

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Учреждение образования
«Витебский государственный технологический университет»

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

**Методические указания по выполнению курсовых работ
для студентов специальности
6-05-0611-01 «Информационные системы и технологии»**

Витебск
2025

УДК 004.4
ББК 32.97
О-75

Составители:

А. С. Соколова, В. Е. Казаков

Одобрено кафедрой «Информационные системы и технологии»
УО «ВГТУ», протокол № 10 от 13.03.2025.

Рекомендовано к изданию редакционно-издательским
советом УО «ВГТУ», протокол № 7 от 26.03.2025.

Основы алгоритмизации и программирования : методические указания по выполнению курсовых работ / А. С. Соколова, В. Е. Казаков. – Витебск : УО «ВГТУ», 2025. – 31 с.

Методические указания являются руководством по организации и выполнению курсовой работы по учебной дисциплине «Основы алгоритмизации и программирования». Приводятся содержание и правила оформления расчетно-пояснительной записки. Представлены материалы, поясняющие выполнение курсовой работы, список рекомендуемой литературы.

Методические указания предназначены для студентов специальности 6-05-0611-01 «Информационные системы и технологии».

Издание в электронном виде расположено в репозитории библиотеки УО «ВГТУ».

УДК 004.4
ББК 32.97

© УО «ВГТУ», 2025

СОДЕРЖАНИЕ

1	ЦЕЛИ И ЗАДАЧИ КУРСОВОГО ПРОЕКТА.....	4
2	ОРГАНИЗАЦИЯ КУРСОВОГО ПРОЕКТИРОВАНИЯ.....	4
3	ПОСТАНОВКА ЗАДАЧИ НА ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ	6
4	СТРУКТУРА И СОДЕРЖАНИЕ КУРСОВОЙ РАБОТЫ	7
4.1	Общие сведения о динамической структуре	8
4.2	Проектирование класса динамической структуры.....	10
4.3	Реализация класса динамической структуры	11
4.4	Решение поставленной задачи.....	14
5	ОФОРМЛЕНИЕ РАСЧЕТНО-ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ.....	17
	ЛИТЕРАТУРА	27
	ПРИЛОЖЕНИЕ А. Пример листа задания на курсовую работу	28
	ПРИЛОЖЕНИЕ Б. Пример титульного листа курсовой работы	30

1 ЦЕЛИ И ЗАДАЧИ КУРСОВОГО ПРОЕКТА

Целью курсовой работы является систематизация и углубление теоретических знаний и практических навыков, полученных в ходе изучения дисциплины «Основы алгоритмизации и программирования».

Задачами курсовой работы являются:

- закрепление знаний о динамических структурах данных;
- закрепление знаний основ объектно-ориентированного программирования;
- формирование навыков применения алгоритмов обработки данных при решении прикладных инженерных задач;
- развитие системного мышления у студентов.

2 ОРГАНИЗАЦИЯ КУРСОВОГО ПРОЕКТИРОВАНИЯ

Темы курсовых работ должны быть разработаны руководителем курсового проектирования и утверждены заведующим кафедрой до начала семестра, в котором предусмотрено выполнение курсовой работы по учебной дисциплине «Основы алгоритмизации и программирования».

Обучающийся вправе выбрать тему курсовой работы из числа утвержденных на кафедре или самостоятельно предложить тему курсовой работы с обоснованием ее целесообразности.

Задание на курсовую работу должно содержать данные, достаточные для постановки задач проектирования, формулировки проблемной ситуации.

Задание по курсовому проектированию разрабатывается руководителем курсовой работы, утверждается заведующим кафедрой и выдается обучающемуся очной формы получения высшего образования в первые две недели после начала семестра, в котором учебным планом предусмотрена курсовая работа.

Обучающимся заочной формы получения высшего образования задание выдается на лабораторно-экзаменационной сессии, предшествующей семестру, в котором учебным планом предусмотрено его выполнение.

Форма задания на курсовое проектирование приведена в приложении А.

На период выполнения курсовой работы распоряжением заведующего кафедрой утверждается расписание групповых и/или индивидуальных консультаций.

Консультации по курсовой работы организуются еженедельно в дополнительное от расписания учебных занятий время. Допускаются дистанционные консультации с использованием коммуникационных средств.

График выполнения курсовой работы утверждается заведующим

кафедрой и доводится до сведения обучающихся (помещаются на информационном стенде кафедры, в СДО университета и др.).

Обучающиеся обязаны выполнять курсовую работу *самостоятельно*, без привлечения третьих лиц, не допускать плагиата, некорректных заимствований, фальсификации и подлога материалов

Ответственность за своевременное выполнение графика курсового проектирования, принятые решения и правильность всех данных несет обучающийся – автор курсовой работы.

По требованию руководителя курсового проектирования обучающийся обязан предоставить на *промежуточную проверку* отдельные части курсовой работы – разделы, фрагменты программ и др. Выявленные руководителем в ходе проверки ошибки и недостатки должны быть письменно обозначены в работе и доведены до сведения обучающегося.

Замечания по содержанию или оформлению выполненной курсовой работы руководитель письменно отмечает в тексте расчетно-пояснительной записки и в графических материалах.

Выполненную курсовую работу обучающийся сдает на *окончательную проверку* руководителю курсового проектирования в соответствии с графиком выполнения курсовой работы, но не позднее, чем *за две недели до начала сессии*.

Срок проверки руководителем курсовой работы, как правило, не должен превышать *пяти рабочих дней*.

Выполненная в полном объеме и допущенная к защите курсовая работа предоставляется на защиту в папке-скоросшивателе или в переплетенном виде.

Защита курсовой работы проводится с целью проверки способности обучающегося самостоятельно решать учебную, исследовательскую или конструкторско-технологическую задачу.

Допуск к защите курсовой работы осуществляется руководителем курсового проектирования после проверки полноты представленных материалов, соответствия их заданию и требованиям оформления, о чем делается соответствующая пометка на титульном листе курсовой работы.

Защита курсовых работ производится перед комиссией по защите курсовых работ, которая формируется заведующим кафедрой в составе двух человек с участием руководителя курсовой работы.

Защита курсовой работы проводится в форме доклада и/или ответов на вопросы, заданные членами комиссии по тематике курсового проектирования.

Результаты защиты курсовой работы оцениваются отметками в баллах по десятибалльной шкале в соответствии с разработанными кафедрой критериям

оценки.

Если работа оценивается неудовлетворительной отметкой, после устранения замечаний, допускается к повторной защите. Как исключение, обучающийся выполняет другую курсовую работу на новую тему.

Пересдача неудовлетворительной отметки, полученной при защите курсовой оценки после неявки обучающегося в установленные сроки без уважительной причины, допускается не более двух раз.

Для второй пересдачи курсовой работы деканом факультета на основании докладной записки заведующего кафедрой назначается комиссия в составе не менее трех человек и устанавливается время ее работы, с которым должен быть ознакомлен обучающийся.

Отметка, выставленная комиссией, является окончательной.

3 ПОСТАНОВКА ЗАДАЧИ НА ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ

В качестве исходного требования к курсовой работе указывается одна из нижеперечисленных динамических структур, для управления которой студент должен реализовать класс:

- линейный односвязный список;
- циклический односвязный список;
- линейный двусвязный список;
- циклический двусвязный список;
- стек;
- очередь;
- дек;
- бинарное поисковое дерево;
- бинарное непоисковое дерево;
- небинарное дерево.

С использованием указанной динамической структуры необходимо решить указанную в листе задания задачу.

Общие требования к разрабатываемому приложению:

- язык программирования C++;
- исходные данные хранятся в текстовом файле;
- описание и реализация класса помещаются в отдельные файлы;
- при реализации класса не использовать контейнеры STL и массивы;
- результаты должны выводиться в файл и в консоль.

