1. Алгоритмизация вычислительных процессов. Основные понятия и классификация вычислительных процессов. Сравнительный анализ различных типов вычислительных процессов.
2. Линейные вычислительные процессы (на примере вычисления степенного полинома по схеме Горнера).
3. Детерминированные циклические вычислительные процессы с управлением по аргументу (на примере вычисления функции в заданном диапазоне изменения аргумента).
4. Детерминированные циклические вычислительные процессы с управлением по аргументу (на примере вычисления функции, представленной суммой или произведением).
5. Детерминированные циклические вычислительные процессы с управлением по аргументу (на примере построения графика тригонометрической функции).
6. Детерминированные циклические вычислительные процессы с управлением по индексу (на примере задачи исследования характера изменения фазового угла и реактивного сопротивления колебательного контура Zэ на различных частотах fi).
7. Итерационные циклические вычислительные процессы с управлением по функции (на примере решения нелинейного уравнения). + обязательный вывод формул
8. Итерационные циклические вычислительные процессы с управлением по аргументу и функции (на примере приближенного решения элементарных функций). + обязательный вывод формул
9. Итерационные циклические вычислительные процессы с управлением по аргументу и функции (на примере построения переходной характеристики заряда конденсатора в схеме RC цепочки с заданной точностью ε).
10. Итерационные циклические вычислительные процессы с управлением по индексу и функции (на примере сортировки элементов массива по заданному признаку).
11. Разветвляющиеся вычислительные процессы (на примере расчета функции по различным формулам в зависимости от заданного диапазона изменения аргумента).
12. Разветвляющиеся вычислительные процессы (на примере поиска максимального (минимального) элемента в массиве).
13. Разветвляющиеся вычислительные процессы (на примере построения вариационного ряда).
14. Комбинированные вычислительные процессы (на конкретном примере).
15. Многоступенчатые вычислительные процессы (на примере задачи нахождения суммы элементов двумерного массива).
16. Массивы (с рассмотрением примеров разных способов ввода элементов массива на языке Pascal).
17. Процедуры и функции (на конкретных примерах).
18. Структуры для организации циклов (на конкретном примере).
19. Структуры для организации разветвлений (на конкретном примере).
20. Вычисление определенного интеграла. Метод прямоугольников (на конкретном примере). + обязательный вывод формул
21. Вычисление определенного интеграла. Метод трапеции (на конкретном примере). + обязательный вывод формул
22. Вычисление определенного интеграла. Метод парабол (на конкретном примере). + обязательный вывод формул
23. Способы представления алгоритма. ГОСТ 19.701-90. ЕСПД. Схемы алгоритмов, программ, данных и систем (с приведением конкретных примеров).

**22. Алгоритмизация вычислительных процессов. Основные понятия и классификация вычислительных процессов. Сравнительный анализ различных типов вычислительных процессов.**

1. **Основные понятия**

**Алгоритмизация** - процесс построения алгоритма решения задачи, результатом которого является выделение этапов процесса обработки данных, формальная запись содержания этих этапов и определение порядка их выполнения.

**Алгоритм** - это точное предписание, определяющее вычислительный процесс, ведущий от варьируемых начальных данных к искомому результату.

**Свойства алгоритма :**

* Дискретность (раздельность) - состоит из отдельных шагов (команд), возможно разделение вычислительного процесса на отдельные элементарные операции, возможность выполнения которых не вызывает сомнений
* Понятность - включает только те команды, что известны исполнителю
* Определенность - при одинаковых исходных данных всегда выдает один и тот же результат, точность указаний, исключающая их произвольное толкование
* Конечность - заканчивается за конечное число шагов
* Массовость - может применяться многократно при различных исходных данных
* Корректность - дает верное решение при любых допустимых исходных данных

**Команда** - это описание действий, которые должен выполнить компьютер: 1) откуда взять исходные данные; 2) что с ними нужно сделать?

