

Міністерство освіти і науки України

Національний університет водного господарства та
природокористування

Кафедра комп'ютерних наук та прикладної математики

04-01-65M

Методичні вказівки

до лабораторних робіт з навчальної дисципліни
«Програмування»
для здобувачів вищої освіти першого (бакалаврського)
рівня за освітньо-професійною програмою «Комп'ютерні
науки» спеціальності 122 «Комп'ютерні науки» денної
форми навчання

Рекомендовано науково-
методичною радою з якості
ННІАКОТ
Протокол № 9 від 30.08.2021 р.

Рівне – 2021

Методичні вказівки до лабораторних робіт з навчальної дисципліни «Програмування» для здобувачів вищої освіти першого (бакалаврського) рівня за освітньо-професійною програмою «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» денної форми навчання [Електронне видання] / Іванчук Н. В. – Рівне : НУВГП, 2021. – 28 с.

Укладач: Іванчук Н. В., к.т.н., доцент кафедри комп'ютерних наук та прикладної математики.

Відповідальний за випуск: Турбал Ю. В., д.т.н., професор, завідувач кафедри комп'ютерних наук та прикладної математики.

Керівник (гарант) освітньої програми
«Комп'ютерні науки»
спеціальності
122 «Комп'ютерні науки»

Мартинюк П. М.

© Н. В. Іванчук, 2021
© НУВГП, 2021

ЗМІСТ

Вступ	4
Лабораторна робота 1. Оптимізація програми	6
Лабораторна робота 2. Використання ООП	7
Лабораторна робота 3. Перевантаження в ООП	10
Лабораторна робота 4. "Дружба" в ООП	13
Лабораторна робота 5. Робота з класом String	15
Лабораторна робота 6. Шаблони класів	18
Лабораторна робота 7. Динамічні структури даних ...	22
Лабораторна робота 8. Написання консольної гри	26
Рекомендована література	27

Вступ

Програма нормативної навчальної дисципліни «Програмування» складена відповідно до освітньо-професійної програми підготовки бакалавра за спеціальністю «Комп'ютерні науки». Міждисциплінарні зв'язки: дисципліна є складовою частиною блоку фундаментальної підготовки та відноситься до навчальних дисциплін циклу загальної підготовки студентів. Дисципліна «Програмування» спрямована на вивчення основ алгоритмізації, особливостей програмування інженерно-технічних та наукових задач мовою високого рівня, методів побудови та використання програмних комплексів, на опанування студентами практичних навиків програмного розв'язання математичних задач. Під час вивчення даної дисципліни студенти здобудуть знання, які допоможуть ефективно використовувати сучасні розробки технологій програмування при вивченні спеціальних дисциплін.

Мета: засвоєння студентами основних понять і методів алгоритмізації, програмування та розв'язку інженерно-економічних задач з використанням персонального комп'ютера, мови програмування C++ та сучасних технологій програмування, формування наукового світогляду та здатності до засвоєння та постійного оновлення професійних знань, оскільки на сьогоднішній момент C++ - одна з найпотужніших і затребуваних мов програмування.

Завдання: сформувати у студентів знання методів алгоритмізації та розв'язання практичних задач з використанням сучасних технологій програмування; підготувати студентів до ефективного використання технологій програмування при вивченні спеціальних дисциплін, підготувати студентів до використання

отриманих знань і навиків при розв'язуванні практичних задач, а також при написанні кваліфікаційних та магістерських робіт.

У результаті вивчення навчальної дисципліни студент повинен знати:

1. Класичні і сучасні методи та прийоми програмування.
2. Базові засади ООП.
3. Поняття розумного вказівника.
4. Особливості перевантаження.
5. Принципи наслідування.

Вміти: а) застосовувати прийоми та методи підготовки задач для їх розв'язання на ПК; б) вивчити основні засоби та прийоми алгоритмізації і програмування типових обчислювальних процесів; в) вивчити методи структурного та об'єктно-орієнтованого програмування; г) отримати практичні навички роботи в середовищах програмування C/C++; е) працювати з дружніми функціями; є) працювати з динамічними структурами даних; ж) використовувати потоки.

Лабораторна робота 1 **Оптимізація програми**

Завдання: оптимізувати по часу код програми, написаної на C++. Продемонструвати час виконання блоку коду до оптимізації та час виконання оптимізованого блоку програми.