Пример задания:

Разработать класс, реализующий динамическую структуру: линейный односвязный список.

С использованием разработанного класса решить следующую задачу: частотный анализ текста, т. е. определение всех слов, встречающихся в тексте и их количества. Регистр не учитывать.

Структура исходного файла: файл содержит анализируемый текст. В тексте могут встречаться помимо букв символы: «,», «.», «:», «;», «)», «(», «!», «?», «'», а также пробел и переход на новую строку.

Вывести список найденных слов, упорядоченный в алфавитном порядке. Каждая строка должна содержать слово и количество его повторений в тексте.

4 СТРУКТУРА И СОДЕРЖАНИЕ КУРСОВОЙ РАБОТЫ

Расчетно-пояснительная записка к курсовой работе, как правило, должна включать:

- титульный лист;
- задание на курсовое проектирование;
- содержание;
- введение;
- основную часть:
 1. Общие сведения о динамической структуре.
 2. Проектирование класса динамической структуры.
 3. Реализация класса динамической структуры.
 4. Решение поставленной задачи.
- заключение;
- список использованных источников;
- приложения (при необходимости).

Форма *титульного листа* курсовой работы приведена в приложении Б.

Во *введении* формулируются цель и задачи курсовой работы.

Основная часть курсового проекта содержит разделы, каждый из которых может состоять из подразделов. Все части курсового проекта должны быть изложены в строгой логической последовательности и взаимосвязи. Каждый раздел имеет определенное целевое назначение и является базой для последующего расширения.

В *заключении* логически и последовательно излагаются результаты разработки и выводы: достигнутые цели, оценка эффективности разработанного решения, его ограничений и возможные направления его развития. Заключение должно быть кратким, четким и сфокусированным на ключевых результатах работы.

В *список использованных источников* нужно включать: учебники и учебные пособия; источники, описывающие синтаксис языка С++ и работу с динамическими структурами данных; официальные интернет-ресурсы средств и инструментов разработки; ресурсы на сайтах сообществ разработчиков с

высоким уровнем доверия и т. п.

Приложения должны включать вспомогательный или дополнительный материал, который загромождает текст основной части работы, но необходим для полноты ее восприятия и оценки практической значимости (изображения графических элементов, диаграммы, таблицы, графики и т. д.).

4.1 Общие сведения о динамической структуре

В данном разделе приводятся краткие теоретические сведения о заданной динамической структуре.

Необходимо:

1. Дать определение заданной динамической структуры.
2. Описать ключевые особенности динамической структуры и графически представить структуру её размещения в памяти.
3. Описать способы реализации динамической структуры (например, для стека необходимо описать наличие варианта реализации с использованием массива и с использованием динамического списка).
4. Перечислить основные и дополнительные операции над динамической структурой.
5. Указать преимущества и недостатки использования динамической структуры для решения прикладных задач в сравнении с другими динамическими структурами.

В заключении необходимо сделать заключение о возможности решения индивидуальной задачи с использованием данной структуры.

Пример:

Линейный связный список – это структура данных, состоящая из элементов одного типа, связанных между собой последовательно посредством указателей.

В такой структуре все элементы линейно упорядочены, но порядок определяется не номерами, как в массиве, а указателями, входящими в состав элементов списка.

Каждый элемент односвязного списка состоит из двух областей памяти: поля данных (информационное поле) D и ссылок (адресное поле) P.



Рисунок 4.1 – Элемент односвязного списка

Поле данных – область памяти, в которой содержатся те данные, ради которых и создается структура.

Полей данных может быть несколько. Каждый элемент списка содержит поле данных, называемое *ключ*, который однозначно идентифицирует этот элемент. Ключ обычно бывает либо целым числом, либо строкой.

Поле ссылок – область памяти, в которой содержится один указатель, связывающий данный элемент со следующим. Последний элемент списка указывает на NULL.

Каждый односвязный список имеет особый элемент – указатель на начало списка, который называется *головой* списка (Head).

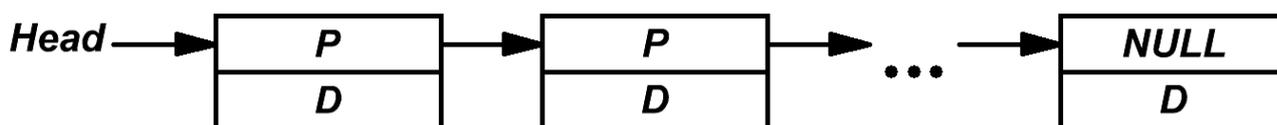


Рисунок 4.2 – Односвязный список

В односвязном списке можно передвигаться только в сторону конца списка. Узнать адрес предыдущего элемента, опираясь на содержимое текущего узла, невозможно.

Реализовать узел односвязного списка можно с помощью класса или структуры. Также можно создавать односвязный список на основе шаблонного класса, оперирующего некоторым обобщенным типом.

Основными операциями, осуществляемыми с однонаправленными списками, являются:

- создание списка;
- печать (просмотр) списка;
- вставка элемента в начало списка;
- вставка элемента в конец списка;
- вставка элемента в указанную позицию в списке;
- удаление элемента из списка;
- поиск элемента в списке;
- проверка наличия элементов в списке;
- удаление списка.

Могут реализовываться и дополнительные операции (например, сортировка списка, перестановка элементов).

Основные преимущества односвязного списка:

- Длина списка ограничивается лишь возможностями памяти компьютера.

- Односвязный список может увеличиваться и уменьшаться во время выполнения программы за счёт выделения и освобождения памяти. Поэтому нет необходимости предварительно выделять память и задавать начальный размер списка.

- Элементы в списке не хранятся в смежных ячейках памяти, в отличие от массива.

- Операции вставки и удаления в связанном списке намного проще. После вставки или удаления элемента нет необходимости сдвигать элементы, нужно только обновить адрес, указанный в следующем указателе.

Основные недостатки односвязного списка определяются их основным свойством, а именно последовательным доступом к данным:

- Для односвязного списка требуется больше памяти, чем для массива. Потому что в списке для хранения адреса следующего элемента требуется указатель, который требует дополнительной памяти для себя.

- В односвязном списке нельзя мгновенно обратиться к любому элементу по индексу. Вместо этого, необходимо последовательно перебирать элементы.

- В односвязном списке, в отличие от двусвязного, невозможен обход элементов в двух направлениях.

- Реализация операций с односвязным списком более сложна по сравнению с операциями над массивом.

4.2 Проектирование класса динамической структуры

Обязательные требования:

1. Запрещается использовать массивы и другие готовые к использованию структуры хранения данных (например, контейнеры STL).

2. Реализовать не менее 2 классов (класс узла динамической структуры и класс самой структуры).

3. Для каждого класса реализовать как минимум 2 конструктора: параметрический и копирования.

4. Разработать все основные методы для реализации основных операций работы с динамической структурой.

5. Разработать 3-4 дополнительных метода, требующих перебора всех элементов структуры (например, метод поиска, или метод копирования части исходной структуры в новую динамическую структуру).

6. Разработать дополнительные методы, необходимые для решения поставленной задачи.

В разделе приводится описание разрабатываемых классов: перечисляются поля и заголовки методов разрабатываемого класса с описанием параметров и возвращаемых значений.

Пример:

Элемент списка представляет собой класс, содержащий три приватных поля:

- `word` – слово (ключ),
- `count` – количество его упоминаний в тексте,
- `next` – указатель на следующий элемент в списке.

Объявим класс списка `List` как дружественный, чтобы из методов класса `List` иметь доступ к скрытым полям класса `Node`.

Также для элемента списка предусмотрим методы:

- Параметрический конструктор. Параметр `newWord` – строка, представляющая слово, которое будет сохранено в поле `word` класса `Node`.
- Метод вывода информации о слове. Метод будет выводить все элементы списка в консоль, поэтому не будет иметь ни параметров, ни возвращаемого значения.

