Уважаемые студенты!
Вам необходимо выполнить следующее:

Задание:

- Изучить теорию;
- Написать краткий конспект

Результаты работы, фотоотчет, предоставить преподавателю на e-mail: v.vika2014@mail.ru

При возникновении вопросов по приведенному материалу обращаться по следующему номеру телефона: (072)1744922

ВНИМАНИЕ!!! При отправке работы, не забывайте указывать ФИО студента, наименование дисциплины, дата проведения занятия (по расписанию).

Тема: Графическое представление базовых структур алгоритмов.

Алгоритмы и их графическое представление

Понятие алгоритма и его свойства

Алгоритм – точно определенное правило действий, для которого задано указание, как и в какой последовательности это правило необходимо применять к исходным данным задачи, чтобы получить ее решение за конечное число шагов.

Свойства алгоритмов:

1. **Детерминированность** (*определенность*) — однозначность результата процесса при заданных исходных данных.

2. **Дискретность** (*прерывность*) — алгоритм должен представлять процесс решения задачи как последовательное выполнение элементарных шагов, возможность выполнения которых человеком или машиной не вызывает сомнения.

3. **Массовость** — алгоритм решения задачи разрабатывается в общем виде, т.е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными. При этом исходные данные могут

выбираться из некоторой области, называемой *областью применимости алгоритма*.

4. **Результативность** — за конечное число шагов алгоритм либо должен приводить к решению задачи, либо после конечного числа шагов останавливаться из-за невозможности получить решение с выдачей соответствующего сообщения, либо неограниченно продолжаться в течение времени, отведенного для исполнения алгоритма, с выдачей промежуточных результатов.

Формы представления алгоритмов

На практике наиболее распространены следующие формы представления алгоритмов:

- **словесная** – запись алгоритма на естественном языке;
- **графическая** – изображение алгоритма в виде схемы, состоящей из графических символов;
- **аналитическая** – запись алгоритма в виде последовательности формул;
- **псевдокодом** – полуформализованное описание алгоритма на условном алгоритмическом языке, включающем как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.;
- **программная** – запись алгоритма в виде текста на одном из языков программирования.

При **графическом представлении** алгоритм изображается в виде последовательности связанных между собой функциональных блоков, отображаемых в виде графических фигур, каждая из которых соответствует выполнению одного или нескольких действий.

Такое представление называется *схемой алгоритма* (иногда называют *блок-схемой*).

Схема алгоритма – графическое изображение его структуры, в котором каждый этап процесса обработки данных представляется в виде различных геометрических фигур (примитивов).

В таблице 1 приведены часто употребляемые графические примитивы в схемах алгоритмов.

Таблица 1

Графические примитивы схем алгоритмов

Название примитива	**Обозначение и пример заполнения**	**Пояснение**
Процесс	x = (c–d)*e a b	Вычислительное действие или последовательность действий
Решение	Да a < b Нет	Выбор направления выполнения алгоритма или программы в зависимости от некоторых изменяемых условий
Модификация	i = 1,50,2	Начало циклического процесса
Предопределенный процесс	Расчет параметров	Вычисления по подпрограмме, стандартной подпрограмме
Ввод-вывод	Ввод A, B, C	Преобразование данных в форму, пригодную для обработки (ввод) или отображения полученных результатов (вывод)
Пуск-останов	Начало	Начало, конец алгоритма, вход и выход в подпрограмму
Документ	Печать a, c	Вывод результатов на печать
Комментарий	Текст пояснения, относящийся к определенному графическому символу	Связь между элементом схемы и пояснением
Узел	5	Указание связи между прерванными линиями потока, связывающего символы
Ссылка на другую страницу		Обозначение связи между различными частями схемы алгоритма, размещенных на разных страницах
Линия потока		Обозначение последовательности связей между символами

Правила составления схем алгоритмов

Составление схем алгоритмов регламентируется ГОСТами:

- **ГОСТ 19.002-80** соответствует международному стандарту **ИСО 2636-73** и определяет правила составления схем алгоритмов;
- **ГОСТ 19.003-80** соответствует международному стандарту **ИСО 1028-73** и регламентирует использование графических примитивов.