Лабораторна робота 2 **Використання ООП**

Завдання: мовою C++ та з використанням ООП виконати наступні завдання згідно варіантів.

1. Написати клас, який містить набір функцій, що виконують різні дії з комплексними числами. Забезпечити перевірку правильності вводу типу чисел.
2. Описати клас *Complex*, що містить поля даних - дійсну та уявну частину комплексного числа. Написати програму для ілюстрації доступу до полів даних з різними специфікаторами доступу. Забезпечити перевірку правильності вводу типу чисел.
3. Написати клас, який реалізує роботу з додатніми раціональними числами. Забезпечити перевірку правильності вводу типу чисел.
4. Написати клас, який реалізує роботу з натуральними числами. Забезпечити перевірку правильності вводу типу чисел.
5. Написати клас, який реалізує роботу з дійсними числами. Забезпечити перевірку правильності вводу типу чисел.
6. Написати клас, який реалізує роботу з парними цілими числами. Забезпечити перевірку правильності вводу типу чисел.
7. Написати клас, який реалізує роботу з числами Фібоначчі. Забезпечити перевірку правильності вводу типу

чисел.

8. Написати клас, який реалізує роботу з простими цілими числами. Забезпечити перевірку правильності вводу типу чисел.

9. Написати клас, який реалізує роботу з непарними цілими числами. Забезпечити перевірку правильності вводу типу чисел.

10. Написати клас, який реалізує роботу з досконалими цілими числами. Забезпечити перевірку правильності вводу типу чисел.

11. Написати клас, який реалізує роботу з алгебраїчними комплексними числами. Забезпечити перевірку правильності вводу типу чисел.

12. Написати клас, який реалізує роботу з трансцендентними комплексними числами. Забезпечити перевірку правильності вводу типу чисел.

13. Написати клас, який реалізує роботу з від'ємними цілими числами. Забезпечити перевірку правильності вводу типу чисел.

14. Написати клас, який реалізує роботу з додатніми цілими числами. Забезпечити перевірку правильності вводу типу чисел.

15. Написати клас, який реалізує роботу з ірраціональними комплексними числами. Забезпечити перевірку правильності вводу типу чисел.

16. Написати клас, який реалізує роботу з від'ємними раціональними числами. Забезпечити перевірку правильності вводу типу чисел.

Лабораторна робота 3 Перевантаження в ООП

Завдання: мовою C++ та з використанням ООП виконати наступні завдання згідно варіантів.

1. Створити клас з перевантаженими функціями-членами розв'язку СЛАР різними методами.
2. Описати клас *Draw*, який містить кілька функцій, що малюють різні геометричні фігури. Використати його в програмі.
3. Придумати корисний тип даних, який не є вбудованим в жодній з мов програмування та реалізувати його як абстрактний тип даних C++. Описати відповідні конструктори та деструктори. Написати програму, що використовує цей тип даних.
4. Написати клас, що описує групу студентів. Опис студента також реалізується за допомогою відповідного класу. Використати при цьому перевантаження.
5. Розробити клас *Person*, який містить відповідні члени для зберігання:
 - a. імені,
 - b. віку,
 - c. статі і
 - d. телефонного номера.Напишіть перевантажені функції-члени, які зможуть змінювати ці члени даних індивідуально.

6. Розробити клас *String*, який в подальшому буде використовуватися для роботи з рядками. Клас повинен містити:

- a. конструктор за замовчуванням, що дозволяє створити рядок довжиною 80 символів;
- b. конструктор, що дозволяє створювати рядок довільного розміру;
- c. конструктор, який створює рядок і ініціалізує його рядком, отриманим від користувача.

Клас повинен містити методи для введення рядків з клавіатури і виведення рядків на екран. Використати при цьому перевантаження.

7. Написати систему перевантажених функцій, які виділяють пам'ять розміром $n \cdot r$ байт та повертають вказівник на тип першого аргумента, де n - ціле значення другого аргумента, r - розмір в байтах першого аргумента.

8. Перевантаживши операції $< >$, написати функцію сортування масиву об'єктів за деякими ознаками його полів: в порядку зростання (спадання) значень одного з полів типу `char`, `int`, `float` чи `double`.

9. Перевантаживши операції $< >$, написати функцію сортування масиву об'єктів за деякими ознаками його полів: в порядку лексикографічної впорядкованості полів типу `*char`.