Описание класса элемента списка `Node` представлен в листинге 4.1.

Листинг 4.1 – Класс `Node`

```
class Node
{
private:
    std::string word;
    int count;
    Node* next;

    friend class List;

public:
    Node(std::string newWord);
    void viewInfo();
};
```

4.3 Реализация класса динамической структуры

В данном разделе приводится код всех методов классов с пояснениями.

Пример:

Реализуем конструктор класса элемента списка `Node` (листинг 4.2). Конструктор принимает один строковый параметр – слово `newWord`, значение которого присваивается полю `word` элемента. Полю `count` присваиваем значение `1`, т. к. элемент создается, когда слово встречается в тексте в первый раз. Полю-указателю на следующий элемент присваиваем значение `NULL`, т. к. элемент будет добавлен в список позднее.

Листинг 4.2 – Конструктор класса `Node`

```
Node::Node(std::string newWord)
{
    word = newWord;
    count = 1;
    next = NULL;
}
```

Для пояснения работы методов, оперирующих указателями, можно воспользоваться поясняющими схемами.

Пример:

Схема изменения связей при вставке элемента в начало списка

представлена на рисунке 4.3, а код, реализующий данный метод, – в листинге 4.3.

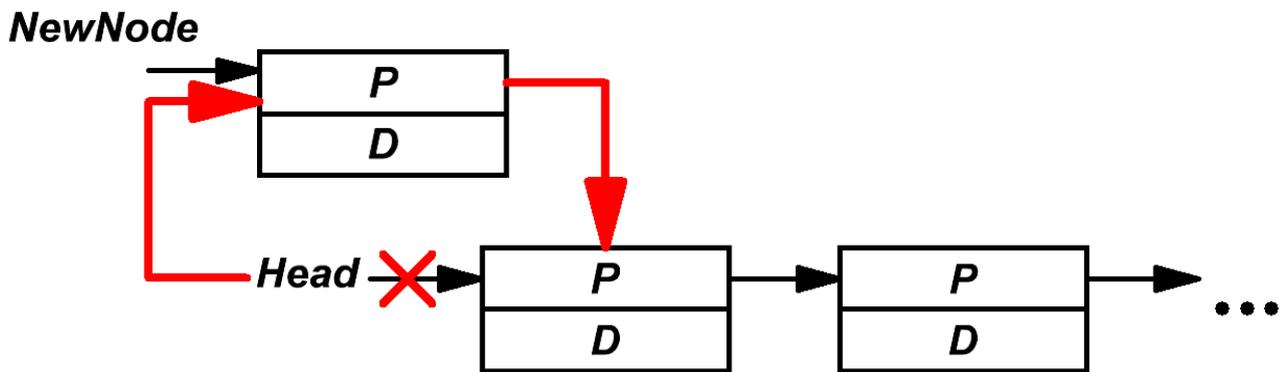


Рисунок 4.3 – Добавление элемента в начало списка

Листинг 4.3 – Добавление элемента в начало списка

```
void List::addFirst(std::string word)
{
    Node* newNode = new Node(word);
    if (isEmpty())
        head = newNode;
    else
    {
        newNode->next = head;
        head = newNode;
    }
    countNodes++;
}
```

Для 2-3 дополнительных (реализующих перебор элементов) методов класса необходимо привести блок-схемы их алгоритмов.

Пример:

Для определения ширины поля вывода слова, необходимо определить максимальную длину слова. Для этого реализуем метод `maxLength()`, блок-схема алгоритма которого показана на рисунке 4.4. Листинг кода, реализующего данный метод, представлен в листинге 4.4.

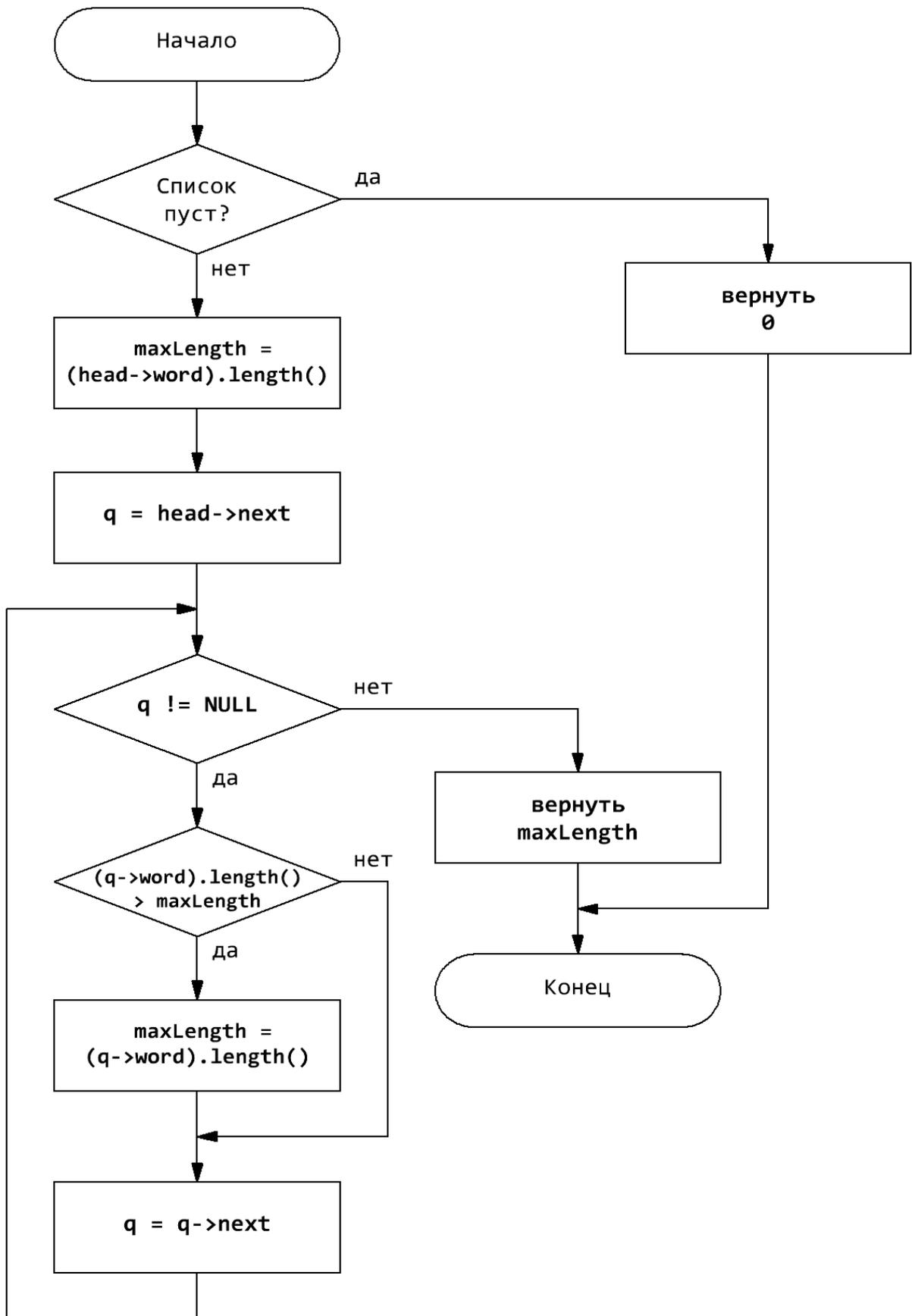


Рисунок 4.4 – Алгоритм поиска максимальной длины слова

Листинг 4.4 – Поиск максимальной длины слова

```
int List::maxLength()
{
    if (isEmpty())
        return 0;
    int maxLength = (head->word).length();
    Node* q = head->next;
    while (q != NULL)
    {
        if ((q->word).length() > maxLength)
            maxLength = (q->word).length();
        q = q->next;
    }
    return maxLength;
}
```

4.4 Решение поставленной задачи

В данном разделе приводиться код основной программы.

