

Системный подход и программирование

С точки зрения Системного подхода любую предметную область можно рассматривать как систему.

Системный подход предусматривает всестороннее исследование сложного объекта, при котором любая система (объект) рассматривается как совокупность взаимосвязанных компонентов, имеющая выход (цель), вход (ресурсы), связь с внешней средой, обратную связь.

Научные проблемы Системного подхода рассматриваются в Теории систем, а инженерные – в Системном анализе.

В соответствии с системным подходом при решении задачи при программировании осуществляется исследование задачи и условий её решения; разрабатываются модели процессов, явлений и объектов, определяющих задачу; разрабатывается алгоритм решения задачи и реализуется как компьютерная программа.

Понятия теории систем и системного анализа

Система – совокупность (множество) отдельных объектов и (обязательные) связи между ними, которые придают данной совокупности больше свойств, чем простая сумма свойств составляющих систему объектов при отсутствии связей между ними (С. от греч. σύνθημα, "составленный").

Теория систем (Общая теория систем) – это наука, предметом которой является разработка логико-концептуального и математического аппарата теоретического исследования объектов, представляющих собой системы.

Целью исследований в рамках этой теории является изучение:

- различных видов и типов систем;
- основных принципов и закономерностей поведения систем;
- функционирования и развития систем.

Системный анализ - совокупность понятий, методов, процедур и технологий для изучения, описания, реализации явлений и процессов различной природы и характера, междисциплинарных проблем; это совокупность общих законов, методов, приемов инженерного исследования таких систем.

Цель системного анализа:

- ◆ полнее понять механизмы функционирования систем;
- ◆ повысить эффективность функционирования систем;
- ◆ получить систематические, планомерно-поэтапные методы (способы) анализа и создания (построения) систем.

Понятия теории систем и системного анализа

Задачи Теории систем и Системного анализа:

- **Декомпозиция** – представление системы в виде подсистем и элементов.
- **Анализ** – определение свойств системы и среды её окружающей.
- **Синтез** – построение системы, обладающей заданными свойствами.
- **Оптимизация** – нахождение систем с экстремальными значениями её функционалов.

Некоторые понятия:

Система – это некоторая совокупность взаимосвязанных объектов (компонентов системы), обладающая свойствами, не сводящимися к свойствам отдельных объектов.

Подсистема – это часть системы со связями и отношениями, любая система состоит из подсистем, подсистема любой системы может быть сама рассмотрена как система (иерархическая вложенность подсистем).

Надсистема – это более крупная система, частью которой является рассматриваемая система.

Элемент – компонент нижнего иерархического уровня системы (не имеющий внутренней структуры).

Границы системы – это материальные и нематериальные ограничители, дистанцирующие систему от окружающей среды.

Место системы – место в иерархии, определяет надсистему, в которую входит данная система, системы одного уровня с которыми взаимодействует система, подсистемы входящие в данную систему.

Рамки системы – ресурсные, технологические (функциональные), географические и другие границы данной системы.

Структура – устойчивая картина взаимоотношений между компонентами системы (картина связей и их стабильностей).

Процесс – динамическое изменение системы во времени.

Функция – процесс, происходящий внутри системы и имеющий определённый результат.

Состояние – положение системы относительно других её положений.

Поведение – целеориентированная активность системы, служащая для осуществления контакта с окружающей средой.

Понятия теории систем и системного анализа

Основные принципы системного анализа:

- **Целостность** – рассмотрение системы одновременно как единое целое и как подсистему для вышестоящих уровней.
- **Иерархичность строения** – наличие множества компонентов, расположенных на основе подчинения компонентов низшего уровня компонентам высшего уровня.
- **Структуризация** – проведение анализа компонентов системы и их взаимосвязи в рамках конкретной организационной структуры. Как правило, процесс функционирования системы обусловлен не столько свойствами ее отдельных компонентов, сколько свойствами самой структуры (связями между компонентами).
- **Множественность** – использование множества математических, кибернетических и экономических моделей для описания отдельных компонентов и системы в целом.

Понятия теории систем и системного анализа

Системный анализ включает изучение и практическое использование следующих восьми аспектов:

- [АСПЕКТ: 1) Точка зрения, с которой рассматриваются предметы, явления, понятия и т.п.
2) Определенная сторона какого-л. явления, понятия и т.п.]

- системно-компонтного**, состоящего в выявлении компонентов, составляющих данную систему;
- системно-структурного**, заключающегося в выяснении внутренних связей и зависимостей между элементами данной системы и позволяющего получить представление о внутренней организации (строении) исследуемого объекта;
- системно-функционального**, предполагающего выявление функций, для выполнения которых созданы и существуют соответствующие компоненты системы;
- системно-целевого**, означающего необходимость научного определения целей исследования, их взаимной увязки между собой;
- системно-ресурсного**, заключающегося в тщательном выявлении ресурсов, требующихся для решения той или иной проблемы;
- системно-интеграционного**, состоящего в определении совокупности качественных свойств системы, обеспечивающих ее целостность и особенность;
- системно-коммуникационного**, означающего необходимость выявления внешних и внутренних (межкомпонентных) связей данной системы с другими, то есть, его связей с окружающей средой;
- системно-исторического**, позволяющего выяснить условия во времени возникновения исследуемой системы, пройденные ей этапы, современное состояние, а также возможные перспективы развития.

