

Методы работы с процедурной информацией

Информацию можно рассматривать не только как совокупность фактических данных, но и как динамическую величину, то есть информацию действия. В таком аспекте информация необходима для решения некоторой задачи.

• **Задача** – это проблема, подлежащая решению.

Различают технологические и функциональные задачи.

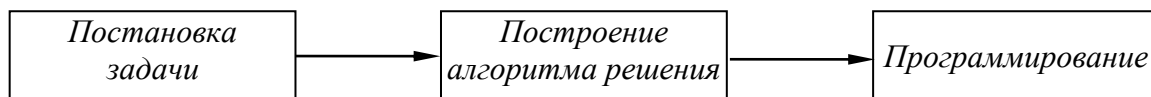
Технологические задачи – ставятся и решаются при организации технологического процесса обработки информации на компьютере. Технологические задачи являются основой для сервисных систем программного обеспечения в виде операционных систем, систем программирования, систем управления базами данных, табличных процессоров, текстовых процессоров и т.д., т.е. при разработке системного программного обеспечения.

Функциональные задачи – ставятся при реализации функций управления в рамках информационных систем предметных областей.

Например, задачи управления деятельностью предприятия или фирмы, управления поставками и перевозками продукции и т.д.

Функциональные задачи в совокупности образуют предметную область и полностью определяют ее специфику. Функциональные задачи решаются при разработке прикладного программного обеспечения.

Процесс решения задач на компьютере можно представить как последовательность трех этапов:



Постановка задачи

• **Постановка задачи** – это точная формулировка решения задачи на компьютере с описанием входной и выходной информации рассматриваемой предметной области.

Постановка задачи связана с конкретизацией основных параметров ее реализации, определением источников и структурой входной и выходной информации, востребованной пользователем.

Для облегчения решения сложной задачи ее необходимо разбивать на подзадачи.

К основным характеристикам задач относятся:

- цель решения задачи, ее место и связи с другими задачами;
- условия решения задачи с использованием средств вычислительной техники;
- требования к периодичности решения задачи;
- ограничения по срокам и точности выходной информации;
- состав и форма представления выходной информации;
- источники входной информации для решения задачи;
- пользователи задачи (кто ее решает и пользуется результатами).

Этап постановки задачи изначально является главным этапом проектирования любой программы, т.к. от того, как будет определена цель и выбрана модель описания предметной области, зависит правильное решение исходной задачи.

Моделирование информации

Для успешной организации информационных процессов с помощью технических и программных средств нужно представить информацию о реальном объекте в таком виде, чтобы ее можно было ввести в информационную систему, обработать, при необходимости сохранить и выдать по первому требованию пользователя. Для этого информацию необходимо смоделировать.

Введем несколько определений.

- **Модель** - это формальное (как правило, приближенное) описание изучаемого объекта или явления, отражающее интересующие нас аспекты.

- **Математическая модель** (формулы + методы + правила) является основой построения информационной модели.

- **Информационная модель** задачи подразумевает определение вида, структуры, последовательности обработки и подачи на экран информации, необходимой для работы программного продукта.

- **Входная информация** определяется как данные, поступающие на вход задачи и используемые для ее решения. Входной информацией служат рукописные документы, файлы базы данных, нормативно-справочная информация, входные сигналы от датчиков устройств и т.д.

- **Выходная информация** по задаче может быть представлена в виде документов, выданных на принтер, файла базы данных, графика на экране дисплея, управляющего сигнала и т.д.

- **Предметная область** (ПрО) - совокупность связанных между собой функций, задач управления, с помощью которых достигается выполнение поставленных целей. Любая предметная область имеет границы.

- **Модель данных (МД)** - это используемая знаковая система (способ абстрагирования предметной области) для обозначения структур данных и операций их обработки.

- **Концептуальная схема ПрО** - описание предметной области в терминах выбранной модели данных.

Распространённой моделью данных является устная или письменная речь. Концептуальная схема ПрО в этом случае выражается в виде совокупности текстов и/или фонограмм. Однако тексты и фонограммы плохо приспособлены для моделирования информации, имеющей сложную структуру. В этом случае используют модели данных, располагающие удобными средствами описания информационной структуры. Существуют различные модели данных для описания информационных структур (иерархические, сетевые, реляционные и т.д.).

Для описания ПрО используют три основных конструктивных элемента - сущность, атрибут и связь

• **Сущность** - это обобщённое понятие для обозначения множества однородных объектов ПрО, информацию о которых необходимо собирать и хранить в информационной системе. Сущность определяется своим уникальным именем и перечнем атрибутов, характеризующих свойства сущности.

• **Атрибут** - это поименованная характеристика сущности, которая принимает значения из некоторого множества допустимых значений. Атрибуты моделируют свойства сущности.

Чтобы задать атрибут в модели, необходимо присвоить ему наименование, определить множество его допустимых значений и указать, для чего он используется. Имя атрибута связывают с его смысловым описанием. В этой связи совокупность имен атрибутов описывает свойства сущности.