Обязательные требования к основной программе:

1. Необходимо продемонстрировать выполнение всех разработанных для класса динамической структуры методов. В том числе тех, которые не используются для решения индивидуальной задачи.

2. Для демонстрации работы каждого метода необходимо вывести в консоль исходное состояние динамической структуры, после вызова метода необходимо вывести на консоль результаты его работы (возвращаемые методом значения, или изменившееся состояние динамической структуры). Например, при демонстрации работы метода добавления в середину списка необходимо до и после его выполнения вывести в консоль все элементы списка.

Для каждого фрагмента основной программы, который демонстрирует работу определённого метода необходимо:

1. Привести код фрагмента с пояснениями.
2. Скриншот консоли, с выводом исходного состояния динамической структуры или входных данных.
3. Скриншот консоли, с выводом конечного состояния динамической структуры и/или полученные результаты.

Аналогичным образом описывается решение основной задачи. Кроме того, для основной задачи необходимо продемонстрировать поиск решения при различных наборах исходных данных, включая граничные наборы данных, например, пустые файлы с входными данными, или файлы, содержащие неверный формат данных.

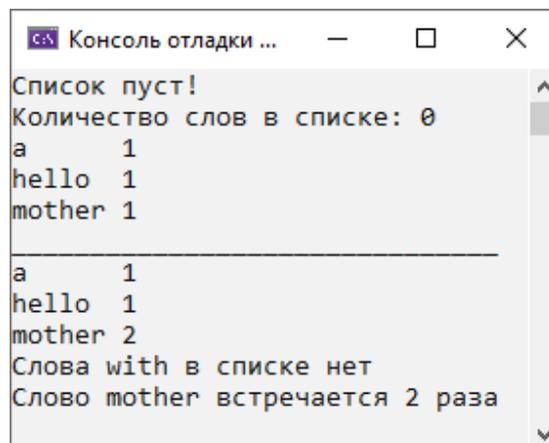
Пример:

Приведем фрагмент кода, демонстрирующий работу разработанных методов просмотра списка, вставки элемента в список, подсчета количества записей и поиска слова в списке (листинг 4.5). Скриншот вывода результатов

выполнения в консоль приведен на рисунке 4.5.

Листинг 4.5 – Код, реализующий демонстрацию работы методов класса

```
// создание экземпляра класса списка (используется конструктор по умолчанию)
List l;
// просмотр списка
l.view();
// вывод количества элементов в списке
cout << "Количество слов в списке: " << l.getCountNodes() << endl;
// добавим в список несколько слов
l.insert("a");
l.insert("mother");
l.insert("hello");
// просмотр списка
cout << "_____ " << endl;
l.view();
// попытаемся добавить слово mother во второй раз
l.insert("mother");
// просмотр списка
l.view();
// ищем несуществующее слово
l.findWord("with");
// ищем существующее слово, записанное с большой буквы
l.findWord("Mother");
```



```
Консоль отладки ...
Список пуст!
Количество слов в списке: 0
a      1
hello  1
mother 1

-----
a      1
hello  1
mother 2
Слова with в списке нет
Слово mother встречается 2 раза
```

Рисунок 4.5 – Консоль с демонстрацией работы методов класса

Аналогичным образом демонстрируется работа остальных открытых методов класса.

Пример:

Задача: частотный анализ текста без учета словоформ и регистра.

Входной файл содержит анализируемый текст. В тексте могут встречаться помимо букв символы: «,», «.», «:», «;», «)», «(», «!», «?», «'», а

также пробел и переход на новую строку.

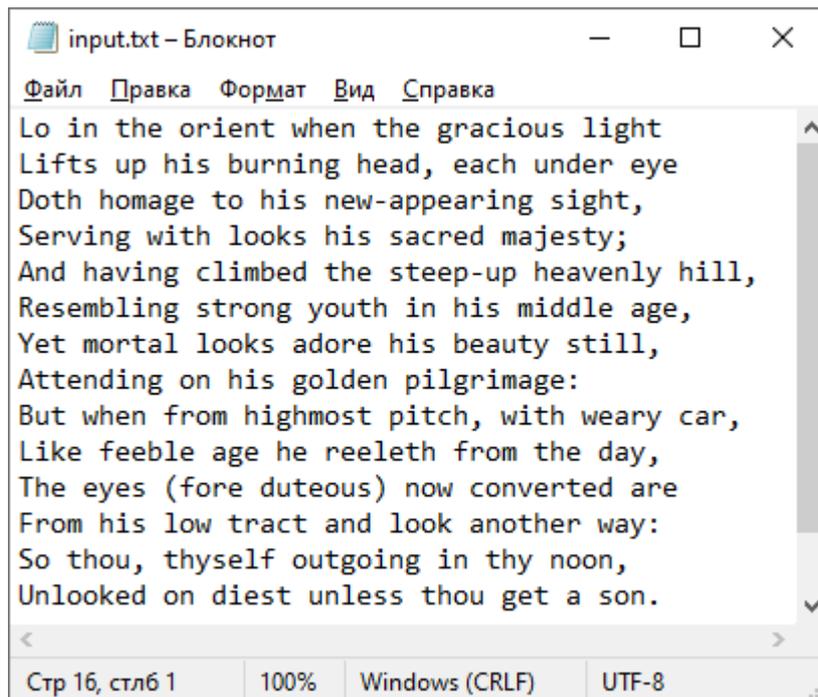


Рисунок 4.6 – Пример исходного файла

Выходной файл должен содержать список, упорядоченный в алфавитном порядке. Для каждого узла в отдельной строке выводится слово и количество появлений в тексте. Следует обратить внимание на форматирование текста в файле: слова и количество его появлений образуют 2 столбца с выравниванием по левому краю. Ширина первого столбца определяется максимальной длиной слова.

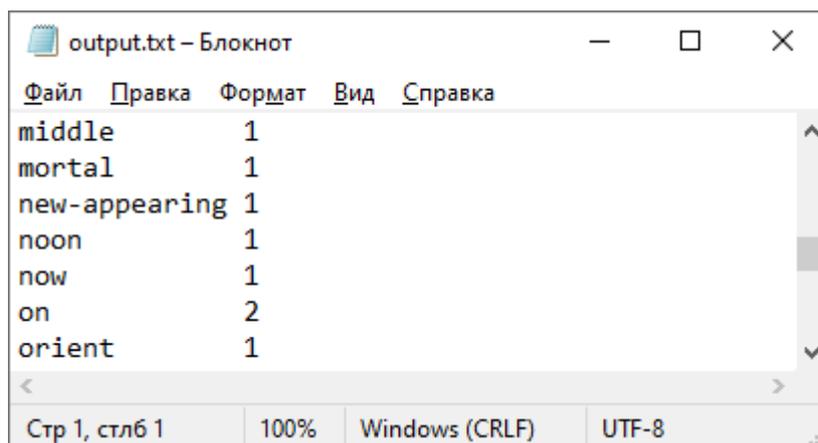


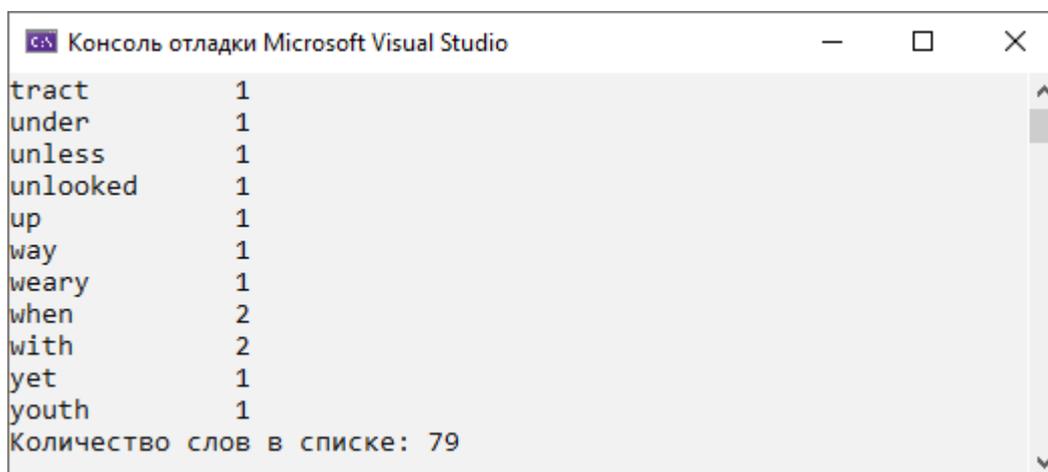
Рисунок 4.7 – Пример фрагмента выходного файла