В соответствии с этими документами при составлении схем алгоритмов необходимо соблюдать следующие основные правила:

1. Графические примитивы соединяются между собой *линиями потока*, которые для каждого шага алгоритма указывают возможных преемников.

2. Внутри примитива дается краткое описание соответствующего шага.

3. Любой шаг или поток можно сопроводить пояснением с помощью примитива «*Комментарий*».

4. Примитивам присваиваются порядковые номера, проставляемые в разрыве линии контура в левом верхнем углу изображения.

5. Линии потока проводятся параллельно линиям внешней рамки схемы. Направление линии потока сверху вниз и слева направо принято за основной. Если они не имеют изломов, их можно не помечать стрелками. В остальных случаях их направление обязательно помечается стрелкой. Линию потока, как правило, подводят к середине символа.

6. Расстояние между параллельными линиями потока должно быть не менее 3 мм, а между другими символами – не менее 5 мм.

7. Линию потока можно обрывать, используя на месте обрыва примитив «*Узел*» в пределах одной страницы или примитив «*Ссылка на другую страницу*», если надо указать переход на другую страницу.

8. Размеры примитивов фиксированы. Если через b обозначить ширину элемента, а через a его высоту, то размер $b = 1{,}5a$, причем a должно принимать значения 10, 15, 20 мм. Допускается увеличивать размер a на величину, кратную 5.

Пример. Составить алгоритм вычисления суммы элементов вектора $\mathbf{a} = (a_1, a_2, \ldots, a_n)$, т.е. $S = \sum_{i=1}^{n} a_i$.

Для определения суммы применим алгоритм ее накопления, используя рекуррентное соотношение

$S_i := S_{i-1} + a_i$; $S_0 = 0$; $i := 1, 2, ..., n$,

где S_i — текущее значение суммы;

S_{i-1} — предыдущее ее значение;

S_0 — начальное значение суммы;

a_i — значение текущего элемента вектора.

Сначала значение суммы равняется нулю, потом последовательно к ней добавляются значения первого, второго и остальных элементов вектора.

Схема алгоритма приведена на рис. П1.1. Поясним ее.

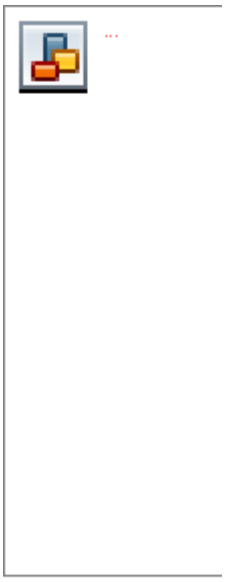

Рис. П1.1. Схема алгоритма вычисления суммы элементов вектора a

1. Схема начинается с символа «*Пуск-останов*». Каждый вычислительный процесс имеет начало, и это отображается на схеме.

2. Начальные данные вводятся в память компьютера. Для обозначение этой операции используется символ «Ввода-вывода», внутри которого указываются, какие элементы вводятся.

3. Сумме присваивается начальное значение, равное нулю (S:=0). Это означает пересылку 0 в область памяти, предназначенную для накопления суммы. Операция на схеме отображается символом «*Процесс*». Принятое обозначение суммы S расшифровывается символом «*Комментарий*».

4. Индексу i, определяющему порядковый номер элемента, присваивается значение 1. При $i = 1$ происходит обращение к первому элемента вектора, т.е. a_1.

5. К сумме S (на первом шаге $i = 1$, $S = 0$ и $a_i = a_1$) прибавляется значение элемента вектора a_i ($S := S + a_i$). В области памяти S записывается новое значение суммы.

6. С увеличением значения i на 1 ($i = i + 1$) определяется порядковый номер следующего элемента вектора.

7. Количество элементов вектора равно n. Отсюда, операция суммирования $S := S + a_i$ должна повторяться n раз, для чего осуществляется проверка: продолжать вычисление суммы или нет. Для выбора направления вычислений применяется символ «*Решение*». Внутри него указывается проверка условия $i <= n$. Если значение i не превысило своего максимального значения n, то операция вычисления суммы повторяется (переход к шагу 5), в противном случае осуществляется переход к шагу 8.

8. Осуществляется вывод полученной суммы S. Эта операция отображается на схеме символом «Ввода-вывода», внутри которой указывается, что именно выводится.

9. Схема заканчивается символом «Пуск-останов».