Понятия теории систем и системного анализа

Классификация систем (С.):

- ◆ **Физические – абстрактные.** Ф. – реальные С. (объекты, явления, процессы) и А. – отображения (модели реальных С.). Для одной реальной С. может быть построено множество моделей (объективная и субъективная реальности, реальная и виртуальная действительности).
- ◆ **Простые – сложные.** Простые – Большие – Сложные: Простые – мало элементов, немного однородных связей; Большие – много элементов, много однородных связей, Сложные – много элементов, много неоднородных связей.
Сложные С. – структурная и функциональная сложность.
Отличительные свойства сложных С.: (а) робастность, (б) неоднородные связи – структурные (в т.ч. иерархические), функциональные, каузальные (причинно-следственные, отношения истинности), казуальные (случайные);
- ◆ **Динамические – статические.** С. имеют изменяющиеся во времени процессы или нет.
- ◆ **Открытые-замкнутые.** По отношению системы к окружающей среде.
- ◆ **Естественные-искусственные-виртуальные-смешанные.**
- ◆ **С управлением – неуправляемые.**
- ◆ **Непрерывные – дискретные.**
- ◆ **Детерминированные – стохастические.**
- ◆ **и др.**

Понятия теории систем и системного анализа

Упрощенная последовательность практических действий при системном анализе некого объекта (системы):

1. Определение задач исследования (анализа) или синтеза системы.
2. Определение назначения и целей системы.
3. Декомпозиция системы: выявление подсистем и элементов.
4. Определение структуры системы (наличия и характеристик связей между компонентами системы).
5. Определение функций системы и её компонентов.
6. Определение задач решаемых системой, её подсистемами и элементами.
7. Определение принципов моделирования компонентов системы и системы в целом (в зависимости от задач исследования (анализа) или синтеза).
8. Разработка модели системы и её компонентов (в настоящее время чаще всего компьютерной программной модели).
9. Исследование модели системы (проведение с ними экспериментов).
10. Анализ результатов исследования модели системы.
- 11-а. Формирование выводов анализа, если исследуется существующая система.
- 11-б. Синтез (создание) новой системы.

Системный подход и программирование

Один из основных инструментов системного анализа при решении задач является **моделирование системы**.

Модели и моделирование

Модель – это упрощенное представление реальности. Моделью является, например, чертеж системы.

Основное свойство модели в том, что она – **семантически замкнутая абстракция системы**. Она строится для того, чтобы лучше понять разрабатываемую систему, визуализировать ее, определить структуру или поведение.

Сложные системы моделировать просто необходимо, поскольку иначе невозможно их воспринять как единое целое.

Моделирование – метод исследования систем на основе переноса изучаемых свойств системы на объекты другой природы. Главное в моделировании – **отношение подобия между объектом моделирования и его моделью**. Это один из основных методов исследования окружающей действительности.

Четыре основных принципа моделирования:

- **Выбор модели оказывает определяющее влияние на подход к решению проблемы и на то, как будет выглядеть это решение,**
- **Каждая модель может быть воплощена с разной степенью абстракции. При этом переход от одной степени абстракции к другой дает нам новую модель,**
- **Лучшими моделями являются те, которые ближе к реальности (по некоторому выбранному критерию),**
- **Системный подход рекомендует использовать совокупность нескольких моделей.**

Системный подход и программирование И+ПРГ

Модели и моделирование

Моделирование при разработке программ: собственно разработку программ можно рассматривать как последовательную трансформацию моделей, двигаясь от самых простых общих моделей к алгоритмической модели и, наконец, к компьютерной программе.

Выделяют три основных вида моделей в зависимости от их сущности. Это **физические модели** (например, макеты), **математические** (например, алгебраическое уравнение) и **информационные** (например, алгоритм или программа).

Описание модели может быть: **вербальное, аналитическое, численное, имитационное.**

Вербальные модели – словесное описание каких-либо свойств объекта.

Аналитические – строгие математические модели объекта.

Численные – основаны на получении большого числа реализаций стохастического (случайного) процесса, который формируется таким образом, чтобы его вероятностные характеристики совпадали с аналогичными величинами решаемой задачи.

Имитационные – описание моделируемого объекта в виде алгоритма его функционирования. Такие модели описывают процессы так, как они проходили бы в действительности, её можно "проиграть" во времени как для одного испытания, так и заданного их множества. При этом результаты будут определяться случайным характером процессов. По этим данным можно получить достаточно устойчивую статистику.

