. Дочірні вікна. Порядок взаємодії між вікнами. Додаткова інформація про обробку повідомлень в головному вікні.

Визволення забраних ресурсів. Особливості віконної процедури. Меню в багатовіконному додатку. Режими огляду та сортування.

Загальні відомості про контексти пристроїв. Концепція GDI. Інтерфейс графічних пристроїв. Ввід об'єктів до контексту пристроїв. Растрові зображення. Функції роботи з растровими зображеннями. Кольори та бітові образи. Формат монохромного бітового образу. Формат кольорового бітового образу. Кольорові профілі. Перетворення кольорів.

Піктограми. Створення піктограм. Функції роботи з піктограмами. Алгоритми рисування. Інструменти рисування. Перо, кість, палітра та області для рисування. Області виводу. Визначення області недійсної, відсікання та повідомлення WM_PAINT.

Режими рисування. Рисування базових графічних елементів.

Тема 11. Програмування на керованому C++

Поняття складки (Assembly). Відображення C++ на специфікацію CLS. Типи даних C++ і CLR. Директива #using та оператор using. Стандартний ввід-вивід. Керовані та некеровані типи. Типова безпека. Управління прибиранням сміття. Створення керованого коду, прапор компіляції /clr.

3. Самостійна робота учнів

Необхідним елементом успішного засвоєння навчального матеріалу дисципліни є самостійна робота учнів з вітчизняною та закордонною спеціальною літературою. Самостійна робота є основним засобом оволодіння навчальним матеріалом у час, вільний від обов'язкових навчальних занять. Самостійна робота учнів передбачає поглиблене вивчення тем з використанням рекомендованої літератури, пошук інформації в Інтернеті, а також додаткову роботу для виконання індивідуальних завдань. Основні види самостійної роботи, які запропоновані учням:

1. Вивчення лекційного матеріалу.
2. Робота з вивчення рекомендованої літератури.
3. Вивчення основних термінів та понять з галузі обчислювальної техніки і програмування.
4. Підготовка до лабораторних і практичних занять.
5. Робота над ІНДЗ.
6. Робота над рефератом.

Тематика рефератів по курсу

1. Історія розвитку мови програмування C++.
2. Типізація даних в C++: вбудовані, похідні типи і класи.
3. Призначення і використання покажчиків в C++.
4. Масиви та їх реалізація в C++.
5. Операції, які використовуються в мові програмування C++. Пріоритети виконання операцій.
6. Конструкції мови програмування C++, що управляють.
7. Робота з рядками в C++.
8. Типові математичні функції мови C++.
9. Стандартні потоки вводу-виводу.
10. Файлові потоки вводу-виводу.
11. Динамічний розподіл і звільнення пам'яті.
12. Загальна характеристика динамічних структур даних.

4. Контрольні запитання для самодіагностики

1. Структура типової програми мов C++.
2. Бібліотечні файли, їх організація та використання.
3. Опишіть можливості застосування інтегрованого середовища Visual C++ .NET для розробки Windows – додатків.
4. Дайте класифікацію типів даних, способи опису їх в програмах, приклади використання.
5. Перерахуйте структури мови C++, що управляють, і вкажіть особливості їх застосування.
6. Формат і робота умовного оператора в повній і скороченій формах. Наведіть приклади.
7. Формат і робота оператора **switch**. Наведіть приклади.
8. Формат і робота оператора циклу **while** (цикл з передумовою). Наведіть приклади.

9. Формат і робота оператора циклу **do/while** (цикл з післяумовою). Наведіть приклади.

10. Формат і робота оператора циклу **for**. Наведіть приклади.

11. Як записати два і більше виразів в умові **while**?

12. Чи відрізняється умова циклу в циклі **while** від умови циклу **do/while**?

Дайте визначення поняття "масив". До яких типів даних відносяться масиви в C++?

13. Наведіть приклад циклу **for** від 0 до 20. Коли виконується кожний з трьох виразів?

