

Министерство науки и высшего образования Российской Федерации

САНКТ-ПЕТЕРБУРГСКИЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

Ю.А. Алексеева

Дисциплина: Инженерная графика

Методические указания к практической работе

*Разработка схемы алгоритма программы в соответствии с
ГОСТ 19.701-90 с использованием редакторов диаграмм и
инструментов ИИ*

Санкт-Петербург

2025 г.

Содержание

Задание на практическую работу.....	2
Методические указания.....	3
Построение схемы алгоритма в редакторе диаграммы Dia Diagram Editor.....	4
Генерация схемы алгоритма с помощью ИИ DeepSeek.com.....	6
Визуализации схемы алгоритма программы, представленной в виде текстового описания на языке PlantUML в сервисе PlantUML Web Server.....	8
Выполнение сравнительного анализа использованных инструментов и выводы.....	9
Оформление отчета.....	10
Критерии оценки работы.....	11
Приложение 1. Пример отчета.....	12

Задание на практическую работу

Тема: Разработка схемы алгоритма программы в соответствии с ГОСТ 19.701-90 с использованием редакторов диаграмм и инструментов ИИ

Цель работы:

1. Освоить проектирование схем алгоритмов программ с использованием условных операторов и циклов.
 2. Научиться оформлять схемы алгоритмов в соответствии с требованиями ГОСТ 19.701-90.
 3. Приобрести навыки работы в специализированном ПО (Dia Diagram Editor).
 4. Изучить возможности использования искусственного интеллекта (DeepSeek.com) для автоматизации создания схем алгоритмов.
 5. Приобрести навыки работы с инструментом редактирования и визуализации схем диаграмм, представленных в виде текстового описания на языке PlantUML (PlantUML Web Server).
-

Задание:

1. Выберите программу, написанную на любом языке программирования (например, из практических заданий по дисциплине «Теория и технология программирования»).
2. Схема алгоритма выбранной программы должна отвечать следующим требованиям:
 - Схема должна содержать **не менее 20 блоков**
 - Обязательно должны присутствовать **циклы и условные операторы**
3. Изучите ГОСТ 19.701-90 ЕСПД. «Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения» (переиздание от 2010 года).
4. Оформите схему алгоритма **в соответствии с ГОСТ 19.701-90**.
5. Постройте схему алгоритма в редакторе диаграмм и с помощью инструментов искусственного интеллекта (ИИ) в соответствии с методическими рекомендациями:
 - Часть 1. Построение схемы алгоритма в редакторе диаграммы **Dia Diagram Editor**
 - Часть 2. Генерация блок-схемы с помощью ИИ **DeepSeek.com**
 - Часть 3. Визуализируйте схему алгоритма программы описанной на текстовом языке PlantUML (результат, выданный DeepSeek) с помощью онлайн-сервиса **PlantUML Web Server**.
6. Выполните сравнительный анализ использованных инструментов и напишите выводы.
7. Оформите отчет в соответствии с методическими указаниями.

Методические указания

1. Выберите программу, написанную на любом языке программирования (например, из практических заданий по дисциплине «Теория и технология программирования»).

2. Проверьте, что схема алгоритма выбранной программы отвечает следующим требованиям:

- 2.1. Схема содержит **не менее 20 блоков**.
 - 2.2. В схеме присутствуют **циклы и условные операторы**.
-

3. Изучите ГОСТ 19.701-90 ЕСПД. «Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения», переиздание от 2010 года. Обратите внимание на оформление и описание условных обозначений (символов), которые описывает стандарт, а также на типы схем в документации по обработке данных, на которые распространяется стандарт.

4. Оформите схему алгоритма в соответствии с ГОСТ 19.701-90. При оформлении используйте:

- а) стандартные обозначения:
 - **Овал** – начало/конец алгоритма.
 - **Прямоугольник** – процесс (действие).
 - **Ромб** – условие (ветвление).
 - **Параллелограмм** – ввод/вывод.
 - **Шестиугольник** – цикл (для for/while).
 - б) Стрелки, четко указывающие направление выполнения.
 - в) Подписи для всех блоков.
 - г) Контроль отсутствия «висячих» элементов.
-

5. Постройте схему алгоритма в редакторе диаграмм, с помощью инструментов искусственного интеллекта (ИИ) и в сервисе PlantUML Web Editor в соответствии с методическими рекомендациями. При выполнении заданий **фиксируйте** время создания схем алгоритмов в разных инструментах, количество действий, особенности процесса.