После создания модели можно разработать алгоритм решения задачи. 9

АЛГОРИТМЫ. Ориентировка

Смысловая ориентировка (зачем делать) –

Алгоритм предназначен для структуризации решаемой задачи и описания процедуры пошагового решения задачи

Функционально-целевая ориентировка (что делать) –

Провести системный анализ задачи:

- выделить цели поставленной задачи
- определить ее место и рамки в задаче верхнего уровня
- разбить задачу на слабосвязанные блоки (подзадачи)
- определить последовательность выполнения действий (шагов решения задачи)
- определить правила взаимодействия блоков (подзадач)

Исполнительская ориентировка (как делать) –

Выполнить следующие действия:

- определить цель решения задачи как ожидаемый результат
- определить виды и параметры связи задачи с задачами верхнего и параллельного уровня
- определить виды и параметры связи подзадач между собой
- определить содержание каждого шага решения задачи

АЛГОРИТМЫ

Простейший алгоритм – это последовательность действий, которые необходимо выполнить, чтобы решить поставленную задачу за конечное число шагов.

Свойства алгоритма:

- он состоит из отдельных шагов (операций)
- каждый шаг выполняет обработку входных данных (аргументов) для получения выходных данных (результатов)
- результат каждого шага однозначно определяется аргументами
- количество шагов конечно
- шаги алгоритма выполняются последовательно, в порядке следования – естественный порядок выполнения
- существуют способы изменения естественного порядка выполнения – управляющие конструкции

Способы описания алгоритмов:

- вербальное (словесное) описание
- графическая схема алгоритма
- таблица решений
- описание средствами алгоритмического языка

Постановка задачи:

- ❖ Определение совокупности *входных данных* (составление перечня),
- ❖ Определение совокупности *выходных данных* (составление перечня),
- ❖ Определение *состава и последовательности действий* для получения из совокупности входных данных совокупности выходных – **АЛГОРИТМ решения задачи,**
- ❖ Определение *методов проверки* разработанного алгоритма (тестирования).

Разработка алгоритма решения

- определение основных действий (групп действий),
- детализация действий, описание основных действий в действиях более низкого уровня – детализация выполняется до тех пор, пока каждый из шагов алгоритма не описывается как **детальная операция: способ выполнения действия с учётом внешних условий (ситуации).**

Каждый из шагов алгоритма может представлять собой:

- простое действие (операцию) по обработке данных, например, ввод/вывод или присваивание значения переменной,
- последовательность из группы операций,
- выбор из двух или более альтернатив в зависимости от истинности или ложности некоторого условия,
- циклическое повторение группы операций,
- изменение порядка выполнения шагов алгоритма.

Вербальное описание алгоритма

Вербальное описание алгоритма может быть двух видов:
свободное словесное описание и **псевдокод**.

Свободное словесное описание

ПРИМЕР 1:

Переход человека через автомобильную дорогу на нерегулируемом переходе

стоя на тротуаре посмотреть налево и, когда слева близко едет машина, то ожидать проезда машин стоя на тротуаре, смотреть налево, если слева близко нет машин, то идти до середины дороги, дойдя до середины дороги – остановиться и посмотреть направо, если справа близко едет машина, то стоять, смотреть направо, если близко машины нет, то перейти дорогу до конца.

ПРИМЕР 2:

Даны три разные числа A, B, C. Вывести их на печать в порядке убывания.

ввести три числа A, B, C
 проверить $A \neq B$, $A \neq C$, $C \neq B$
 если: $A > B$

то: $max1=A$, $max2=B$

иначе: $max1=B$, $max2=A$

если: $max1 > C$

то:

если: $max2 > C$

то: $max3=C$

иначе: $max3=max2$

$max2=C$

иначе: $max3=max2$

$max2=max1$

$max1=C$

вывести на печать $max1$, $max2$, $max3$

Вербальное описание алгоритма

Псевдокод

Псевдокод – описание алгоритма обычным разговорным языком, но немного формализованным и сокращенным для достижения сходства с ЯП высокого уровня.

Правила записи алгоритма псевдокодом:

- действия (операции) записываются простым языком,
- каждая команда (операция) записывается отдельной строкой,
- управляющие структуры алгоритма обозначаются ключевыми словами и отступами,
- последовательности операций (команд) записываются слева направо, сверху вниз и имеют один вход и один выход,
- последовательности действий и/или операций могут быть объединены в модули, модулю можно присвоить имя.

Пример использования псевдокода для описания алгоритма "Переход человека через автомобильную дорогу на нерегулируемом переходе":

стоя на тротуаре посмотреть налево
если слева близко едет машина
то стоять,

ждать проезда машин,
 смотреть налево

если слева близко не едет машина
то идти до середины дороги

если всё

иначе идти до середины дороги

если всё

дойдя до середины дороги остановиться
 посмотреть направо

если справа близко едет машина

то стоять,

ждать проезда машин,
 смотреть направо

если справа близко не едет машина

то перейти дорогу до конца

если всё

иначе перейти дорогу до конца

если всё

Графическая схема алгоритма

Графическое представление алгоритма имеет две разновидности:
блок-схема и схема Насси-Шнайдермана.