14. Де в заголовку **for** може використовуватися кома?

15. Чи можна опускати крапки з комою в циклі **for**?

16. Як працює оператор **break**?

17. Як працює оператор **continue**?

18. Як нумеруються елементи масиву `a[5]`?

19. Чи може розмірність масиву визначатися змінною?

20. Як ініціалізувати масив? Чи можна при цьому опускати розмірність?

21. Що буде, якщо в списку ініціалізації менше елементів, ніж розмірність масиву? А якщо більше?

22. Чи можна ініціалізувати символьний масив рядком символів?

23. Чи можна привласнити масив масиву?

24. Визначити поняття покажчика в C++. Чим відрізняються покажчики від посилань?

25. Що відбудеться, якщо буде вихід за межу масиву?

26. Що значить запис `a[1]` для масиву `a[5][5]`?

27. Що тут визначено: `int *i1, i2`?

28. Як визначається операція отримання адреси в C++?

29. Що таке покажчик типу `void*`? Чи можна його розіменувати?

30. Що позначає окремо ім'я `a` для масиву `a[10]`?

31. На що вказує `ptr + 10`, якщо `ptr` вказує на перший елемент масиву?

32. Чим відрізняється робота з масивом від роботи з покажчиком?

33. Як розмістити в C++ об'єкт у вільній пам'яті?

34. Як звільнити пам'ять об'єкта?

35. Опишіть структури даних – масиви. Поясніть організацію багатовимірних масивів. Для чого використовуються покажчики і посилання?

36. Програмні модулі C++ (функції). Визначення функцій, виклик, прототипи.

37. Створіть функцію, яка приймає три аргументи: ім'я масиву елементів

типу `int`, розмір масиву і значення типу `int`. Функція повинна встановити для кожного елементу масиву значення типу `int`.

38. Як здійснюється організація файлової системи в C++?
39. Дайте визначення поняття "потік" в C++. Назвіть і охарактеризуйте стандартні потоки вводу-виводу.
40. Опишіть ієрархію потокових класів в C++.
41. Назвіть засоби форматування вводу-виводу.
42. Вкажіть потоки файлового вводу-виводу.
43. Які операції використовуються для запису даних у файл і для читання з файлу?
44. У чому відмінність консольного застосування від графічного?
45. Призначення і склад головної функції `WinMain()`.
46. Призначення і склад структур `WNDCLASS` і `MSG`.
47. Перерахуйте характеристики класу вікна.
48. Опишіть алгоритм створення і показу вікна.
49. Призначення і принцип реалізації циклу обробки повідомлень.
50. Призначення і склад віконної функції.
51. Опишіть алгоритм створення кнопок в Windows-вікні.
52. Укажіть принципи обробки повідомлень `WM_CHAR`, `WM_LBUTTONDOWN`, `WM_RBUTTONDOWN` і повідомлень групи `WM_COMMAND`.
53. Укажіть призначення функцій `MessageBox()`, `GetWindowText()`, `SetWindowText()`, `SendMessage()`.
54. Як створювати діалогові вікна за допомогою редактора ресурсів?
55. Вкажіть призначення типових елементів управління діалогових вікон (`Button`, `Edit Control`, `Static Text`, `Radio Button`, `Check Box`) і формати їх опису у файлах ресурсів.
56. Перерахуйте призначення функцій `DialogBox()`, `GetDlgItemText()`, `GetDlgItem()`, `SetWindowText()`.
57. Сформулюйте концепцію побудови інтерфейсу графічних пристроїв (GDI).
58. У чому суть контекстного представлення пристроїв?
59. Призначення бібліотеки STL?
60. У чому особливості обробки події `WM_PAINT`.
61. Перерахуйте основні директиви компілятора.
62. Наведіть приклади функцій що працюють зі списками.
63. Вкажіть призначення файлів шаблонного проекту простого Windows-додатка.

