УВАЖАЕМЫЕ СТУДЕНТЫ!

ВАМ НЕОБХОДИМО ВЫПОЛНИТЬ СЛЕДУЮЩЕЕ:

1. Ознакомиться с теорией и законспектировать лекцию не меньше трех листов, составить и ответить на вопросы.

2. Предоставит отчет конспекта лекции прислать в виде скриншото в течении трех дней. тел. 072-17-44-9-22.

3. Отправить преподавателю на почту **v.vika2014@mail.ru** и указать свою Ф.И.О, группу, и название дисциплины

Тема: Формы представления алгоритма.

Алгоритмы можно записывать разными способами, называемыми ***формой представления алгоритма***. На практике наиболее распространены следующие формы представления алгоритмов:

- *словесная* (записи на естественном языке);
- *графическая* (изображения из графических символов);
- *псевдокоды* (полуформализованные описания алгоритмов на условном алгоритмическом языке, включающие в себя как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.);
- *программная* (тексты на языках программирования).

Словесная форма представления алгоритмов

Словесная форма записи не так широко распространена в литературе из-за ее многословности и отсутствия наглядности.

Рассмотрим запись алгоритма в словесной форме на примере алгоритма нахождения максимального из двух значений:

Определим форматы переменных X, Y, M, где X и Y – значения для сравнения, M – переменная для хранения максимального значения.

1. Получим два значения чисел X и Y для сравнения;
2. сравним X и Y;

3. если X меньше Y, значит большее число Y;
4. поместим в переменную M значение Y;
5. если X не меньше (больше) Y, значит большее число X;
6. поместим в переменную M значение X.

Как видно из данного примера словесный способ описания обладает следующими недостатками:

- описание строго не формализуемо;
- запись получилась многословной;
- отдельные предписания (действия) допускают неоднозначность толкования.

Именно эти три причины не позволили получить широкое распространение словесной форме записи.

Графический способ представления алгоритмов

Графический способ оказался очень удобным средством изображения алгоритмов и получил широкое распространение в научной и учебной литературе.

Структурная (блок-) схема алгоритма – графическое изображение алгоритма в виде схемы связанных между собой с помощью стрелок (линий перехода) блоков – графических символов, каждый из которых соответствует одному шагу алгоритма. Внутри каждого блока дается описание соответствующего действия.

Графическое изображение алгоритма широко используется перед программированием задачи вследствие его наглядности, т.к. зрительное восприятие обычно облегчает процесс написания программы, ее корректировки при возможных ошибках, осмысливание процесса обработки информации.

Можно встретить даже такое утверждение: «Внешне алгоритм представляет собой схему – набор прямоугольников и других символов, внутри которых записывается, что вычисляется, что вводится в машину и что

выдается на печать и другие средства отображения информации». Здесь форма представления алгоритма смешивается с самим алгоритмом.

Принцип программирования «сверху вниз» требует, чтобы блок-схема поэтапно конкретизировалась и каждый блок «расписывался» до элементарных операций. Но такой подход можно осуществить при решении несложных задач. При решении сколько-нибудь серьезной задачи блок-схема «расползется» до такой степени, что ее невозможно будет охватить одним взглядом.

Блок-схемы алгоритмов удобно использовать для объяснения работы уже готового алгоритма, при этом в качестве блоков берутся действительно блоки алгоритма, работа которых не требует пояснений. Блок-схема алгоритма должна служить для упрощения изображения алгоритма, а не для усложнения.

В таблице приведены наиболее часто употребляемые символы.

Название символа	**Обозначение и пример заполнения**	**Пояснение**
Процесс	x=(a-b)/sin(1)	Вычислительное действие или последовательность действий
Решение	да a<b нет	Проверка условий
Модификация	i=1,50,2	Начало цикла
Предопределенный процесс	Расчет параметров	Вычисления по подпрограмме, стандартной подпрограмме
Ввод-вывод	Ввод a,b,c	Ввод-вывод в общем виде
Пуск-останов	Начало	Начало, конец алгоритма, вход и выход вподпрограмму
Документ	Печать a,c	Вывод результатов на печать

Блок «процесс» применяется для обозначения действия или последовательности действий, изменяющих значение, форму представления или размещения данных. Для улучшения наглядности схемы несколько отдельных блоков обработки можно объединять в один блок. Представление отдельных операций достаточно свободно.

Блок «решение» используется для обозначения переходов управления по условию. В каждом блоке «решение» должны быть указаны вопрос, условие или сравнение, которые он определяет.

Блок «модификация» используется для организации циклических конструкций. (Слово модификация означает видоизменение, преобразование). Внутри блока записывается параметр цикла, для которого указываются его начальное значение, граничное условие и шаг изменения значения параметра для каждого повторения.

Блок «предопределенный процесс» используется для указания обращений к вспомогательным алгоритмам, существующим автономно в виде некоторых самостоятельных модулей, и для обращений к библиотечным подпрограммам.

Псевдокод

Псевдокод представляет собой систему обозначений и правил, предназначенную для единообразной записи алгоритмов. Он занимает промежуточное место между естественным и формальным языками. С одной стороны, псевдокод близок к обычному естественному языку, поэтому алгоритмы могут на нем записываться и читаться как обычный текст. С другой стороны, в псевдокоде используются некоторые формальные конструкции и математическая символика, что приближает запись алгоритма к общепринятой математической записи. В псевдокоде не приняты строгие синтаксические правила для записи команд, присущие формальным языкам, что облегчает запись алгоритма на стадии его проектирования и дает возможность использовать более широкий набор команд, рассчитанный на абстрактного исполнителя. Однако в псевдокоде обычно имеются некоторые конструкции, присущие формальным языкам, что облегчает переход от записи на псевдокоде к записи алгоритма на формальном языке. В частности, в псевдокоде, так же, как и в формальных языках, есть служебные слова, смысл которых определен раз и навсегда. Они выделяются в печатном тексте жирным шрифтом, а в рукописном тексте подчеркиваются. Единого или

формального определения псевдокода не существует, поэтому возможны различные псевдокоды, отличающиеся набором служебных слов и основных (базовых) конструкций. К таким конструкциям обычно относят ветвления (*если ...то ... иначе ...*) и циклы (*цикл от ... до ...*, *цикл пока*, *цикл до...*).

При записи алгоритма в словесной форме, в виде блок-схемы или на псевдокоде допускается определенный произвол при изображении команд. Вместе с тем такая запись точна настолько, что позволяет человеку понять суть дела и исполнить алгоритм.

Однако на практике в качестве исполнителей алгоритмов используются специальные автоматы — компьютеры. Поэтому алгоритм, предназначенный для исполнения на компьютере, должен быть записан на «понятном» ему языке. И здесь на первый план выдвигается необходимость точной записи команд, не оставляющей места для произвольного толкования их исполнителем. Следовательно, язык для записи алгоритмов должен быть формализован. Такой язык принято называть языком программирования, а запись алгоритма на этом языке — программой для компьютера.