Блок схемы

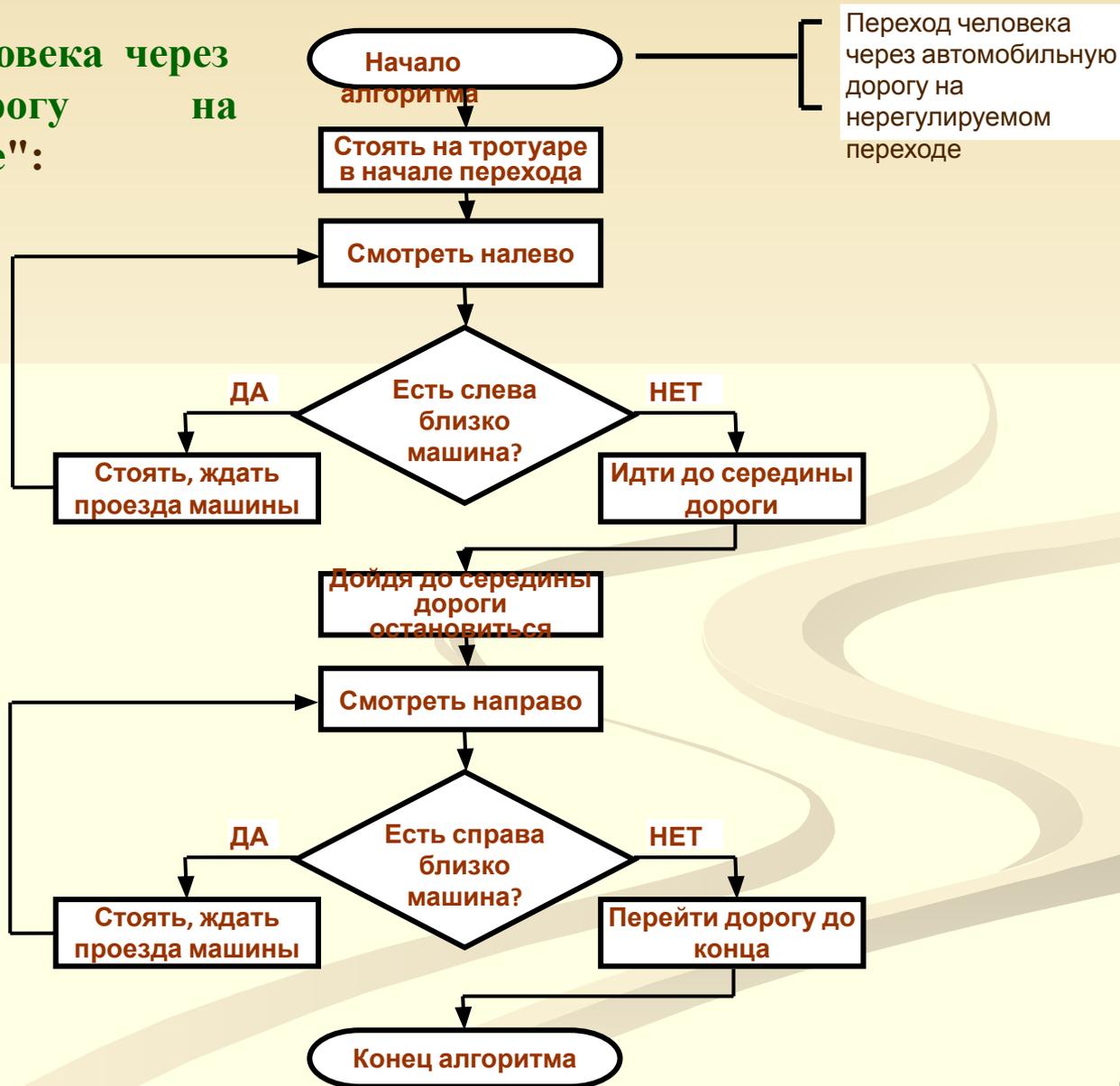
В блок-схеме условными графическими элементами и надписями внутри и рядом с ними описываются действия и операции алгоритма:

- графические элементы алгоритмов представляют базовые операции деятельности (последовательности действий, выбор конкретного действия в зависимости от условия, повторение последовательности действий в зависимости от условия – цикл и др.),**
- надписи внутри графических элементов описывают выполняемые действия и условия,**
- надписи вне графических элементов – это комментарии и результаты проверки выполнения условий (ДА или НЕТ и TRUE или FALSE).**

Графическая схема алгоритма: Блок-схемы

Алгоритм "Переход человека через автомобильную дорогу на нерегулируемом переходе":

Из приведенного примера видно, что на блок-схеме наиболее простым способом можно показывать обратные связи алгоритмов, что в других способах представления алгоритма сделать труднее.

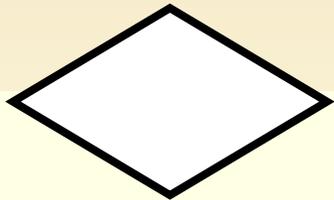


Графическая схема алгоритма

(некоторые элементы
по ГОСТ 19.003-80)



ПРОЦЕСС – выполнение операций или группы операций, в результате которых изменяется значение, форма представления или расположение данных



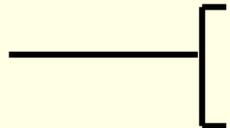
РЕШЕНИЕ – выбор направления выполнения алгоритма или программы в зависимости от некоторых переменных условий



ПУСК – ОСТАНОВ – начало, конец, прерывание процесса обработки данных или выполнения программы



ЛИНИЯ ПОТОКА – указание последовательности связей между элементами (можно использовать стрелки на концах для указания направления связей).