10. Перевантажити оператор доступу до члена класу так, щоб він повертав об'єкт.

11. Створіть клас **Date**, який буде містити інформацію про дату (день, місяць, рік). За допомогою механізму

перевантаження операторів, визначте операцію різниці двох дат (результат у вигляді кількості днів між датами), а також операцію збільшення дати на певну кількість днів.

12. Створіть клас динамічного масиву, в якому реалізована перевірка виходу за межі масиву. Перевантажте оператори: [], +, -.

13. Створіть клас динамічного масиву, в якому реалізована перевірка виходу за межі масиву. Перевантажте оператори: =, ++ (додавання елемента в кінець масиву), -- (видалення елемента з кінця масиву).

14. Додати в стрічковий клас функцію, яка створює рядок, що містить перетин двох рядків, тобто загальні символи для двох рядків. Наприклад, результатом перетину рядків "sdqcg" "rgfas34" буде рядок "sg". Для реалізації функції перевантажити оператор * (бінарне множення).

15. C++ не перевіряє автоматично під час виконання ситуацію виходу індекса за границю масива. Розробити тип даних масив, який виконує перевірку границь.

16. Написати тип string, що має ряд специфічних властивостей (наприклад, лічильник звернень, перевантажені оператори +, -, тощо).

Лабораторна робота 4 **"Дружба" в ООП**

Завдання: мовою C++ та з використанням ООП і «дружби» виконати наступні завдання згідно варіантів.

1. Створити клас для множення матриць (елементи матриць вводяться з клавіатури користувачем).
2. Створити клас для транспонування матриць (кількість матриць, їх розмірність та елементи вводяться з клавіатури користувачем).
3. Створити клас для присвоєння матриць одна одній (напрямок присвоєння, розмірність матриць та елементи вводяться з клавіатури користувачем).
4. Створити клас для встановлення і отримання довільного елемента матриці (кількість матриць, їх розмірність та елементи вводяться з клавіатури користувачем).
5. Створити клас для множення матриць на скаляр (кількість матриць, їх розмірність, елементи та скаляр вводяться з клавіатури користувачем).
6. Створити клас, який перевіряє чи є матриця симетричною (розмірність матриці та її елементи вводяться з клавіатури користувачем).
7. Створити клас, який перевіряє чи є матриця нормальною (розмірність матриці та її елементи вводяться з клавіатури користувачем).

- 8.** Створити клас який перевіряє чи матриця є ідеальною (розмірність матриці та її елементи вводяться з клавіатури користувачем).
- 9.** Створити клас, який перевіряє чи є матриця діагональною (розмірність матриці та її елементи вводяться з клавіатури користувачем).
- 10.** Створити клас для знаходження оберненої матриці (елементи матриці вводяться з клавіатури користувачем).
- 11.** Створити клас, який перевіряє чи є матриця нижньотрикутною (розмірність матриці та її елементи вводяться з клавіатури користувачем).
- 12.** Створити клас, який перевіряє чи є матриця верхньотрикутною (розмірність матриці та її елементи вводяться з клавіатури користувачем).
- 13.** Створити клас, який знаходить визначник матриці (розмірність матриці та її елементи вводяться з клавіатури користувачем).
- 14.** Створити клас, який знаходить слід матриці (розмірність матриці та її елементи вводяться з клавіатури користувачем).
- 15.** Створити клас, який перевіряє чи є матриця ортогональною (розмірність матриці та її елементи вводяться з клавіатури користувачем).
- 16.** Створити клас для додавання матриць (кількість матриць, їх розмірність та елементи вводяться з клавіатури користувачем).

Лабораторна робота 5 **Робота з класом String**

Завдання: мовою C++ та з використанням ООП виконати наступні завдання згідно варіантів.

1. Користувач вводить з клавіатури рядок символів. Ознака кінця рядка - символ '**n**' (перехід на новий рядок). Рядок складається з слів, які відокремлені один від одного пробілами. Вивести на екран найдовше слово і його порядковий номер.

2. Визначити, чи є введене користувачем число складеним.

3. Користувач вводить з клавіатури рядок символів, який обов'язково закінчується символом точки. Видалити з рядка перші літери кожного слова.

4. Напишіть функцію, яка приймає два цілих числа **n** і **k** (введених користувачем з клавіатури) і повертає число, що містить **k** перших цифр числа **n**.