Далее (листинг 4.6) приводится фрагмент кода основной программы, решающей основную задачу, с пояснениями каждой строки кода. Исходный и результирующий файл были представлены на рисунках 4.6 и 4.7

соответственно, скриншот вывода результатов выполнения в консоль – на рисунке 4.8.

Листинг 4.6 – Код, реализующий решение поставленной задачи

```
// создание экземпляра класса списка (используется конструктор по умолчанию)
List l;
// добавление слов в список из файла
l.addFromFile("input.txt");
l.view(); // просмотр полученного списка
// вывод в консоль количества элементов в списке
cout << "Количество слов в списке: " << l.getCountNodes() << endl;
l.writeToFile("output.txt"); // запись списка в файл
```



```
Консоль отладки Microsoft Visual Studio
tract      1
under     1
unless    1
unlooked  1
up        1
way       1
weary     1
when      2
with      2
yet       1
youth     1
Количество слов в списке: 79
```

Рисунок 4.8 – Консоль с представленным решением задачи

5 ОФОРМЛЕНИЕ РАСЧЕТНО-ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ

Расчетно-пояснительная записка курсового проекта выполняется в соответствии с ЕСКД и с соблюдением требований на документацию, чертежи, планы и схемы, принятые в сфере информационных технологий и автоматизации.

Требования к оформлению **текста** расчетно-пояснительной записки:

- шрифт – Times New Roman 14 пт, обычный, нормальный, черный;
- межстрочный интервал – 1,5;
- красная строка – 1,25 см;
- отступы – 0;
- интервалы между абзацами – 0;
- поля: правое – не менее 10 мм, верхнее и нижнее – не менее 20 мм, левое – не менее 30 мм;
- нумерация страниц – сквозная;

- заголовки 1-го уровня (разделов) – шрифт – Times New Roman 14 пт, все прописные, жирный, нормальный, черный, выравнивание по ширине с красной строкой 1,25 см;

- заголовки 2-го уровня (подразделов) – шрифт – Times New Roman 14 пт, как в предложениях, жирный, нормальный, черный, выравнивание по ширине с красной строкой 1,25 см;

- заголовки 3-го и 4-го уровней (пунктов и подпунктов) – шрифт – Times New Roman 14 пт, как в предложениях, обычный, нормальный, черный, выравнивание по ширине с красной строкой 1,25 см.

Заголовки всех уровней записываются без точки в конце, не подчеркивая. Переносы в словах в заголовках не допускаются. Если заголовок состоит из двух предложений, их разделяют точкой.

Между заголовком и текстом и между заголовками раздела и подраздела – пустая строка.

Разделы должны иметь порядковые номера в пределах всей расчетно-пояснительной записки, обозначенные арабскими цифрами без точки.

Подразделы должны иметь нумерацию в пределах каждого раздела. Номер подраздела состоит из номеров раздела и подраздела, разделенных точкой. В конце номера подраздела точка не ставится.

Подразделы могут состоять из одного или нескольких пунктов. Нумерация пунктов должна быть в пределах подраздела, и номер пункта должен состоять из номеров раздела, подраздела и пункта, разделенных точками.

Пункты, при необходимости, могут быть разбиты на подпункты, которые должны иметь порядковую нумерацию в пределах каждого пункта.

В **тексте** расчетно-пояснительной записки не допускается:

- применять обороты разговорной речи, техницизмы, профессионализмы;
- применять для одного и того же понятия различные научно-технические термины, близкие по смыслу (синонимы), а также иностранные слова и термины при наличии равнозначных слов и терминов в русском языке;

- применять произвольные словообразования;

- применять сокращения слов, кроме установленных правилами русской орфографии, соответствующими государственным стандартам, а также в данном документе;

- сокращать обозначения единиц физических величин, если они употребляются без цифр, за исключением единиц физических величин в головках и боковиках таблиц, и в расшифровках буквенных обозначений, входящих в формулы и рисунки.

Если в документе приводятся поясняющие надписи, наносимые непосредственно на изготавливаемое изделие (например, на планки, таблички к элементам управления и т. п.), их выделяют шрифтом (без кавычек), например, ВКЛ., ОТКЛ., или кавычками – если надпись состоит из цифр и/или знаков.

Наименования команд, режимов, сигналов и т. п. в тексте следует выделять кавычками, например, «Сигнал +27 включено».

В тексте документа числовые значения величин с обозначением единиц

физических величин и единиц счета следует писать цифрами, а числа без обозначения единиц физических величин и единиц счета от единицы до девяти – словами.

Недопустимо отделять единицу физической величины от числового значения (переносить их на разные строки или страницы).

Дробные числа необходимо приводить в виде десятичных дробей, за исключением размеров в дюймах, которые следует записывать 1/4"; 1/2" (но не $\frac{1}{4}$, $\frac{1}{2}$).

При невозможности выразить числовое значение в виде десятичной дроби, допускается записывать в виде простой дроби в одну строчку через косую черту, например, 5/32; (50А-4С)/(40В+20).

В тексте расчетно-пояснительной записки могут быть приведены **перечисления**.

Перед каждым элементом перечисления следует ставить тире. При необходимости ссылки в тексте расчетно-пояснительной записки на один из элементов перечисления вместо тире ставят строчные буквы русского алфавита со скобкой, начиная с буквы «а» (за исключением букв ё, з, й, о, ч, ь, ы, ь). Простые перечисления отделяются запятой, сложные – точкой с запятой. При наличии конкретного числа перечислений допускается перед каждым элементом перечисления ставить арабские цифры, после которых ставится скобка. Перечисления приводятся с абзацного отступа в столбик.

Пример 1. Информационно-сервисная служба для обслуживания удаленных пользователей включает следующие модули:

- удаленный заказ,
- виртуальная справочная служба,
- виртуальный читальный зал.

Пример 2. Работа по оцифровке включала следующие технологические этапы:

- а) первичный осмотр и структурирование исходных материалов;
- б) сканирование документов;
- в) обработка и проверка полученных образов;
- г) структурирование оцифрованного массива;
- д) выходной контроль качества массивов графических образов.

Пример 3. Разрабатываемое сверхмощное устройство можно будет применять в различных отраслях реального сектора экономики:

- в машиностроении:
 - 1) для очистки отливок от формовочной смеси;
 - 2) для очистки лопаток турбин авиационных двигателей;
 - 3) для холодной штамповки из листа;
- в ремонте техники:
 - 1) устранение наслоений на внутренних стенках труб;
 - 2) очистка каналов и отверстий небольшого диаметра от грязи.

Рисунки подписываются снизу посередине. Перед рисунком и после его названия вставляется пустая строка. Выравнивание рисунка и его названия – по центру без красной строки.