64. Опишіть ресурси, що використовуються в шаблонному проекті і механізми їх модифікації.

5. Рекомендована література

5.1 Основна

1. Браткевич В. В Основы программирования и алгоритмические языки (язык программирования C++): Конспект лекций Ч. 1 / В. В. Браткевич, Ю. В. Перколаб, Л. И. Лукашева. – Харьков: Изд. ХГЭУ, 2001. – 68 с.

2. Браткевич В. В. Основы программирования и алгоритмические языки (язык программирования C++): Конспект лекций Ч. 2 / В. В. Браткевич, Ю. В. Перколаб, Л. И. Лукашева. – Харьков: Изд. ХГЭУ, 2002. – 152 с.

3. Методические рекомендации к лабораторным работам по курсу "Основы программирования и алгоритмические языки" для студентов специальности 7.080401 дневной формы обучения. Ч. 1. / Сост.: Ю. В. Перколаб, В. В. Браткевич, Л. И. Лукашева. – Харьков: Изд. ХГЭУ, 2001. – 68 с.

4. Методические рекомендации к лабораторным работам по курсу "Основы программирования и алгоритмические языки" для студентов специальности 7.080401 дневной формы обучения. Ч. 2. / Сост.: Ю. В. Перколаб, В. В. Браткевич, Л. И. Лукашева. – Харьков: Изд. ХГЭУ, 2002. – 68 с.

5. Методичні рекомендації до виконання робіт з курсу "Основи програмування і алгоритмічні мови" (робота з файлами) для студентів спеціальностей 7.080401, 7.080407 усіх форм навчання. Ч. 3 / Укл. Ю. В. Перколаб, В. В. Браткевич, І. О. Бондар. – Харків. Вид. ХНЕУ, 2005. – 64 с.

6. Пирогов В. Ю. Программирование на Visual C++.NET. – СПб.: БХВ-Петербург, 2003. – 800 с.

7. Румянцев П. В. Азбука программирования в Win32 API. – М.: Горячая линия – Телеком 2001. – 312 с.

8. Харви Дейтел. Как программировать на C++ / Харви Дейтел, Пол Дейтел. [Пер. с англ. – М.: ЗАО Изд. "БИНОМ", 2001 г. – 1024 с.

9. Шеферд Д. Программирование на Microsoft Visual C++.NET / Пер. с англ. – М.: Издательско-торговый дом "Русская редакция", 2003. – 928 с.

5.2 Додаткова

10. Верма Р. Д. Справочник по функциям Win32 API. – М.: Горячая линия – Телеком, 2002. – 488 с.

11. Методические рекомендации к выполнению лабораторных работ по курсу "Системное программирование" для студентов всех форм обучения по специальности 7.080401. Ч. 1 / Сост. Д. Ю. Голубничий, В. Ф. Третьяк.

12. Павловская Т. А. C/C++. Программирование на языке высокого уровня. – СПб.: Питер, 2006. – 461 с.

13. Подбельский В. В. Язык C++: Учеб. пособие. – 4-е изд. – М.:

Финансы и статистика, 1999. – 560 с.

14. Саймон Р. Windows 2000 API. Энциклопедия программиста.
/ Пер. с англ. – К.: ООО "ДиасофтЮП", 2002. – 1088 с.

15. Страуструп Б. Язык программирования C++. / Пер. с англ. – 3-е изд. – СПб.; М.: "Невский диалект" – Издательство БИНОМ", 1999 – 991 с.

5.3 Ресурси мережі Internet

16. Опис стандарту C++ANSI / ISO // <http://reality.sqi.com/austern/std-c++/fag.html>

17. Підручник для початківців що вивчають Microsoft Visual C++ // http://chesworth.com/pv/languages/c/visual_cpptutorial.htm

18. Список питань, що часто задаються (з відповідями) / <http://www.math.uio.no/nett/fag/C-fag/fag.html>