5. Час задається в форматі година, хвилина, секунда. Реалізувати:

a. віднімання з часу вказаної користувачем кількості секунд;

b. підрахунок числа секунд між двома моментами часу, що лежать в межах однієї доби.

6. Виконати перетворення матриці, а саме - перестановку рядків і стовпців. Для квадратної матриці розміром **n** переставляйте стовпці і рядки таким чином, щоб

елемент матриці з максимальним значенням, по модулю, розташовувався в нижньому правому куті матриці. Заповнення початкової матриці організувати з клавіатури.

7. Поміняти приголосні букви на голосні у введеному користувачем рядку, а голосні - на відповідний ASCII код, використовуючи функції.

8. Організувати введення рядка, кожне слово в рядку відділяється від інших слів пробілами, їх може бути необмежена кількість. Знайти найкоротше слово в рядку.

9. Дана матриця. Знайти суму елементів в тих стовпцях, які не містять від'ємні елементи. Кількість стовпців і рядків матриці повинен вводити користувач. За введеними даним, динамічно, повинна виділитися пам'ять під матрицю. Матрицю заповнювати випадковими значеннями, як від'ємними, так і додатними.

10. Написати фрагмент програми, що видаляє i -й (вводиться користувачем) елемент з масиву цілих чисел розміру N , $i \leq N$. Фрагмент оформити у вигляді функції, масив передати в функцію через параметри.

11. Знайти частку і залишок від ділення використовуючи тільки операції додавання і віднімання (числа вводяться з клавіатури).

12. Дано масив розміру n (вводиться користувачем), заповнити його випадковими числами. Знайти всі непарні числа масиву.

- 13.** Задана матриця A розмірністю $n \times m$ (вводяться користувачем). Записати всі елементи матриці в одновимірний масив.
- 14.** У масиві 7×7 цілого типу (заповненому випадковим чином) в кожному стовпці впорядкувати елементи, розташовані між максимальним і мінімальним.
- 15.** Потрібно обчислити вираз в якому є дужки (круглі, квадратні, фігурні) і знаки множення, ділення, додавання, віднімання. Довжина виразу не повинна перевищувати мільйон символів. Вираз може бути будь-яким, програма повинна вирішувати всі види виразів.
- 16.** Необхідно написати програму, яка рахує за допомогою вкладених циклів і оптимізувати її так, щоб при додаванні лічильника часу роботи програми, була помітна оптимізація вбудованих циклів.

Лабораторна робота 6 **Шаблони класів**

Завдання: мовою C++ та з використанням ООП виконати наступні завдання згідно варіантів.

1. Написати функціональний **шаблон** сортування масиву елементів довільного базового типу. Порівняти розмір виконуваного коду шаблону та відповідних функцій.
2. Написати функціональний **шаблон** сортування масиву елементів довільного типу (включаючи класи користувача).
3. Написати програму для ілюстрації дій компілятора при виклиці функції. Для цього визначити деяку функцію, функціональний **шаблон** та розглянути випадок коли параметри при виклиці функції точно відповідають тим, що вказані при її описі.
4. Написати програму для ілюстрації дій компілятора при виклиці функції. Для цього визначити деяку функцію, функціональний **шаблон** та розглянути випадок коли параметри при виклиці в точності не відповідають вказаним при описі, але існує шаблон, який дозволяє згенерувати функцію з точною відповідністю параметрів.
5. Написати програму для ілюстрації дій компілятора при виклиці функції. Для цього визначити деяку функцію, функціональний шаблон та розглянути випадок коли не існує функції чи **шаблону** з точною відповідністю параметрів, але знаходимось в межах правил перетворення типів.

6. Розробити **Двонаправлений** список. Організувати додавання елемента в кінець списку і видалення зі списку першого елемента.

7. Написати програму, яка повинна виконувати наступні операції:

- **Створювати двозв'язний список,**
- **Операцію доповнення списку.**

8. Написати програму, яка повинна виконувати наступні операції:

- **Створювати двозв'язний список,**
- **Операцію видалення елементів списку.**

9. Написати програму, яка повинна виконувати наступні операції:

- **Створювати двозв'язний список,**
- **Операцію перегляду списку.**
- **Збереження списку в файл.**

10. У програму вводиться набір символів, що закінчується символом крапки. Введення організувати з клавіатури або з файлу, вибір надається користувачеві. Набір символів потрібно розбити на слова і сформуванати **Однозв'язний** список, інформаційне поле елементів якого містить посилання на чергове слово. Словом вважається послідовність літер латинського алфавіту довільної довжини. Всі символи, відмінні від букв латинського алфавіту вважаються роздільниками слів.