Нумеруются рисунки в пределах раздела. Номер состоит из номера раздела и порядкового номера рисунка в разделе, разделенных точкой. Номер записывается после слова «Рисунок», после номера ставится тире и с прописной буквы записывается название рисунка. Например:

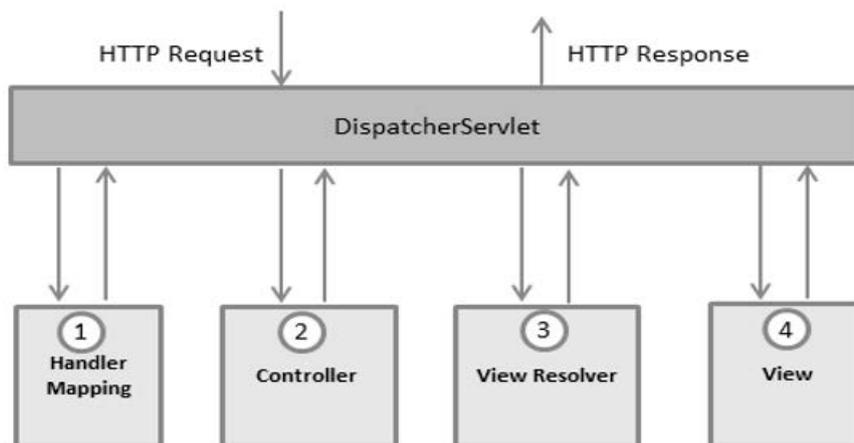


Рисунок 5.1 – Обработка запроса Spring MVC

Если рисунок состоит из нескольких частей, то они нумеруются буквами русского алфавита. Название каждой из частей приводится после названия рисунка. Например:

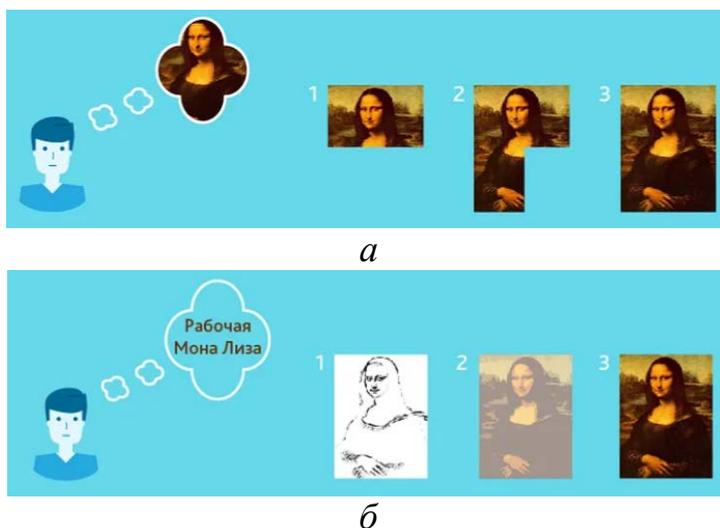


Рисунок 5.2 – Иллюстрация моделей жизненного цикла ПО:
а – инкрементная модель; *б* – итеративная модель

Если на рисунке имеются буквенные или цифровые обозначения, то после названия рисунка ставится двоеточие и далее с новой строки идет расшифровка данных обозначений. Например:

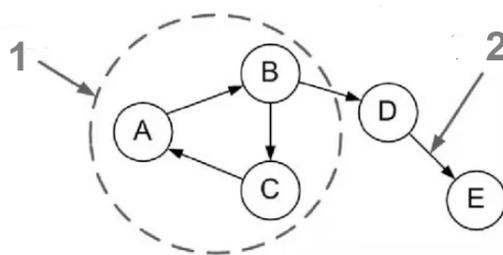


Рисунок 5.3 – Схема ориентированного графа:
1 – цикл; 2 – ребро

На рисунки могут быть приведены ссылки в тексте документа. При ссылке следует писать слово «рисунок» с указанием его номера. Ссылки оформляются следующим образом, например, «...приведен на рисунке 5.3» либо (рис. 5.3).

Рисунок, в зависимости от его размера, помещают под текстом, в котором впервые дана ссылка на него, или на следующей странице.

Таблицы подписываются сверху с красной строки, выравнивание – по ширине. После таблицы и перед ее подписью вставляется пустая строка.

Заголовки граф и строк таблицы следует писать с прописной буквы. В конце заголовков и подзаголовков таблиц точки не ставят.

Шрифт в таблицах может быть уменьшен до 10 пт.

Таблицы, за исключением таблиц приложений, следует нумеровать арабскими цифрами с нумерацией в пределах раздела. Номер таблицы состоит из номера раздела и порядкового номера таблицы в разделе, разделенных точкой. Номер записывается после слова «Таблица», после номера ставится тире и с прописной буквы записывается название.

Если в конце страницы таблица прерывается и ее продолжение будет на следующей странице, в первой части таблицы нижнюю горизонтальную линию, ограничивающую таблицу, не проводят. При этом столбцы таблицы нумеруются под шапкой, а следующую часть таблицы начинают со строки с номерами столбцов. Если в конце страницы остается только шапка таблицы, то на следующую страницу переносят всю таблицу с названием.

Название таблицы указывают один раз над первой частью таблицы, над следующими частями пишут слова «Продолжение таблицы» с указанием номера таблицы. Над последней частью пишут слова «Окончание таблицы» с указанием номера таблицы. Например:

Таблица 5.1 – Сравнение программ

Фирма	Стоимость разработки или внедрения	Доп. раб. места	Задачи, решаемые системой
1	2	3	4
1С Битрикс24	Стоимость программы для использования на пяти компьютерах – 29 990 руб.	900 руб.	Постановка задачи, быстрый переход к важным задачам в работе, готовые отчеты по задачам, учет ресурсов, управление проектами и т. д.

Окончание таблицы 5.1

1	2	3	4
Мегаплан	Стоимость программы для использования на пяти компьютерах – 35 640 руб.	900 руб.	Группирует и объединяет проекты, сортирует дела, готовые отчеты по заданиям, удобный каталог сделок для менеджера продаж и т. д.
Фирма «1С»	Стоимость программы для использования на пяти компьютерах – 31 800 руб.	500 руб.	Учет заказов, заказ-нарядов, состояний заказов, ведение базы нормативных временных затрат на выполнение работ, учет выполненных работ, учет прямых и косвенных затрат, расчет фактической себестоимости, контроль выполнения заданий сотрудниками, выявление отклонений от запланированных графиков и т. д.

На таблицы могут быть приведены ссылки в тексте документа. При ссылке следует писать слово «таблица» с указанием ее номера. Ссылки оформляются следующим образом, например, «...приведены в таблице 3.1» или (таблица 5.1).

Таблицу, в зависимости от ее размера, помещают под текстом, в котором впервые дана ссылка на нее, или на следующей странице.

В основной части расчетно-пояснительной записки могут приводиться *Листинги* фрагментов кода, которые помещают под текстом, в котором они впервые упоминаются.

Для листингов необходимо использовать моноширинные шрифты (например, Courier New, Consolas), размер 9 – 12 пт. Желательно, чтобы строки кода не переносились на новую строку.

Листинги, за исключением листингов приложений, следует нумеровать арабскими цифрами с нумерацией в пределах раздела. Номер листинга состоит из номера раздела и порядкового номера листинга в разделе, разделенных точкой. Номер записывается после слова «Листинг», после номера ставится тире и с прописной буквы записывается название. Листинги подписываются сверху с красной строки, выравнивание – по ширине. После листинга и перед его подписью вставляется пустая строка. Например:

Листинг 5.1 – Приложение «Hello world!»