Построение схемы алгоритма в редакторе диаграммы Dia Diagram Editor

Цель задания: освоить создание блок-схем алгоритмов в соответствии с ГОСТ 19.701-90 с использованием редактора Dia Diagram Editor.

Dia Diagram Editor — это свободный кроссплатформенный редактор диаграмм, доступный и для самостоятельной установки. Программа позволяет создавать различные виды диаграмм, включая блок-схемы алгоритмов, UML-диаграммы, ER-диаграммы баз данных, сетевые схемы и другие.

Шаг 1. Запуск программы и создание проекта

1. Откройте Dia Diagram Editor.
2. Создайте новый документ: **Файл** → **Новый** (или используйте горячие клавиши **Ctrl+N**).

Шаг 2. Выбор элементов блок-схемы

1. В левой панели выберите категорию **Flowchart**.
2. Добавьте на рабочую область необходимые элементы:
 - Овал – для обозначения начала и конца алгоритма.
 - Прямоугольник – для выполнения операций (например, вычислений).
 - Ромб – для условий и ветвлений (if/else).
 - Параллелограмм – для ввода/вывода данных.
 - Шестиугольник – для обозначения циклов (опционально).

Шаг 3. Расположение и соединение элементов

1. Перетащите элементы на рабочую область, соблюдая логическую последовательность алгоритма.
2. Соедините элементы с помощью инструмента **Соединительная линия** (на панели инструментов).
 - Используйте **стрелки** для указания направления потока.
 - Для условий из ромба выведите **две линии** (Да/Нет).

Шаг 4. Добавление текста

1. Для подписей линий используйте инструмент **Текст** (на панели инструментов).
2. Щелкните по элементу, чтобы добавить текст.

Шаг 5. Проверка и оформление

1. Убедитесь, что:

- Все блоки соединены, нет «висячих» элементов.
- Схема соответствует логике исходного кода.
- Используются правильные фигуры по ГОСТ 19.701-90.

2. Отформатируйте схему:

- Измените размер блоков при необходимости.
- Настройте шрифт и цвет через **Свойства объекта** (правый клик → Свойства).

Шаг 6. Сохранение результатов

1. Сохраните проект в формате .dia: **Файл** → **Сохранить как**.

2. Экпортируйте схему для отчета: **Файл** → **Экспорт** → выберите формат (PNG, PDF).

Генерация схемы алгоритма с помощью ИИ DeepSeek.com

Цель задания: освоить автоматизированное создание схем алгоритмов с использованием ИИ-платформы DeepSeek.com

DeepSeek.com – это платформа ИИ, разработанная для обработки естественного языка и генерации программного кода. Сервис предоставляет расширенные возможности анализа и визуализации алгоритмов, включая автоматическое преобразование программного кода в текстовое описание схем алгоритмов на языке PlantUML.

Шаг 1. Подготовка к работе

1. Откройте веб-браузер и перейдите на сайт [DeepSeek.com](https://deepseek.com)
2. Выберите режим работы с кодом (если доступно несколько режимов)
3. Подготовьте исходный код программы (на любом языке программирования)

Шаг 2. Формирование промпта

1. Введите четкий и структурированный запрос — промпт¹, например: «*Построй схему алгоритма для этого кода: [вставить код]*».

Шаг 3. Получение и анализ результата

1. Дождитесь генерации блок-схемы
2. Проверьте наличие всех структурных элементов, корректность отображения логики алгоритма, соответствие стандартам ГОСТ
3. При необходимости уточните запрос, например: «*Исправь блок-схему: замени [описание что заменить] на [требуемый результат]*» или «*Исправь блок-схему: добавь недостающие связи*»

Шаг 4. Экспорт и сохранение

1. Скопируйте полученную схему как изображение (скриншот) и/или в текстовом формате.
2. Сохраните для отчета:
 - исходный промпт и ответ ИИ
 - изображение/описание схемы

Качество результата, выдаваемого сервисами ИИ сильно зависит от качества промпта. Поэтому в данной работе вам предложено попробовать несколько промптов для сравнения качества результатов и получения навыков эффективной работы с ИИ.

¹Промпт (от англ. *prompt* — «запрос», «подсказка») — это запрос к нейросети с целью получить желаемое изображение или текст.