**Программа** - это 1) алгоритм, записанный на каком-либо языке программирования; 2) набор команд для компьютера

**Программа** - алгоритм, записанный в форме, воспринимаемой машиной.

Программа содержит наряду с описанием данных команд, в какой последовательности, над какими данными и какие операции должна выполнять машина, а также в какой форме следует получить результат. Это обеспечивают различные **операторы.**

1. **Классификация ВП**

**Линейные ВП** - алгоритмы, в которых операции выполняются последовательно одна за другой, в естественном и единственном порядке следования.

В таких алгоритмах все блоки имеют последовательное соединение логической связью передачи информационных потоков. В алгоритмах с линейной структурой каждый оператор выполняется только один раз и выполняется целиков, т.е выполняются ВСЕ операторы.

Линейные ВП, как правило, являются составной частью других ВП.

**Разветвляющиеся ВП -** алгоритмы, в которых в зависимости от выполнения некоторого логического условия происходит разветвление вычислений по одному из нескольких возможных направлений.

Каждое возможное направление вычислений называется ветвью, которая ведет к общему выходу, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран.

Структура ветвления существует в трех основынх вариантах:

* если-то **(if..then)**
* если-то-иначе **(if-then-else)**
* выбор **(case of)**

**Циклические ВП -** предусматривает многократное повторение действий в одной и той же последовательности по одним и тем же математическим зависимостям, но при разных значениях некоторой специально изменяемой величины.

ЦВП позволяют существенно сократить объем программы за счет многократного выполнения группы повторяющихся вычислений, так называемого **тела цикла.**

Специально изменяемый по заданному закону параметр, входящий в тело цикла называется **переменной цикла.** Она отслеживает условие окончания цикла.

Существуют следующие разновидности ЦВП:

* Оператор цикла с параметром (типа счетчик);
* Оператор цикла с предусловием;
* Оператор цикла с постусловием

Если количество повторов известно заранее, то используется оператор цикла типа счетчик, т.е **детерминированный ВП.**

Если количество повторов неизвестно, но задано некоторое условие окончания или продолжения цикла применяются операторы цикла с предусловием и постусловием. Такие циклы используют для построение **итерационных ВП**. Особенностью **итерационного цикла** является то, что число повторений операторов тела цикла заранее неизвестно. На каждом шаге вычислений происходит ***последовательное приближение к искомому результату и проверка условия достижения последнего.***

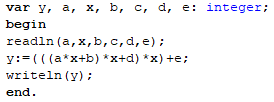
# 23. Линейные вычислительные процессы (на примере вычисления степенного полинома по схеме Горнера).

Линейные вычислительные процессы (ЛВП) характеризуются последовательным выполнением операторов программы и блоков вычислительного алгоритма. ЛВП, как правило, являются составной частью циклического или разветвляющегося вычислительного процесса.

При разработке вычислительного алгоритма целесообразно стремиться к минимизации количества вычислительных операторов и к минимальному использованию объема памяти, не ухудшая при этом точность вычисления выбранной последовательностью вычислительных действий. Например, степенной полином целесообразно вычислять по схеме Горнера:



Код:

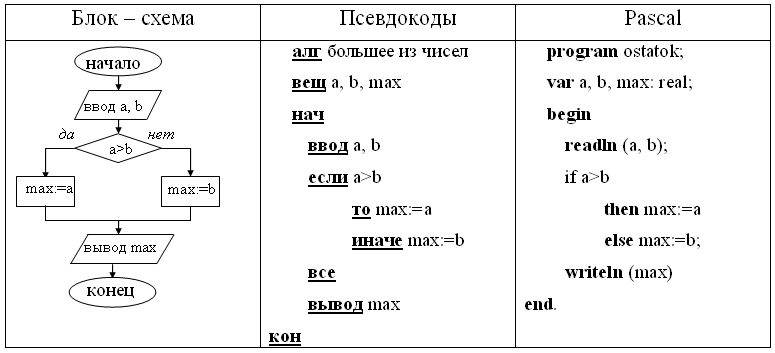


Блок-схема:

**44. Способы представления алгоритма. ГОСТ 19.701-90. ЕСПД. Схемы алгоритмов, программ, данных и систем (с приведением конкретных примеров).**