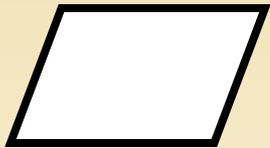
КОММЕНТАРИЙ – связь между элементом схемы и пояснением



ПРЕДОПРЕДЕЛЕННЫЙ ПРОЦЕСС (ПОДПРОГРАММА) – использование ранее созданных и отдельно описанных алгоритмов или программ

Графическая схема алгоритма

(по ГОСТ 19.003-80)



ВВОД - ВЫВОД – преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод)



ДОКУМЕНТ – ввод-вывод данных, носителем которых служит бумага



СОЕДИНИТЕЛЬ – указание связи между прерванными линиями потока внутри одной страницы



МЕЖСТРАНИЧНЫЙ СОЕДИНИТЕЛЬ – указание связи между разъединенными частями схем алгоритмов и программ, расположенных на разных листах



ПАРАЛЛЕЛЬНЫЕ ДЕЙСТВИЯ – начало или окончание двух и более одновременно выполняемых операций

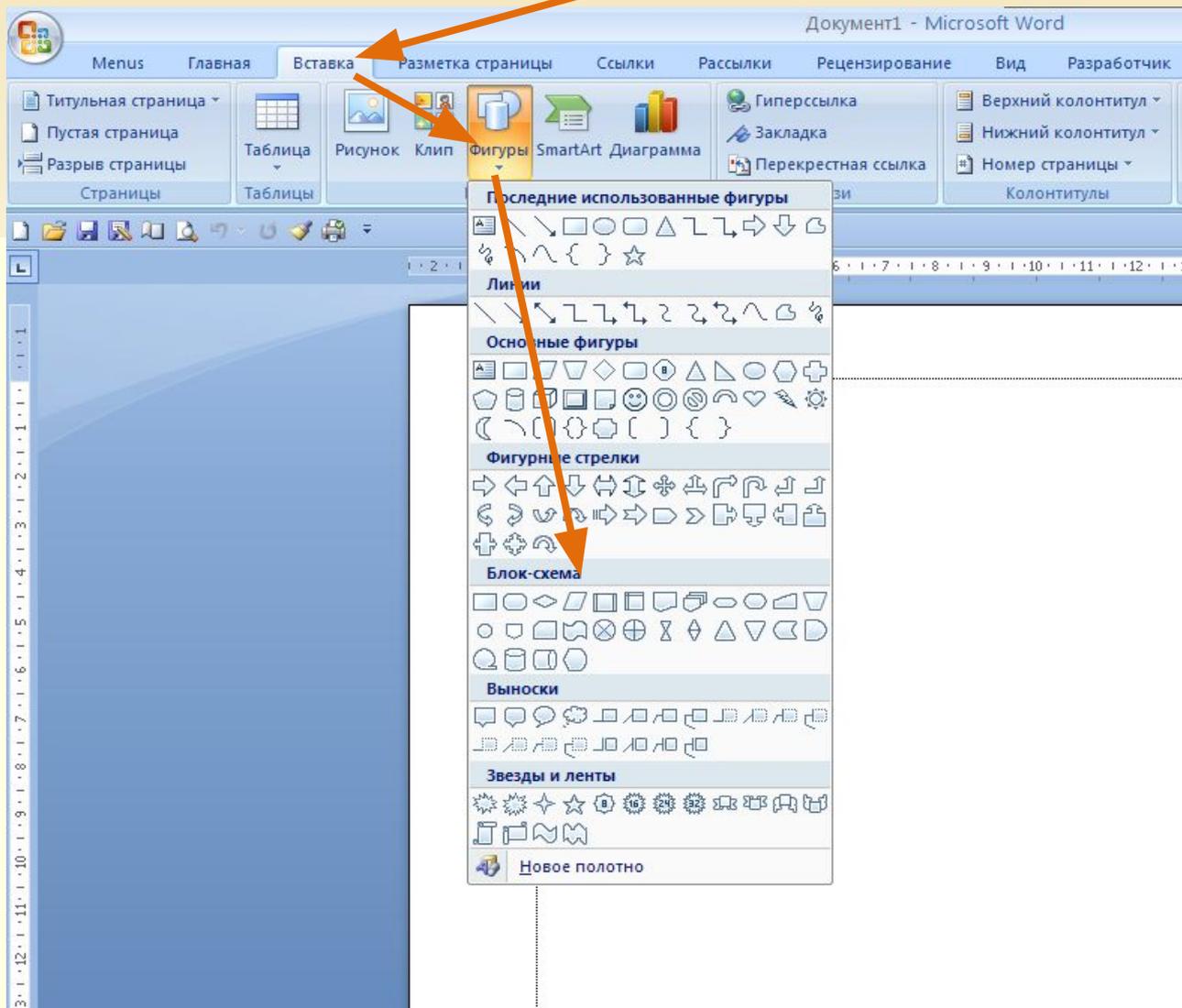


МОДИФИКАЦИЯ – выполнение операций, меняющих команды или группу команд, изменяющих программу (**используется для обозначения циклов**)