Вариант 1. Простой промпт, например:

«Построй схему алгоритма для этого кода: [вставить код]»

Вариант 2. Четкий и структурированный промпт, соответствующий заданию, например:

«Преобразуй следующий код в блок-схему алгоритма по ГОСТ 19.701-90. Убедись, что:

- Используются стандартные фигуры (овал – начало/конец, ромб – условие, прямоугольник – действие).*
- Все циклы (for/while) и условные операторы (if/else) выделены.*
- Схема содержит не менее 20 блоков.*

Направления потоков указаны стрелками.

Вот код: [вставить код программы].»

Вариант 3. Четкий и структурированный промпт с профессиональной терминологией, например:

«Для приведённого ниже кода построй схему алгоритма по ГОСТ 19.701-90 в нотации PlantUML. Код программы: [вставить код программы]».

Визуализации схемы алгоритма программы, представленной в виде текстового описания на языке PlantUML в сервисе PlantUML Web Server

Цель задания: освоить создание схем алгоритмов по текстовому описанию на языке PlantUML с использованием онлайн-сервиса PlantUML Web Server.

Шаг 1. Подготовка к работе

1. Откройте веб-браузер и перейдите на официальный сайт сервиса PlantUML Web Server: <https://www.plantuml.com>
2. Ознакомьтесь с интерфейсом:
 - Панель - область для ввода кода
 - Панель - предпросмотр диаграммы в реальном времени

Шаг 2. Написание кода схемы

1. Используйте код описания схемы вашего алгоритма, предложенного сервисом DeepSeek.com

Шаг 3. Визуализация и проверка

1. Введите код в панель редактора - область для ввода кода
2. Система автоматически отобразит диаграмму на панели - предпросмотр диаграммы
3. Проверьте:
 - Корректность логических переходов
 - Наличие всех необходимых элементов
 - Соответствие ГОСТ 19.701-90

Шаг 4. Настройка внешнего вида (опционально, можно не выполнять в данной практической работе)

- Вид схемы можно редактировать, внося изменения в код описания схемы. Можно корректировать блоки для соответствия их с ГОСТ, изменять размеры и цвет блоков, направление стрелок и т.п.

Шаг 5. Экспорт результатов

6. Сохраните диаграмму. Возможные форматы сохранения:
 - PNG (для вставки в отчет)
 - SVG (для масштабируемой графики)
 - ASCII (текстовое представление)
7. Скопируйте исходный код для последующего редактирования

Выполнение сравнительного анализа использованных инструментов и выводы

Цель этапа: выявление преимуществ и недостатков каждого инструмента (Dia Diagram Editor, DeepSeek.com, PlantUML Web Server) для построения блок-схем алгоритмов и определить оптимальные условия их применения.

Шаг 1. Определение критериев сравнения

Например, каждый инструмент можно оценить по следующим параметрам: удобство интерфейса, возможности создания схем в соответствии ГОСТ 19.701-90, скорость работы, гибкость/возможность редактирования, возможности по экспортированию результатов: форматы файлов и качество изображений.

Шаг 2. Проведение анализа

Для проведения анализа можно использовать таблицу, например:

Критерий	Dia Diagram Editor	DeepSeek.com	PlantUML Web Server
Критерий 1			
Критерий 2			
Критерий ...			

Шаг 3. Составление выводов

На основе анализа необходимо сформулировать выводы: преимущества каждого инструмента, недостатки, рекомендации по применению.

Оформление отчета

1. **Титульный лист** (название работы, ФИО, группа).
 2. **Исходный код программы.**
 3. **Схемы алгоритмов:**
 - Вариант схемы, выполненный в Dia Diagram Editor
 - Варианты схемы, выполненные в DeepSeek.com + примеры используемых промптов и результаты ответов по ним
 - Вариант схемы, выполненный в PlantUML Web Server
 4. **Сравнительный анализ:**
 - Сравнение использованных инструментов по выбранным критериям
 - Оценка удобства работы в указанных программах.
 5. **Вывод** (какие навыки освоены, сложности, предложения по использованию).
 6. **Список использованных источников**
-

Критерии оценки работы

Критерии оценки:

- Соответствие блок-схем ГОСТ 19.701-90.
 - Наличие всех требуемых элементов (20+ блоков, циклы, условия).
 - Корректность работы с Dia Diagram Editor, DeepSeek.com, PlantUML Web Server.
 - Глубина анализа и выводов в отчёте.
-

Примечание:

- Если Dia недоступен, можно использовать **Draw.io** (сохранять в .png/.pdf + исходный файл).
 - Для ИИ допустимы аналоги DeepSeek.com (например, ChatGPT с плагином рисования схем).
-

Пример оформления отчета представлен в Приложении 1.