1. ***Словесный способ*** - содержание этапов вычисления задается на естественном языке в произвольной форме с требуемой детализацией:

Недостатки: 1) отсутствует наглядность ВП, т.к нет достаточной формализации; 2) страдают многословностью записей; 3) допускают неоднозначность толкования отдельных предписаний. Пример: Записать алгоритм нахождения наибольшего общего делителя (НОД) двух натуральных чисел ***(Алгоритм Эвклида)***

1. задать два числа;
2. если числа равны, то взять любое из них в качестве ответа и остановиться, в противном случае продолжить выполнение алгоритма;
3. определить большее из чисел;
4. заменить большее из чисел разностью большего и меньшего чисел;
5. повторить алгоритм с шага 2.
6. ***Псевдокод*** - компактный язык описания алгоритмов, использующий ключевые слова языков программирования, но опускающий несущественные подробности и специфический синтаксис.
7. ***Графический способ (блок-схема)*** - это графическое изображение логической структуры алгоритма, в котором каждый этап процесса переработки данных представляется в виде геометрических фигур (блоков), имеющих определенную конфигурацию. в зависимости от характера выполняемых операций. Блочные символы соединяются ***линиями переходов,*** определяющими очередность выполняемых действий.

| **Блок** | **Обозначение** | **Описание** |
| --- | --- | --- |
| Терминатор начала и конца работы функции |  | Терминатором начинается и заканчивается любая функция. |
| Операции ввода и вывода данных |  | В ГОСТ определено множество символов ввода/вывода, например вывод на магнитные ленты, дисплеи и т.п. Если источник данных не принципиален, обычно используется символ параллелограмма. Подробности ввода/вывода могут быть указаны в комментариях. |
| Выполнение операций над данными |  | В блоке операций обычно размещают одно или несколько (ГОСТ не запрещает) операций присваивания, не требующих вызова внешних функций. |
| Блок, иллюстрирующий ветвление алгоритма |  | Блок в виде ромба имеет один вход и несколько подписанных выходов. В случае, если блок имеет 2 выхода (соответствует оператору ветвления), на них подписывается результат сравнения — «да/нет». Если из блока выходит большее число линий (оператор выбора), внутри него записывается имя переменной, а на выходящих дугах — значения этой переменной. |
| Вызов внешней процедуры |  | Вызов внешних процедур и функций помещается в прямоугольник с дополнительными вертикальными линиями. |
| Ручной ввод |  | Символ отображает данные, вводимые вручную во время обработки с устройства любого типа. |
| Подготовка данных |  | Символ «подготовка данных» в произвольной форме (в ГОСТ нет ни пояснений, ни примеров), задает входные значения. Используется обычно для задания циклов со счетчиком. |
| Соединитель |  | В случае, если блок-схема не помещается на лист, используется символ соединителя, отражающий переход потока управления между листами. Символ может использоваться и на одном листе, если по каким-либо причинам тянуть линию неудобно. |
| Комментарий |  | Комментарий может быть соединен как с одним блоком, так и группой. Группа блоков выделяется на схеме пунктирной линией. |

| [Топология](http://sernam.ru/book_e_math.php?id=135) | Преимущества | Недостатки |
| --- | --- | --- |
| 1 | 2 | 3 |
| «Шина» | - экономный расход кабеля;  - недорогая и несложная в использовании среда передачи;  - простота и надежность;  - легкая расширяемость | - при значительных объемах трафика уменьшается пропускная способность;  - трудная локализация проблем;  -выход из строя любого сегмента кабеля остановит работу всей сети |
| «Кольцо» | - все PC имеют равный доступ;  - количество пользователей не сказывается на производительности | - выход из строя одной PC выводит из строя всю сеть;  - трудно локализовать проблемы;  - изменение конфигурации сети требует остановки всей сети |
| «Звезда» | - легко производить монтаж сети или модифицировать сеть, добавляя новые PC;  - централизованный контроль и управление;  - выход из строя одного PC или одного сегмента кабеля не влияет на работу всей сети | Выход из строя или отключение питания концентратора (коммутатора) выводит из строя всю сеть; большой расход кабеля |