Графическая схема алгоритма

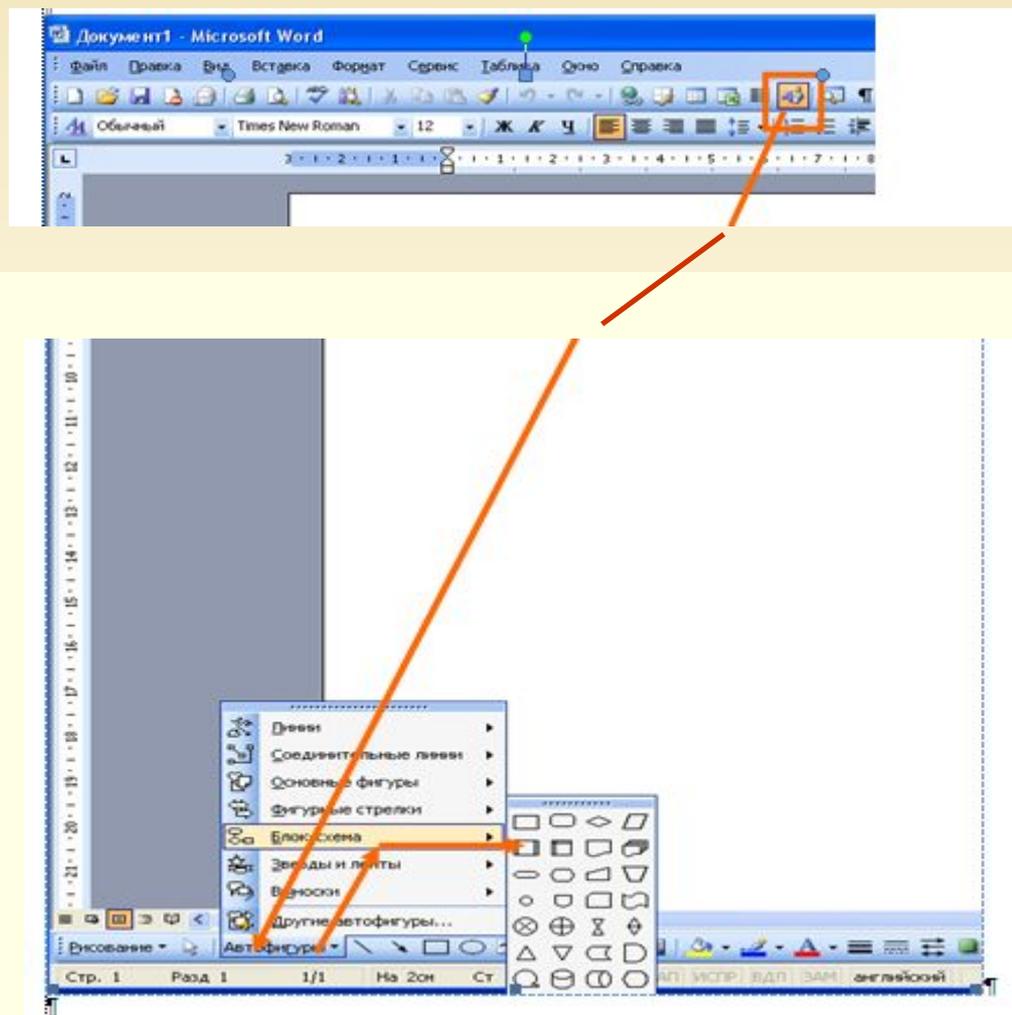
Для рисования блок-схем следует использовать специально предназначенные для этого средства Microsoft Word так и называющиеся – **Блок-схема**.

Расположение средства "**Блок-схема**" в MS Word 2007



Графическая схема алгоритма

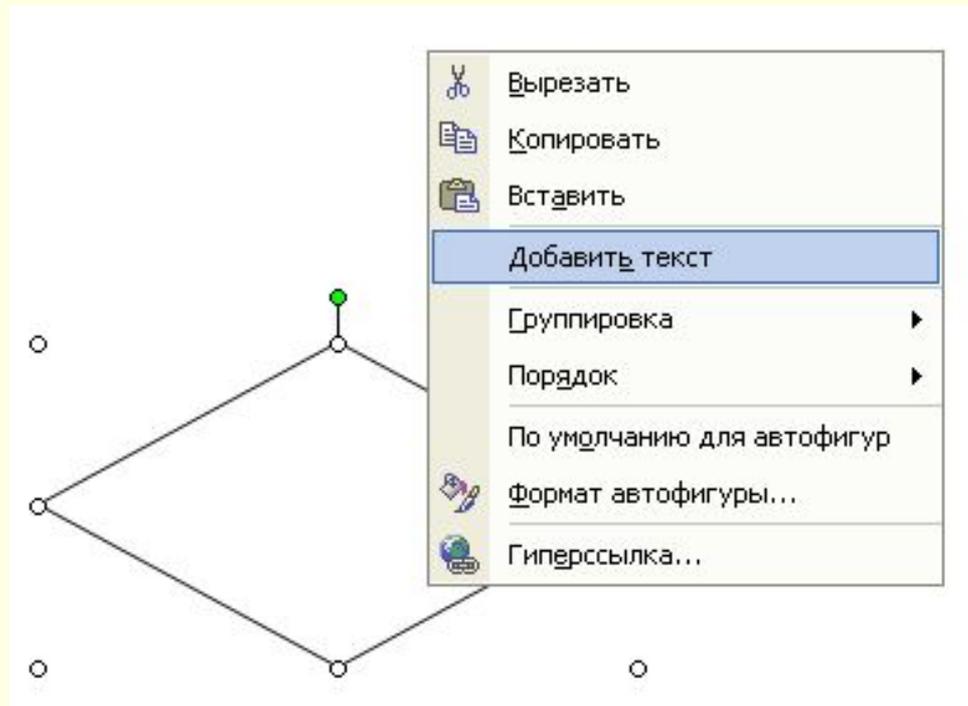
Расположение средства "Блок-схема" в MS Word 2003