Приложение 1. Пример отчета

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»

Институт машиностроения, материалов и транспорта
Высшая школа проектной деятельности и инноваций в промышленности

ОТЧЕТ
по дисциплине «Инженерная и научная графика»
Построение схемы алгоритма

Выполнил:
студент гр. _____

_____ Фамилия И.О.
подпись

_____ дата

Проверил

_____ Фамилия И.О.
подпись

_____ дата

Санкт-Петербург, ____г.

РЕФЕРАТ

Отчет 14 с., 3 рис., 1 источник

СХЕМА АЛГОРИТМА, DIA DIAGRAM EDITION, DEEPSEEK, PLANTUML WEB SERVER.

Объектом исследования является схема алгоритма.

Цель работы — освоить проектирование схем алгоритмов в соответствии с требованиями ГОСТ 19.701-90 и приобрести навыки работы в специализированном свободно распространяемом ПО (Dia Diagram Editor, DeepSeek.com, Plantuml Web Server).

ОГЛАВЛЕНИЕ

РЕФЕРАТ	2
ВВЕДЕНИЕ	4
1. КОД ПРОГРАММЫ	5
2. ПОСТРОЕНИЕ СХЕМЫ АЛГОРИТМА В DIA DIAGRAMS EDITION	7
3. ПОСТРОЕНИЕ СХЕМЫ АЛГОРИТМА В DEEPSEEK.COM	8
4. ВИЗУАЛИЗАЦИЯ СХЕМЫ АЛГОРИТМА В СЕРВИСЕ PLANTUML WEB SERVER	12
5. СРАСНИТЕЛЬНЫЙ АНАЛИЗ И ВЫВОДЫ	13
6. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	14

ВВЕДЕНИЕ

В данной работе изучаются методы проектирования блок-схем алгоритмов программ с использованием Dia Diagram Editor, возможностей искусственного интеллекта на примере DeepSeek.com для автоматической генерации схем и сервиса визуализации PlantUML Web Server. Цель — научиться строить схемы алгоритмов с помощью разных современных инструментов.

ГОСТ 19.701-90 ЕСПД «Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения» — ключевой стандарт для визуализации алгоритмов в России. Его знание обязательно для IT-специалистов, технических писателей и студентов. Стандарт устанавливает условные обозначения для схем алгоритмов, программ, данных и систем; правила выполнения этих схем в документации на программное обеспечение; требования к графическому отображению логики программ для обеспечения единообразия и читаемости; обеспечивает понятность схем для всех участников проекта; упрощает аудит и поддержку кода.

Dia Diagram Editor — это свободный кроссплатформенный редактор диаграмм, доступный и для самостоятельной установки. Позволяет создавать различные виды диаграмм, включая блок-схемы алгоритмов, UML-диаграммы, ER-диаграммы баз данных, сетевые схемы и другие. **Dia** является удобным инструментом для визуализации алгоритмов в соответствии с требованиями **ГОСТ 19.701-90**.

DeepSeek.com — это платформа ИИ, разработанная для обработки естественного языка и генерации программного кода. Сервис предоставляет расширенные возможности анализа и визуализации алгоритмов, включая автоматическое преобразование программного кода в блок-схемы. Выдает описание схемы алгоритма программы с текстовым виде на языке PlantUML. Требуется визуализации схемы другими инструментами и проверки правильности описания алгоритма.

PlantUML Web Server — это онлайн-сервис для текстового описания диаграмм с автоматической визуализацией. Позволяет создавать блок-схемы, UML-диаграммы, ER-модели и другие схемы через простой DSL (Domain-Specific Language), выполняет автоматическую визуализацию текстового описания диаграмм на PlantUML.

В ходе работы рассмотрены особенности работы с каждым инструментом, проведен анализ их эффективности, а также даны рекомендации по выбору оптимального решения для конкретных задач. Особое внимание уделено соответствию результатов требованиям ГОСТ 19.701-90.