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello world!" << endl;
    return 0;
}
```

При ссылке на листинг следует писать слово «листинг» с указанием его номера.

Формулы выравниваются посередине и нумеруются по правому краю с помощью табуляции, причем номер записывается в круглых скобках. Нумерация формул в пределах раздела. Номер формулы состоит из номера раздела и порядкового номера формулы в разделе, разделенных точкой. От текста строка с формулой отделяется пустыми строками.

Ссылки в тексте на порядковые номера формул дают в скобках, например, «... в формуле (5.1)...».

Пояснения символов и числовых коэффициентов, входящих в формулу, если они не пояснены ранее в тексте, должны быть приведены непосредственно под формулой. Пояснения каждого символа следует давать с новой строки в той последовательности, в которой символы приведены в формуле. Первая строка пояснения должна начинаться со слова «где» без двоеточия после него.

Переносить формулы на следующую строку допускается только на знаках выполняемых операций, причем знак в начале следующей строки повторяют. При переносе формулы на знаке умножения применяют знак «х».

Пример:

$$q = \frac{\lambda_3}{\delta_3} (t_{31} - t_{32}), \quad (5.1)$$

где λ_3 – коэффициент теплопроводности эталона;

δ_3 – толщина эталона;

t_{31} и t_{32} – температуры наружных поверхностей эталона.

Список использованных источников оформляется в соответствии со следующими стандартами:

1. ГОСТ 7.1-2003 «Библиографическая запись. Библиографическое описание. Общие требования и правила составления».

2. ГОСТ 7.80-2000 «Библиографическая запись. Заголовок. Общие требования и правила составления».

3. ГОСТ 7.82-2001 «Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления».

4. ГОСТ 7.0.12-2011 «Библиографическая запись. Сокращение слов на русском языке. Общие требования и правила».

5. СТБ 7.12-2001 «Библиографическая запись. Сокращение слов и словосочетаний на белорусском языке».

6. ГОСТ 7.11-2004 (ИСО 832:1994) «Библиографическая запись. Сокращение слов и словосочетаний на иностранных европейских языках».

Источники в списке располагают в порядке появления ссылки на них. Каждый источник нумеруется арабскими цифрами с точкой (например: 1., 2., 3. и т. д.) и начинается с красной строки.

Ссылки на источник в тексте даются в форме записи его порядкового

номера в списке источников, заключенного в квадратные скобки. Например: [26].

Далее представлены примеры записи различных источников.

Книги с одним, двумя или тремя авторами:

1. Шотт, А. В. Курс лекций по частной хирургии / А. В. Шотт, В. А. Шотт. – Минск: Асар, 2004. – 525 с.
2. Чикатуева, Л. А. Маркетинг : учеб. пособие / Л. А. Чикатуева, Н. В. Третьякова; под ред. В. П. Федько. – Ростов на Дону: Феникс, 2004. – 413 с.

Книги с более чем тремя авторами:

3. Энергетические риски в малой открытой экономике : науч.-практ. пособие / А. А. Быков [и др.]. – Минск : Мисанта, 2013. – 107 с.

Сборники:

4. Наноструктуры в конденсированных средах : сб. науч. ст. / НАН Беларуси, Ин-т тепло- и массообмена ; редкол.: П. А. Витязь [и др.]. – Минск : Ин-т тепло- и массообмена, 2013. – 409 с.

Многотомные издания в целом:

5. Багдановіч, М. Поўны збор твораў : у 3 т. / М. Багдановіч. – 2-е выд. – Мінск : Беларус. навука, 2001. – 3 т.

Отдельные тома в многотомном издании:

6. Багдановіч, М. Поўны збор твораў : у 3 т. / М. Багдановіч. – 2-е выд. – Мінск : Беларус. навука, 2001. – Т. 1 : Вершы, паэмы, пераклады, наследаванні, чарнавыя накіды. – 751 с.

Каталоги:

7. Сухие строительные смеси : [пром. кат. / ПТ ООО] Тайфун. – [Б. м. : б. и., б. г.]. – 179 с. : цв. ил.

8. Каталог древесных растений основных коллекционных фондов Беларуси / Центр. ботан. сад НАН Беларуси ; сост.: И. М. Гаранович [и др.] ; науч. ред. В. В. Титок. – Минск : Право и экономика, 2013. – 133 с.

Методические указания:

9. Основы компьютеризации технологий в системах автоматизации. Основы программирования на алгоритмическом языке : методические указания по выполнению лабораторных работ / сост. А. С. Соколова, В. Е. Казаков. – Витебск : УО «ВГТУ», 2021. – 80 с.

Статьи из сборников, справочных изданий:

10. Божанов, П. В. Направления развития транспортного комплекса Беларуси / П. В. Божанов // Современные концепции развития транспорта и логистики в Республике Беларусь : сб. ст. / Ин-т бизнеса и менеджмента

технологий Белорус. гос. ун-та ; сост.: В. В. Апанасович, А. Д. Молокович. – Минск, 2014. – С. 56–64.

11. Аляхновіч, М. М. Электронны мікраскоп / М. М. Аляхновіч // Беларуская энцыклапедыя : у 18 т. / Беларус. Энцыкл. ; рэдкал.: Г. П. Пашкоў [і інш.]. – Мінск, 2004. – Т. 18, кн. 1. – С. 100.

Статья из журнала:

12. Абламейко, С. В. Оперативное совмещение цифровых аэрокосмоснимков и карт по опорным точкам в специализированной геоинформационной системе / С. В. Абламейко, С. П. Боричев // Вестн. Белорус. гос. ун-та. Сер. 1, Физика. Математика. Информатика. – 2010. – № 3. – С. 57–66.

Законы и законодательные материалы, инструкции:

13. Конституция Республики Беларусь : с изм. и доп., принятыми на респ. референдумах 24 нояб. 1996 г. и 17 окт. 2004 г. – Минск : Нац. центр правовой информ. Респ. Беларусь, 2016. – 62 с.

14. О Парке высоких технологий [Электронный ресурс] : Декрет Президента Респ. Беларусь, 22 сент. 2005 г., № 12 : в ред. Декрета Президента Респ. Беларусь от 03.11.2014 г. // ЭТАЛОН. Законодательство Республики Беларусь / Нац. центр правовой информ. Респ. Беларусь. – Минск, 2016.

15. О защите прав потребителей : Закон Респ. Беларусь от 9 янв. 2002 г. № 90-З : в ред. от 8 июля 2008 г. № 366-З : с изм. и доп. от 2 мая 2012 г. № 353-З. – Минск : Амалфея, 2013. – 59 с.

16. Об утверждении основных показателей развития лесного хозяйства на 2014 год [Электронный ресурс] : постановление Совета Министров Респ. Беларусь, 24 дек. 2013 г., № 1124 // ЭТАЛОН. Законодательство Республики Беларусь / Нац. центр правовой информ. Респ. Беларусь. – Минск, 2016.

17. Инструкция по определению энтомологических показателей эпидемического сезона малярии : утв. М-вом здравоохранения Респ. Беларусь 02.05.13. – Минск : Респ. центр гигиены, эпидемиологии и обществ. здоровья, 2013. – 15 с.

Стандарты:

18. Библиографическая запись. Сокращение слов на русском языке : общие требования и правила : ГОСТ 7.0.12–2011. – Москва : Стандартинформ, 2012. – 17с. – (Система стандартов по информации, библиотечному и издательскому делу).

19. Узел крепления крановых рельсов к стальным подкрановым балкам. Технические условия : СТБ 2135-2010. – Введ. 01.07.11 (с отменой на территории РБ ГОСТ 24741-81). – Минск : Белорус. гос. ин-т стандартизации и сертификации, 2011. – 6 с.

Патенты и иные документы об интеллектуальной собственности:

20. Способ определения содержания белка в семенах льна масличного : пат. ВУ 13275 / В. В. Титок, Л. В. Хотылева, Л. В. Корень, С. И. Вакула, В. Н. Леонтьев, Л. М. Шостак. – Опубл. 30.06.2010.

21. Унифицированный тренажер летных экипажей летательных аппаратов : полез. модель RU 14689 / М. С. Данилов, Ю. А. Косарев, В. Ю. Луканичев, М. М. Сильвестров, В. В. Сомик. – Опубл. 10.08.2000.

