

# Практическое занятие №1. Разработка и визуализация алгоритмов. Построение блок-схем.

## 1. Тема работы

Разработка алгоритма по словесному описанию и его визуальное представление в виде блок-схемы с использованием инструмента `draw.io`.

## 2. Цель работы

Сформировать умение анализировать словесное описание задачи, разрабатывать на его основе формальный алгоритм и корректно отображать его в виде блок-схемы, соблюдая общепринятые стандарты.

## 3. Задачи работы

- Закрепить понимание основных алгоритмических конструкций (линейная, ветвление, цикл).
- Научиться переводить словесное описание процесса в формальный алгоритм.
- Освоить принципы построения блок-схем.
- Приобрести практические навыки работы с графическим редактором `draw.io` для создания блок-схем.

## 4. Ход работы

1. **Подготовка:** Изобразите и опишите основные элементы блок-схем и правила их построения:
  - **Терминатор** (начало/конец алгоритма).
  - **Процесс** (вычислительное действие или последовательность действий).
  - **Решение** (условие, ветвление).
  - **Ввод/Вывод** данных.
  - **Предопределенный процесс** (вызов подпрограммы или функции).
  - **Соединитель** и **Линии потока**.
2. **Анализ задания:** Получите у преподавателя вариант алгоритма (см. раздел "Варианты заданий"). Внимательно прочитайте и проанализируйте его, выделив ключевые шаги, условия и циклы.
3. **Разработка алгоритма:** Продумайте логику работы алгоритма на бумаге.
4. **Построение блок-схемы:**

- Создайте новую диаграмму и выберите базовый шаблон.
- В палитре элементов найдите раздел "Блок-схема" и перетащите необходимые фигуры на холст.
- Постройте блок-схему вашего алгоритма, соединяя элементы линиями потока. Соблюдайте четкость и аккуратность.
- Подпишите все элементы: укажите внутри фигур конкретные действия (например, `a = b + c`) или условия (например, `X > 0?`).
- Убедитесь, что блок-схема имеет единственный элемент "Начало" и "Конец".

5. **Сохранение и отправка:** Сохраните проект в формате `.xml` (для последующего редактирования) и экспортируйте готовую блок-схему в формате `.pdf` или изображение ( `.png` , `.jpg` ) для сдачи преподавателю.

6. **Оформление отчета:** Оформите отчет по выполненной работе.

Отчёт должен быть оформлен в виде электронного документа (например, Word) и содержать следующие пункты:

- ФИО студента, группа, дата.
- Цель работы.
- Задание по варианту: Дословно перепишите текст вашего задания.
- Блок-схемы алгоритмов: Вставка готового изображения блок-схемы, созданной в draw.io. Схема должна быть четкой, легко читаемой и содержать все необходимые элементы.
- Ответы на контрольные вопросы.
- Вывод по работе: Краткое описание того, что было сделано, с какими сложностями столкнулись и как их преодолели. Ответ на вопрос: достигнута ли цель работы?

---

## 5. Контрольные вопросы

1. Дайте определение алгоритма. Перечислите основные свойства алгоритмов.
2. Что такое блок-схема? Каковы ее основные advantages перед словесным описанием?
3. Перечислите и изобразите основные элементы блок-схем и их назначение.
4. Как на блок-схеме обозначается:
  - начало и конец программы?
  - ввод данных с клавиатуры?
  - проверка условия?
  - вызов вспомогательного алгоритма?

5. Какие три основные типовые структуры лежат в основе любого алгоритма? Приведите примеры их изображения на блок-схемах.
- 

## 6. Варианты заданий

### Задания на структуру «Ветвление»

#### Вариант 1 (Ветвление): Алгоритм определения скидки

Алгоритм получает на вход сумму покупки. Если сумма покупки больше 1000 рублей, то рассчитывается скидка в размере 5% от суммы. Если сумма покупки больше 5000 рублей, то скидка составляет 10%. Если сумма не превышает ни один из порогов, скидка не начисляется. В конце алгоритм должен вывести итоговую сумму к оплате.

#### Вариант 2 (Ветвление): Алгоритм проверки числа

Алгоритм запрашивает у пользователя целое число. Далее он определяет и выводит на экран сообщение о том, является ли это число положительным, отрицательным или нулём, а также является ли оно чётным или нечётным.

#### Вариант 3 (Ветвление): Алгоритм принятия решения

Алгоритм запрашивает у пользователя температуру воздуха на улице. Если температура ниже 0°C, вывести сообщение «Надень зимнюю куртку». Если температура между 0°C и 15°C, вывести «Надень весеннюю куртку». Если температура выше 15°C, вывести «Можно идти в футболке».

#### Вариант 4 (Ветвление): Алгоритм проверки пароля

Алгоритм запрашивает у пользователя логин и пароль. Если логин «admin» и пароль «12345», вывести сообщение «Доступ разрешён». В противном случае вывести «Неверный логин или пароль».

#### Вариант 5 (Ветвление): Алгоритм определения високосного года

Алгоритм запрашивает у пользователя год. Год является високосным, если он кратен 4, но не кратен 100, либо если он кратен 400. Алгоритм должен вывести «Год високосный» или «Год не високосный».

### Задания на структуру «Цикл»

#### Вариант 1 (Цикл): Алгоритм подсчёта суммы

Алгоритм запрашивает у пользователя 10 целых чисел. Он должен подсчитать и вывести сумму только тех из них, которые являются положительными (больше нуля).

#### **Вариант 2 (Цикл): Алгоритм поиска факториала**

Алгоритм запрашивает у пользователя целое положительное число N. Необходимо вычислить и вывести факториал этого числа – произведение всех чисел от 1 до N. (Например,  $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$ ).

#### **Вариант 3 (Цикл): Алгоритм вывода последовательности**

Алгоритм запрашивает у пользователя целое число  $N > 0$ . Алгоритм должен вывести на экран все целые числа от 1 до N, при этом каждое число должно быть выведено на новой строке.

#### **Вариант 4 (Цикл): Алгоритм проверки на простоту**

Алгоритм запрашивает у пользователя целое число  $N > 1$ . Алгоритм должен проверить, является ли это число простым (делится без остатка только на 1 и на себя), и вывести сообщение «Простое» или «Составное».

#### **Вариант 5 (Цикл): Алгоритм обработки числа**

Алгоритм запрашивает у пользователя целое положительное число. Необходимо подсчитать и вывести сумму цифр, из которых состоит это число. (Например, для числа 12345 сумма цифр равна  $1+2+3+4+5=15$ ).