1. КОД ПРОГРАММЫ

```
namespace Дисциплина_Теория и технология программирования_Задание 4
{
    using System;
    using System.ComponentModel.Design;
    using System.Data;

    struct Material
    {
        public string Name;
        public double Density;
    }

    class Programm
    {
        static void Main()
        {
            static List<Material> Spisoklen()
            {
                List<Material> materials = new List<Material>
                {
                    new Material { Name = "Алюминий", Density = 2.7 },
                    new Material { Name = "Цинк", Density = 7.1 },
                    new Material { Name = "Медь", Density = 8.9 },
                    new Material { Name = "Бронза", Density = 7.8 },
                    new Material { Name = "Серебро", Density = 10.5 },
                    new Material { Name = "Золото", Density = 19.3 },
                    new Material { Name = "Олово", Density = 7.3 },
                    new Material { Name = "Литий", Density = 0.5 }
                };
                return materials;
            }
            static int Menu(List<Material> mat)
            {
                for (int i = 0; i < mat.Count; i++)
                {
                    Console.WriteLine("{0,2} {1,10} {2,10} г/см.куб", i + 1, mat[i].Name,
mat[i].Density);
                }
                Console.WriteLine("\nВыберите материал, для завершения работы программы
введите 0:");
                int matumber = Convert.ToInt32(Console.ReadLine());
                return matumber;
            }
            static double Volume(double l, double d, double c)
            {
                return l * d * c;
            }
            Double l1 = 0.0;
            Double l2 = 0.0;
            Double h1 = 0.0;
            Double h2 = 0.0;
            Double sh1 = 0.0;
            Double sh2 = 0.0;
            Double V1 = 0.0;
            Double V2 = 0.0;
            Double V = 0.0;
            Double m = 0.0;
            List<Material> materials = Spisoklen();

            if (materials.Count == 0)
            {
                Console.WriteLine("Список материалов пуст.");
                return;
            }
            int n = Menu(materials);
            while (n != 0)
            {
                if (n >= 1 & n <= materials.Count)
                {
                    Console.WriteLine("Параметры стержня 1");
                    Console.WriteLine("Длина стержня: ");
                    l1 = Convert.ToDouble(Console.ReadLine());
                    Console.WriteLine("Высота стержня: ");
                    h1 = Convert.ToDouble(Console.ReadLine());
                }
            }
        }
    }
}
```

```

        Console.WriteLine("Ширина стержня: ");
        sh1 = Convert.ToDouble(Console.ReadLine());
        V1 = Volume(l1, h1, sh1);
        Console.WriteLine("Параметры стержня 2");
        Console.WriteLine("Длина стержня: ");
        l2 = Convert.ToDouble(Console.ReadLine());
        Console.WriteLine("Высота стержня: ");
        h2 = Convert.ToDouble(Console.ReadLine());

        Console.WriteLine("Ширина стержня: ");
        sh2 = Convert.ToDouble(Console.ReadLine());

        V2 = Volume(l2, h2, sh2);

        if (V1 > V2)
        {
            V = V1 - V2;
        }
        else
        {
            V = V2 - V1;
        }

        Console.WriteLine($"Длина трубы 1, см: {l1}\nШирина трубы 1,
        трубы 1, см: {h1}\nДлина трубы 2, см: {l2}\nШирина трубы 2,
        {h2}\nМатериал: {materials[n-1].Name} - {materials[n-
        1].Density} г / см
        куб\n-----\nОбъем трубы, см
        куб: {V}");
    }

    m = V * materials[n-1].Density;
    if (m >= 1000)
    {
        m /= 1000;
        Console.WriteLine($"Масса, кг: {m}\n");
    }
    else
    {
        Console.WriteLine($"Масса, г: {m}\n");
    }
    n = Menu(materials);
}
else { }
}
}
}
}

```

2. ПОСТРОЕНИЕ СХЕМЫ АЛГОРИТМА В DIA DIAGRAM EDITION

Для того чтобы построить схему алгоритма в редакторе Dia Diagram Editor, необходимо:

1. Запустить программу и создать новый проект.
2. В разделе «Flowchart» выбрать элементы для построения схемы (овалы (начало/конец), прямоугольники (действия), ромбы (условия)).
3. Расположить элементы в требуемом порядке.
4. Соединить элементы стрелками и линиями.
5. Добавить текст, описание блоков
6. Сохранить исходный файл с расширением .dia, png/.pdf для отчета.

Схема алгоритма представлена на рис. 1.