Ресурсы удаленного доступа:

22. Национальный статистический комитет Республики Беларусь [Электронный ресурс]. – Режим доступа: <http://belstat.gov.by/>. – Дата доступа: 24.06.2016.

23. Ильинский, И. М. Высшее образование для XXI века [Электронный ресурс] / И. М. Ильинский // Знание. Понимание. Умение – 2012. – № 4. – Режим доступа: <http://www.zpu-journal.ru/zpu/contents/2012/4/>. – Дата доступа: 23.05.2013.

24. World heritage list [Electronic resource] // UNESCO, World Heritage Centre. – Mode of access: <http://whc.unesco.org/en/list>. – Date of access: 20.06.2016.

Составные части ресурсов удаленного доступа:

25. О демографической ситуации в январе – июне 2016 г. [Электронный ресурс] // Национальный статистический комитет Республики Беларусь. – Режим доступа: http://www.belstat.gov.by/o-belstate_2/novosti-i-meropriyatiya/novosti/o_demograficheskoy_situatsii_v_yanvare_iyune_2016_g/. – Дата доступа: 29.07.2016.

Приложения оформляют в конце расчетно-пояснительной записки, располагая их в порядке появления ссылок в тексте.

Ссылки оформляются следующим образом, например, «...приведен в приложении А» либо (приложение А).

Приложения обозначают заглавными буквами русского алфавита, начиная с А (за исключением букв Ё, З, Й, О, Ч, Ъ, Ы, Ь). Допускается обозначать приложения буквами латинского алфавита, за исключением букв I и O.

Таблицы и рисунки каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначения приложения, например: «Рисунок А.2», «Таблица А.1», если они приведены в приложении А.

Приложения, как правило, выполняют на листах формата А4, А3, А2, А1 и А0.

Все приложения должны быть перечислены в содержании документа (при наличии) с указанием их номеров и заголовков.

Все приложения должны иметь общую с остальной частью расчетно-пояснительной записки сквозную нумерацию страниц.

ЛИТЕРАТУРА

1. Баранова, И. В. Программирование на языке С++ : учебное пособие / И. В. Баранова. – Красноярск : СФУ, 2022. – 292 с.
2. Керниган, Б. У. Практика программирования / Б. У. Керниган, Р. Пайк. – Пер. с англ. – М. : Издательский дом «Вильямс», 2004. – 288 с.
3. Навроцкий, А. А. Основы алгоритмизации и программирования в среде Visual С++ : учеб.-метод. пособие / А. А. Навроцкий. Минск : БГУИР, 2014. – 160 с.
4. Панкратов, А. С. Технология программирования на языке С++. Динамические структуры, объекты, классы : учебное пособие / А. С. Панкратов, С. И. Салпагаров. – Москва : Российский университет дружбы народов, 2021. – 70 с.
5. Рейзлин, В. И. Язык С++ и программирование на нём : учебное пособие / В. И. Рейзлин ; Томский политехнический университет. – 3-е изд., перераб. – Томск : Изд-во Томского политехнического университета, 2021. – 208 с.
6. Русакова, З. Н. Структуры данных в С++ : учебное пособие / З. Н. Русакова, И. В. Рудаков. – Москва : МГТУ им. Баумана, 2020. – 158 с.
7. Седжвик, Р. Фундаментальные алгоритмы на С++. Анализ/Структуры данных/Сортировка/Поиск / Р. Седжвик. – Пер. с англ. – К. : Издательство «ДиаСофт», 2001. – 668 с.

ПРИЛОЖЕНИЕ А
ПРИМЕР ЛИСТА ЗАДАНИЯ НА КУРСОВУЮ РАБОТУ

Учреждение образования
«Витебский государственный технологический университет»
Факультет информационных технологий и робототехники
Кафедра «Информационные системы и технологии»

УТВЕРЖДАЮ
заведующий кафедрой ИСиТ
_____ Казаков В.Е.
« ____ » _____ 2025 г.

ЗАДАНИЕ
на курсовое проектирование
по дисциплине «Основы алгоритмизации и программирования»

Студенту _____ Иванову Ивану Ивановичу _____ группы _____ Ит-13

1. Тема курсового проекта Решение задач с использованием динамических структур данных

2. Сроки сдачи курсового проекта 26.05.2025 г.

3. Исходные данные к курсовому проекту _____
Разработать класс, реализующий динамическую структуру: линейный односвязный список

С использованием разработанного класса решить следующую задачу:
Частотный анализ текста, т. е. определение всех слов, встречающихся в тексте и их количества. Регистр и словоформы не учитывать.

Структура исходного файла: файл содержит анализируемый текст. В тексте могут встречаться помимо букв символы: «,», «.», «:», «;», «)», «(», «!», «?», «'», а также пробел и переход на новую строку.

Вывести: список, упорядоченный в алфавитном порядке. Для каждого узла в отдельной строке выводить слово и количество повторений в тексте.

4. Содержание расчетно-пояснительной записки

Введение

1. Общие сведения о динамической структуре

2. Проектирование класса динамической структуры

3. Реализация класса динамической структуры

4. Решение поставленной задачи

Заключение

5. Характеристика графического материала и/или презентации: _____

6. Руководитель курсового проектирования:

старший преподаватель каф. ИСиТ Соколова А.С.

7. Календарный график курсового проектирования

№	Содержание этапа работы	Сроки выполнения	Подпись или замечания руководителя
1	Введение. Общие сведения о динамической структуре	24.02.2025 – 09.03.2025	
2	Проектирование класса динамической структуры	10.03.2025 – 30.03.2025	
3	Реализация класса динамической структуры	31.03.2025 – 20.04.2025	
4	Решение поставленной задачи. Заключение	21.04.2025 – 18.05.2025	
5	Оформление курсовой работы	19.05.2025 – 25.05.2025	
6	Предоставление на окончательную проверку курсовой работы руководителю	26.05.2025	
7	Защита курсовой работы	02.06.2025 – 07.06.2025	

Руководитель

А. С. Соколова

Задание принял к исполнению «_____» _____ 2025 г. _____
подпись

Проинформирован об обязанности выполнять курсовую работу самостоятельно, без привлечения третьих лиц, не допускать плагиата, некорректных заимствований, фальсификации и подлога материалов

подпись

ПРИЛОЖЕНИЕ Б
ПРИМЕР ТИТУЛЬНОГО ЛИСТА КУРСОВОЙ РАБОТЫ

Министерство образования Республики Беларусь
Учреждение образования
«Витебский государственный технологический университет»

Кафедра «Информационные системы и технологии»

КУРСОВАЯ РАБОТА

по дисциплине «Основы алгоритмизации и программирования»
на тему «Решение задач с использованием динамических структур данных»

Выполнил:

студент факультета информационных
технологий и робототехники
группы ИТ-13

подпись

И. И. Иванов

Проверил:

руководитель
ст. преп. каф. ИСиТ

подпись

А. С. Соколова

отметка о допуске к защите

«_____» _____ 20__ г. _____

дата *подпись руководителя*

Витебск 2025

Учебное издание

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

Методические указания по выполнению курсовых работ

Составители:
Соколова Анна Сергеевна
Козаков Вадим Евгеньевич

Редактор *Р.А. Никифорова*
Корректор *А.С. Прокопюк*
Компьютерная верстка *А.С. Соколова*

Подписано к печати 07.04.2025. Усл. печ. листов 1,9.
Уч.-изд. листов 2,5. Заказ № 78.

Учреждение образования «Витебский государственный технологический университет»
210038, г. Витебск, Московский пр., 72.

Отпечатано на ризографе учреждения образования
«Витебский государственный технологический университет».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий № 1/172 от 12 февраля 2014 г.
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий № 3/1497 от 30 мая 2017 г.