11. Запрограмувати функції для роботи з **Пріоритетною чергою**. Черга запитів формується відповідно до пріоритету, зняття виконується підряд, починаючи з молодших адрес (тобто з початку черги). Черга повинна являти собою масив, в якому повинен виконуватися зсув після кожного читання і зсув - після досягнення границі пам'яті, яка виділена для черги. Пріоритет: мінімальне значення числового параметра, при збігу параметрів – LIFO.

12. Запрограмувати функції для роботи з **Пріоритетною чергою**. Черга запитів формується відповідно до пріоритету, зняття виконується підряд, починаючи з молодших адрес (тобто з початку черги). Черга повинна являти собою масив, в якому повинен виконуватися зсув після кожного читання і зсув - після досягнення границі пам'яті, яка виділена для черги. Пріоритет: максимальне значення числового параметра, при збігу параметрів – LIFO.

13. Розробити додаток, що імітує **Чергу** друку принтера. Повинні бути клієнти, що посилають запити на принтер, у кожного з яких є свій пріоритет. Кожен новий клієнт потрапляє в чергу в залежності від свого пріоритету.

14. Додати в клас **Однозв'язний список** наступні функції: вставка елемента в задану позицію, видалення елемента з заданої позиції, пошук заданого елемента (функція повертає позицію знайденого елемента в разі успіху або NULL в разі невдачі).

15. Додати в клас **Стек** наступні функції: додавання елемента в стек, видалення елемента зі стека, перевірку чи стек порожній.

16. Реалізувати шаблонний клас **Черга** на основі двозв'язного списку.

Лабораторна робота 7 **Динамічні структури даних**

Завдання: мовою C++ та з використанням ООП виконати наступні завдання згідно варіантів.

1. Описавши клас **myfile**, полем якого виступає змінна типу ***FILE**, визначити ряд власних функцій роботи з файлом (відкриття в текстовому режимі, запису та зчитування інформації, закриття), використовуючи відповідні бібліотечні функції C++. Для відкриття файлу використовувати конструктори файлових потоків.

2. Створити шаблонний клас-контейнер **Array**, який являє собою масив, що дозволяє зберігати об'єкти заданого типу. Клас повинен реалізовувати такі функції:

GetSize - отримання розміру масиву (кількість елементів, під які виділено пам'ять);

SetSize – встановлення розміру масиву;

GetUpperBound - отримання останнього допустимого індексу в масиві;

3. Створити шаблонний клас-контейнер **Array**, який являє собою масив, що дозволяє зберігати об'єкти заданого типу. Клас повинен реалізовувати такі функції:

IsEmpty - масив порожній?

FreeExtra - видалити "зайву" пам'ять;

RemoveAll - видалити всі;

4. Створити шаблонний клас-контейнер **Array**, який являє собою масив, що дозволяє зберігати об'єкти заданого типу. Клас повинен реалізовувати такі функції:

GetAt - одержання певного елемента (за індексом);

SetAt - встановлення нового значення для певного елемента (індекс елемента повинен бути менше поточного розміру масиву);

operator [] - для реалізації двох попередніх функцій;

5. Створити шаблонний клас-контейнер **Array**, який являє собою масив, що дозволяє зберігати об'єкти заданого типу. Клас повинен реалізовувати такі функції:

Add - додавання елемента в масив;

Append - "складання" двох масивів;

operator [] - для реалізації двох попередніх функцій;

6. Створити шаблонний клас-контейнер **Array**, який являє собою масив, що дозволяє зберігати об'єкти заданого типу. Клас повинен реалізовувати такі функції:

GetData - отримання адреси масиву з даними;

Insert - вставка елемента (-ів) в задану позицію;

RemoveAt - видалення елемента (-ів) із заданою позиції;

7. Створити телефонний довідник для здійснення наступних операцій:

Додавання абонентів в базу.

Видалення абонентів з бази.

Збереження бази в файл.