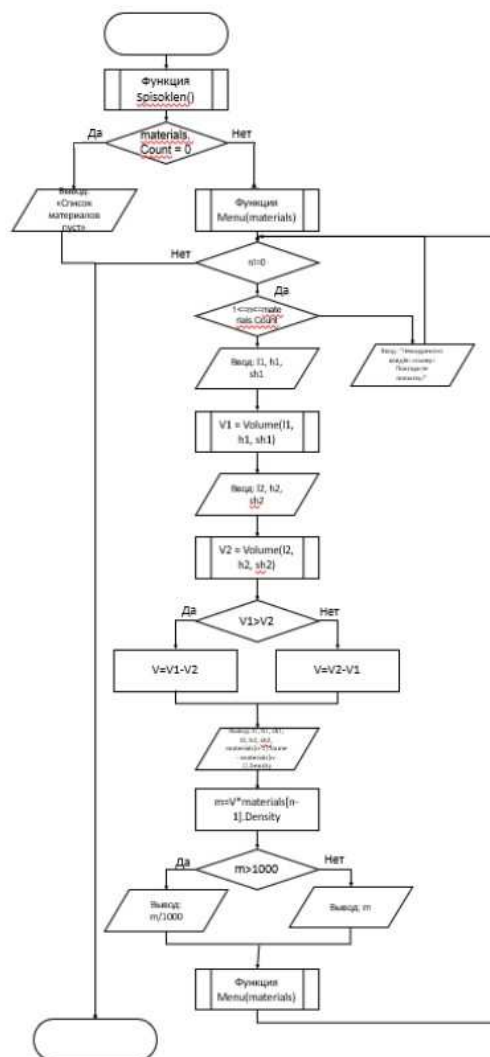


Рис. 1. Схема алгоритма программы, выполненная в программе Dia Diagram Editor

3. ПОСТРОЕНИЕ СХЕМЫ АЛГОРИТМА В DEEPSEEK.COM

Для создания блок-схемы с использованием DeepSeek.com необходимо:

1. Открыть веб-интерфейс DeepSeek.com и выбрать режим работы с кодом.
2. Ввести исходный код программы или алгоритма в текстовое поле.
3. Задать промт.
4. Получить сгенерированную блок-схему и проверить ее корректность.
5. При необходимости уточнить запрос для исправления неточностей.
6. Сохранить результат в графическом формате (.png/.jpg) для включения в отчет.

Пример промпта вариант 1: «Построй схему алгоритма для этого кода: [вставлен код из пункта 1.]»

Результат ответа к промπτу 1 приведен на рис. 2.



```
plantuml
@startuml
skinparam defaultFontName "Arial"
skinparam defaultFontSize 14
skinparam conditionStyle inside
skinparam arrowColor #000000

start
:Инициализация списка материалов;
:Вызов Menu(materials);
repeat
    if (n >= 1 && n <= materials.Count?) then (Да)
        :Ввод параметров стержня 1 (l1, h1, sh1);
        :V1 = Volume(l1, h1, sh1);
        :Ввод параметров стержня 2 (l2, h2, sh2);
        :V2 = Volume(l2, h2, sh2);
        if (V1 > V2?) then (Да)
            :V = V1 - V2;
        else (Нет)
            :V = V2 - V1;
        endif
        :Вывод параметров труб;
        :m = V * materials[n-1].Density;
        if (m >= 1000?) then (Да)
            :m /= 1000;
            :Вывод "Масса, кг: m";
        else (Нет)
            :Вывод "Масса, г: m";
        endif
    else (Нет)
        :Некорректный ввод;
    endif
    :Вызов Menu(materials);
repeat while (n != 0?)
stop
@enduml
```

Рис. 2. Текстовое описание схемы алгоритма - результат ответа DeepSeek.com на промпт №1

Пример промпта вариант 2: «Преобразуй следующий код в блок-схему алгоритма по ГОСТ 19.701-90. Убедись, что:

- Используются стандартные фигуры (овал – начало/конец, ромб – условие, прямоугольник – действие).
- Все циклы (for/while) и условные операторы (if/else) выделены.
- Схема содержит не менее 20 блоков.

Направления потоков указаны стрелками.