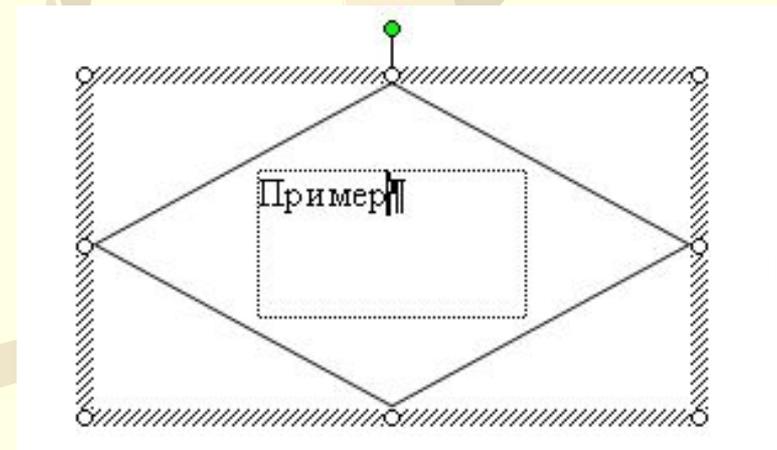
Графическая схема алгоритма

Выбрав необходимый элемент блок-схемы (автофигуру), необходимо разместить его в нужном месте страницы, затем добавить следующий элемент и связать элементы линиями потока.

Затем надо разместить пояснительный текст в элементах блок-схемы, для этого надо выделить элемент, поставить стрелку курсора на внутреннее поле элемента и нажать правую клавишу мыши и выбрать в контекстном меню пункт "Добавить текст":



Внутри элемента появится поле для ввода текста:



Графическая схема алгоритма

Если в контекстном меню выбрать поле "Формат автофигуры", то можно выполнить:

- цветную заливку внутреннего поля элемента,
- уменьшить поля отступов (снизу, сверху, справа и слева),
- увеличить толщину линий автофигур,
- сделать линии автофигур цветными,
- и многое другое.

Для того чтобы готовый алгоритм можно было перемещать по страниц без опасения разрушить взаимное расположение элементов и связей, необходимо выделить все автофигуры в алгоритме (удерживая клавишу **SHIFT**), затем установить курсор на рамку любой автофигуры, нажать правую клавишу мыши и выбрать в поле "Группировка" команду "Группировать".

Алгоритм можно считать завершённым.

Графическая схема алгоритма

Пример

Даны три разные числа A, B, C . Вывести их на печать в порядке убывания.

ввести три числа A, B, C
проверить $A \neq B, A \neq C, C \neq B$
если: $A > B$

то: $\max1 = A, \max2 = B$
иначе: $\max1 = B, \max2 = A$

если: $\max1 > C$

то:

если: $\max2 > C$

то: $\max3 = C$

иначе: $\max3 = \max2$

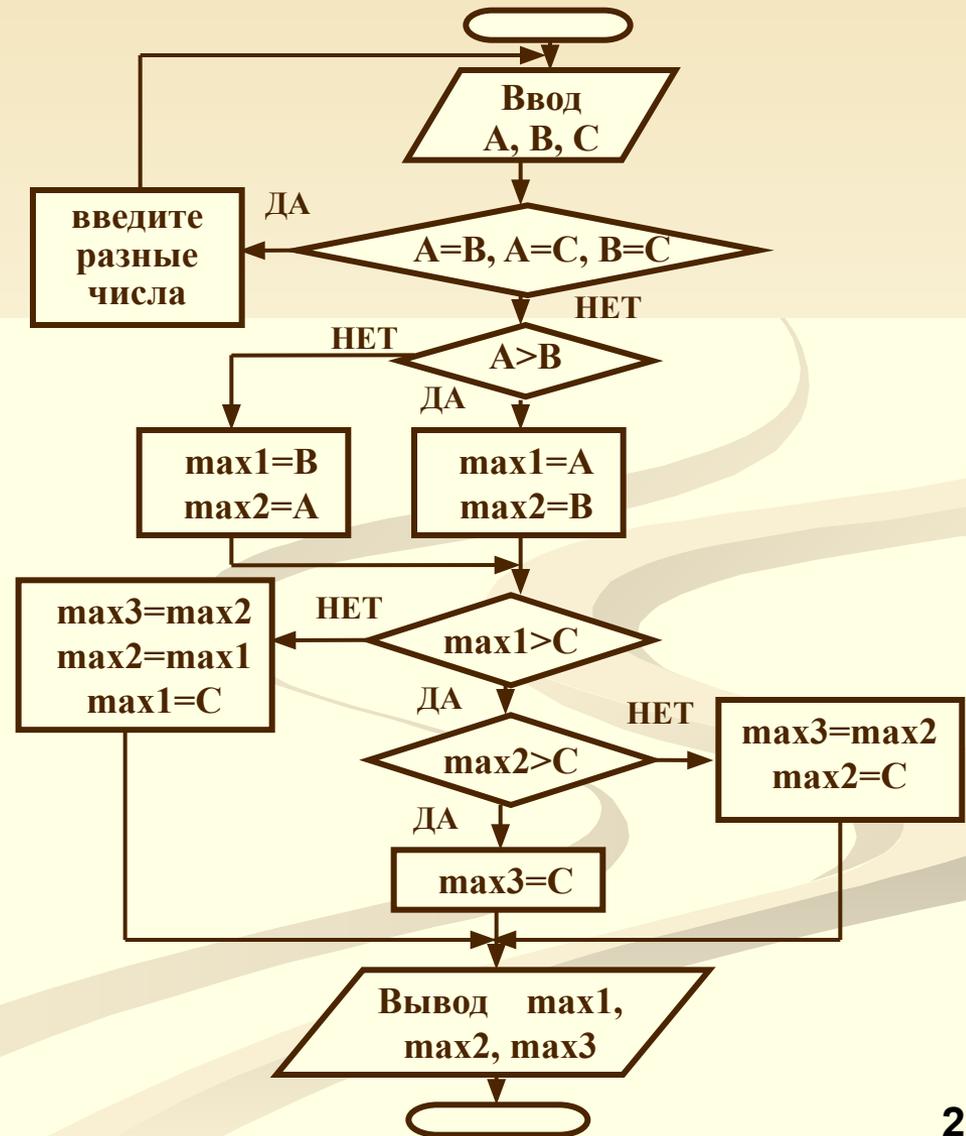
$\max2 = C$

иначе: $\max3 = \max2$

$\max2 = \max1$

$\max1 = C$

вывести на печать $\max1, \max2, \max3$



Графическая схема алгоритма

Схемы Насси-Шнейдермана

Такая схема визуально представляет собой подобный псевдокоду текст, помещённый в прямоугольные блоки, структурирующие алгоритм. Это более наглядный способ представления алгоритмов, чем псевдокод, но менее громоздкий, чем блок-схемы.

Алгоритм "Переход человека через автомобильную дорогу на нерегулируемом переходе"

Стоять на тротуаре в начале перехода

Смотреть налево

Слева близко едет машина?

ДА

НЕТ

Стоять, ждать проезда машины

Идти до середины дороги

Дойдя до середины дороги остановиться

Смотреть направо

Справа близко едет машина?

ДА

НЕТ

Стоять, ждать проезда
машины

Перейти дорогу до
конца

Таблица решений

Табличный способ изображения алгоритма – это **таблица решений**.

В таблице для каждого логического выражения показывается действие (операция). Например:

Таблица решений алгоритма "Переход человека через автомобильную дорогу на нерегулируемом переходе":

Условие	Решение (действие/операция)
Стоим в начале перехода	Смотреть налево на дорогу
Слева близко едет машина	Ожидать проезда машин
Слева не едет близко машина	Идти по переходу до середины дороги
Дошли до середины дороги	Остановиться, смотреть направо
Справа едет близко машина	Ожидать проезда машин
Справа не едет близко машина	Перейти дорогу по переходу до конца

Описание алгоритма средствами И+ПРГ языка программирования

Алгоритм описывается с помощью синтаксических конструкций какого-нибудь языка программирования.

Пример реализации алгоритма **"Вычисление факториала 25!"** на языке Си

```
#include <stdio.h>
int main ()
{
    int n, k; float p;
    printf ("Введите факториал k = ");
    scanf ("%d",&k); /* Ввод факториала */
    p = 1; n = 1;      /* Начальные значения */
    while (n <= k)
    {
        p = p * n; /* Вычисление в цикле */
        n++; /*Приращение переменной цикла*/
    }
    printf("Значение факториала  p = %G\n ",p);
    return 0;
}
```

Блок-схемы алгоритмов

ЗАДАНИЕ

Составить блок-схему алгоритма решения задач на компьютере: ввести необходимые данные, произвести требуемые вычисления, вывести результат на экран.

Задание 1. Дан радиус окружности. Вычислить длину окружности и площадь круга.

Задание 2. Составить алгоритм решения линейного уравнения $ax+b=0$, при $a \neq 0$.

Задание 3. Дано значение стороны прямоугольника. Вычислить его периметр и площадь.

Задание 4. Дано число A . Пользуясь только операцией умножения вычислить A^{10} за четыре операции.

Учесть возможность ввода неверных данных