8. Створити телефонний довідник для здійснення наступних операцій:

Модифікація даних абонента.

**Пошук абонентів за номером.
Завантаження бази з файлу.**

9. Створити телефонний довідник для здійснення наступних операцій:

**Пошук абонентів за прізвищем.
Можливість збереження знайденої інформації в файл.**

10. Створити телефонний довідник для здійснення наступних операцій:

**Роздруківка в алфавітному порядку абонентів із заданого діапазону номерів та прізвищ;
Можливість збереження знайденої інформації в файл.**

11. Напишіть програму, яка сортує записи в файлі в порядку зростання номера телефону. У текстовому файлі знаходяться записи про номери телефонів. У кожному рядку записана інформація: номер телефону, ПІБ, адреса.

12. У текстовому файлі, що містить текст програми на мові C++, перевірити відповідність фігурних дужок { і }. Результат перевірки вивести на екран і записати у вигляді фрази в текстовий файл. Результат роботи програми (висновок) помістити в окремий текстовий файл, продублювати на екрані.

13. Є файл, в якому записаний деякий текст англійською мовою. Потрібно вивести в алфавітному порядку всі букви, які зустрічаються в файлі. Кожну букву виводити тільки один раз.

14. У файлі **test.in** записана матриця розміром $N*N$. Знайти її визначник. Результат записати в **test.out**. Для роботи з файлами використовувати функції з заголовного файлу **cstdio**.

15. Описати клас **Рядок**. Обов'язкові поля класу:

- вказівник **pChar** зберігає адресу динамічно виділеної пам'яті для розміщення символів рядка;
- значення типу **word** зберігає довжину рядка в байтах.

Реалізувати обов'язковий метод наступного призначення:

- конструктор без параметрів;

16. Описати базовий клас **Рядок**. Обов'язкові поля класу:

- конструктор, що приймає як параметр символ;
- - метод отримання довжини рядка;
- - метод очищення рядки (робить рядок порожнім);
- - деструктор.

Лабораторна робота 8
Написання консольної гри

Завдання: мовою програмування C++ та з використанням ООП написати власну консольну гру.

Рекомендована література

1. Давыдов В. Г. Технологии программирования С++. СПб. : БХВ, 2005. 672 с.
2. Прата С. Язык программирования С++: Лекции и упражнения. Москва : ДиаСофтЮП, 2005. 1104 с.
3. Программирование на С++ : учебное пособие / Под ред. А. Д. Хомоненко. СПб. : КОРОНАПРИНТ, 2003. 512 с.
4. Трофименко О. Г., Прокоп Ю. В., Швайко І. Г., Буката Л. М. та ін. С++. Основи програмування. Теорія та практика : підручник /за ред. О. Г. Трофименко. Одеса : Фенікс, 2010. 544 с.
5. Шилдт Г. Самоучитель С++: Пер. с англ. - 3-е изд. СПб. : БХВ-Петербург, 2005. 688 с.
6. Эджер Д. Библиотека программиста С++. М. : ООО «И.Д. Вильямс», 2008. 1376 с.
7. Александреску А. Современное программирование на С++. М. : ООО «И. Д. Вильямс», 2008. 336 с.
8. Власюк А. П., Мартинюк П. М., Прищепа О. В., Філатова І. А., Філатов М. С., Рощенюк А. М., Демчук О. С., Демчук М. Б., Мічута О. Р., Цветкова Т. П., Федорчук Н. А. Лабораторний практикум з програмування : навч. посібник / за ред. А. П. Власюка. Рівне : НУВГП, 2010. 495 с.
9. Ишкова Э. А. С++. Начала программирования. Москва : БИНОМ, 2004. 368 с.
10. Лафоре Р. Объектно-ориентированное программирование в С++. СПб. : Питер, 2003. 928 с.
11. Павловская Т. А. Программирование на языке высокого уровня С/С++. СПб. : Питер, 2002. 464 с.
12. Павловская Т. А., Щупак Ю. А. С/С++. Структурное программирование : практикум. СПб. : Питер, 2003. 240 с.
13. Пильщиков В. Н., Горячая И. В., Бордаченкова Е. А. Решение задач с использованием рекурсии : учебно-методическое пособие. Москва, 2012. 37 с.

14. Пономарёв В. Программирование на C++/C# в Visual Studio .NET 2003. СПб. : БХВ, 2004. 352 с.