Вот код: [вставлен код из пункта 1]»

Результат ответа к промпту 2:

@startuml

```
skinparam defaultFontName "Arial"
skinparam defaultFontSize 14
skinparam conditionStyle inside
skinparam arrowColor #000000
```

start

```
:Инициализация переменных:
l1=l2=h1=h2=sh1=sh2=v1=v2=v=m=0.0;
:Создание списка материалов
(materials = Spisoklen());
```

```
if (materials.Count == 0?) then (Да)
    :Вывод "Список материалов пуст";
    stop
else (Нет)
    :Вызов Menu(materials)
    -> n;
endif
```

```
while (n != 0?) is (Да)
    if (n >= 1 && n <= materials.Count?) then (Да)
        :Вывод "Параметры стержня 1";
        :Ввод l1, h1, sh1;
        :V1 = Volume(l1, h1, sh1);
        :Вывод "Параметры стержня 2";
        :Ввод l2, h2, sh2;
        :V2 = Volume(l2, h2, sh2);
```

```
        if (V1 > V2?) then (Да)
            :V = V1 - V2;
        else (Нет)
            :V = V2 - V1;
        endif
```

```
        :Вывод параметров труб;
        :m = V * materials[n-1].Density;
```

```
        if (m >= 1000?) then (Да)
            :m /= 1000;
            :Вывод "Масса, кг: " + m;
        else (Нет)
            :Вывод "Масса, г: " + m;
        endif
    else (Нет)
```

```

        :Некорректный выбор материала;
    endif
    :Вызов Menu(materials)
    -> n;
endwhile (Нет)

stop
@enduml

```

Пример промпта вариант 3: «Для приведённого ниже кода построй схему алгоритма по ГОСТ 19.701-90 в нотации PlantUML. Код программы: [вставлен код из пункта 1.]»

Результат ответа к промпту 3:

```

@startuml
skinparam defaultFontName "Arial"
skinparam defaultFontSize 14
skinparam conditionStyle inside
skinparam arrowColor #000000

start
:Инициализация переменных:
l1=l2=h1=h2=sh1=sh2=V1=V2=V=m=0.0;
:Создание списка материалов
(materials = Spisoklen());

if (materials.Count == 0?) then (Да)
    :Вывод "Список материалов пуст";
    stop
else (Нет)
    :Вызов Menu(materials)
    -> n;
endif

while (n != 0?) is (Да)
    if (n >= 1 && n <= materials.Count?) then (Да)
        :Вывод "Параметры стержня 1";
        :Ввод l1;
        :Ввод h1;
        :Ввод sh1;
        :V1 = Volume(l1, h1, sh1);

        :Вывод "Параметры стержня 2";
        :Ввод l2;
        :Ввод h2;
        :Ввод sh2;
        :V2 = Volume(l2, h2, sh2);

        if (V1 > V2?) then (Да)
            :V = V1 - V2;
        else (Нет)
            :V = V2 - V1;
        endif

        :Вывод параметров труб;
        :m = V * materials[n-1].Density;

        if (m >= 1000?) then (Да)
            :m /= 1000;
            :Вывод "Масса, кг: " + m;

```

```
        else (Нет)
            :Вывод "Масса, г: " + m;
        endif
    else (Нет)
        :Некорректный выбор материала;
    endif
    :Вызов Menu(materials)
    -> n;
endwhile (Нет)

stop
@enduml
```


4. ВИЗУАЛИЗАЦИИ СХЕМЫ АЛГОРИТМА ПРОГРАММЫ В SERVICE PLANTUML WEB SERVER

Для визуализации схемы с использованием PlantUML Web Server необходимо:

1. Открыть веб-интерфейс www.plantuml.com
7. Ввести описание блок-схемы на языке PlantUML в текстовое поле.
8. Получить визуализированную блок-схему и проверить ее корректность.
9. При необходимости откорректировать код описания схемы для исправления неточностей.
10. Сохранить результат в графическом формате для включения в отчет.

Пример схемы алгоритма, выполненного на основе текстового описания PlantUML сгенерированного DeepSeek.com на основе промпта №3 с кодом исходной программы приведен на рис.3.

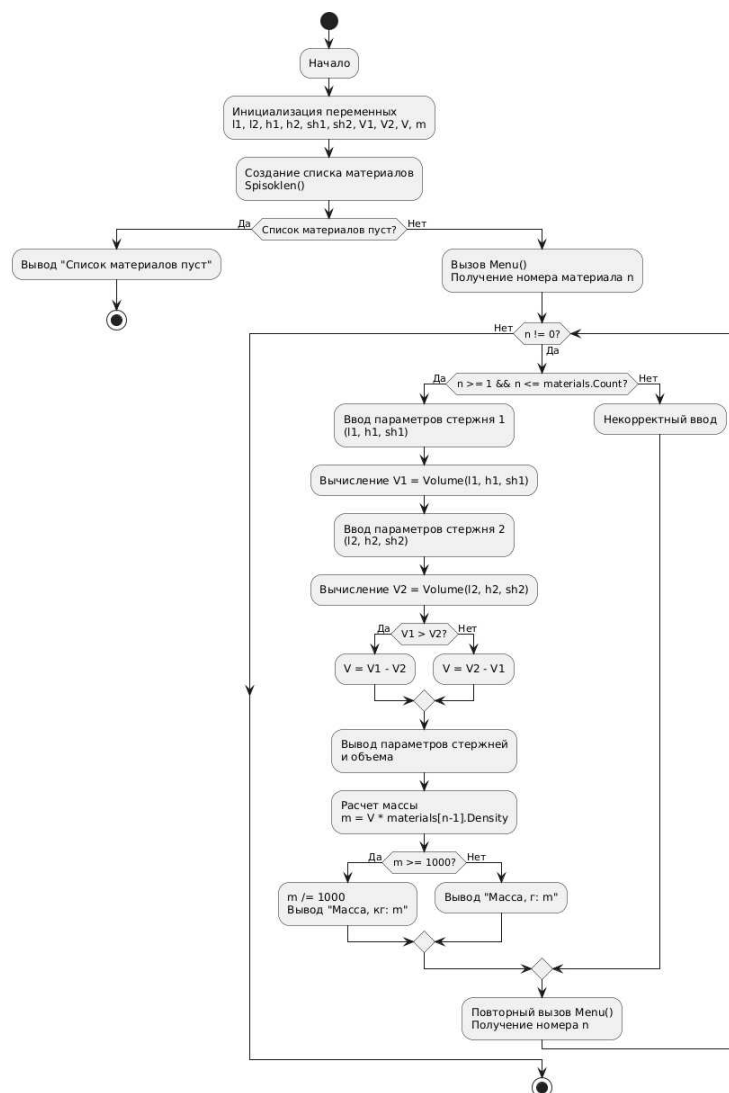


Рис. 3. Схема алгоритма, визуализированного в программе PlantUML Web Server.

5. СРАВНИТЕЛЬНЫЙ АНАЛИЗ И ВЫВОДЫ

Сравним изученные инструменты по следующим параметрам: удобство интерфейса, возможности создания схем в соответствии ГОСТ 19.701-90, скорость работы, гибкость/возможность редактирования, возможности по экспортированию результатов: форматы файлов и качество изображений. Значения критериев указаны в Табл. 1.

Табл. 1. Значения критериев оценки инструментов Dia Diagram Editor, DeepSeek.com, PlantUML Web Server

Критерий	Dia Diagram Editor	DeepSeek.com	PlantUML Web Server
Тип интерфейса	Графический	Чат-бот	Текстовый редактор
Соответствие ГОСТ	Соответствует	Не соответствует, представляет только текстовое описание	Не все элементы отображения соответствуют ГОСТ
Время создания	12-30 мин	2-5 мин	5-10 мин
Сложность освоения	Средняя	Низкая	Высокая
Гибкость	Высокая	Ограниченная	Средняя
Экспорт	PNG, PDF, SVG	Скриншот	PNG, SVG, ASCII

Выводы:

1. В ходе работы успешно освоены принципы проектирования схем алгоритмов в соответствии с ГОСТ 19.701-90
2. Работа позволила сравнить три подхода к созданию схем алгоритмов (в графическом редакторе, ИИ-генерация, текстовое описание) и определить оптимальные сферы их применения.
3. Для работ, требующих строгого соответствия ГОСТ 19.701-90, оптимальным является использование Dia Diagram Editor, несмотря на большее время создания схем.
4. DeepSeek.com демонстрирует высокую скорость генерации, но требует обязательной последующей проверки и визуализации текстового описания схемы алгоритма, оптимально подходит для быстрого прототипирования.
5. PlantUML Web Server наиболее удобен для разработчиков, работающих с системами версионного контроля, оптимален для командной разработки.

6. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения — Введ. 1992-01-01. — М.: Стандартиформ, 2010