• **Связь** - это обобщённое понятие, предназначенное для обозначения выделенного в ПрО отношения между двумя или более сущностями. Как и сущности, каждая категория пользователей выделяет связи в соответствии со своей концепцией ПрО.

Пример. Выделить информационные объекты учебного процесса.

Описывая предметную область «**Учебный процесс**», можно выделить несколько сущностей: **студент, преподаватель, предмет**.

Сущность **Студент** можно охарактеризовать следующими атрибутами:

ФИО, ДАТА_РОЖДЕНИЯ, АДРЕС, ДАТА_ПОСТУПЛЕНИЯ, НОМЕР_ЗАЧЕТНОЙ_КНИЖКИ, НОМЕР_ГРУППЫ.

Сущность **Преподаватель** характеризуется следующими атрибутами:

ФИО, АДРЕС, ДОЛЖНОСТЬ, КАФЕДРА, УЧЕНОЕ_ЗВАНИЕ, ТЕЛЕФОН_РАБОЧИЙ, ТЕЛЕФОН_ДОМАШНИЙ.

Сущность **Предмет** характеризуется следующими атрибутами:

НАЗВАНИЕ, КАФЕДРА, ЛЕКЦИИ, ПРАКТИЧЕСКИЕ_ЗАНЯТИЯ, ЛАБОРАТОРНЫЕ_РАБОТЫ, КУРСОВЫЕ_РАБОТЫ, ОТЧЕТНОСТЬ, СЕМЕСТР.

Между сущностями **Студент, Преподаватель** и **Предмет** существует связь **Экзамен**. Связь, как и сущность, может иметь атрибуты. Связь экзамен характеризуется, например, атрибутами **ОЦЕНКА** и **ДАТА_ПРОВЕДЕНИЯ**.

Предметная область определена, если известны существующие в ней объекты, их свойства и отношения (связи).

При описании той или иной ПрО желательно, чтобы соблюдались следующие требования:

- полнота охвата объектов (сущностей) рассматриваемой области;
- однозначность атрибутов;
- возможность включения новых объектов (сущностей).

Построение алгоритма решения задачи

После разработки модели и определения применяемых математических методов строится алгоритм, т.е. строгая последовательность действий по решению поставленной задачи.

• **Алгоритм** – система точно сформулированных правил, однозначно определяющая процесс преобразования допустимых исходных данных (входной информации) в желаемый результат (выходную информацию) за конечное число шагов.

Алгоритм включает в себя также порядок диалога с пользователем, порядок использования устройств машины и т.д. При этом алгоритм не должен быть привязан к конкретному языку программирования, т.е. по нему можно написать программу на любом языке программирования.

Алгоритм должен обладать следующими свойствами:

• **дискретностью** – разбиением процесса обработки информации на более простые этапы (шаги), выполнение которых компьютером или человеком не вызывает затруднений;

• **определенностью** (детерминированностью) – однозначностью получаемого результата при одних и тех же исходных данных;

• **результативностью** – обязательным получением желаемого результата за конечное число шагов при допустимых исходных данных;

• **массовостью** – применимостью алгоритма для решения определенного класса задач.

В алгоритме отражаются логика и способ формирования результатов решения с указанием необходимых расчетных формул, логических условий, соотношений для контроля достоверности выходных результатов. В алгоритме обязательно должны быть предусмотрены все ситуации, которые могут возникнуть в процессе решения задачи. В зависимости от степени детализации, поставленных целей, методов и технических средств решения задачи используются различные способы представления алгоритмов. На практике наиболее распространены следующие формы представления алгоритмов: содержательная (текстуальная) форма; графическая форма (блок-схема); программная (на языках программирования компьютера).

Остановимся подробнее на графическом представлении алгоритма.

Графическая форма представления алгоритмов

Является более компактной и наглядной по сравнению с текстовой формой. Алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий. Такое графическое представление называется блок-схемой алгоритма.

Основные условные обозначения функциональных блоков, принятые в схемах алгоритмов:

<i>Графическое обозначение</i>	<i>Наименование</i>	<i>Пояснения</i>
	Пуск-останов	Начало, конец, прерывание процесса обработки данных.
	Процесс, действие	Операция, в результате которой изменяется значение данных.
	Условие	Разветвление алгоритма в зависимости от некоторых условий.
	Модификация	Начало циклической операции.
	Программа, подпрограмма	Часть алгоритма, требующая дополнительной детализации на последующих шагах.
	Ввод-вывод данных	Ввод-вывод без указаний конкретного носителя.
	Документ	Ввод-вывод данных, носителем которых является бумага.
	Дисплей	Ввод-вывод данных на дисплей.
	Магнитный диск	Ввод-вывод данных на магнитный диск.
	Комментарий	Связь между элементом схемы и пояснением к нему.
	Соединители	Связь между прерванными линиями на одной странице, связь между прерванными частями схем на разных стр.

Пример различных форм алгоритма

Задача: Даны три числа $A1$, $A2$, $A3$. Найти максимальное число A_{\max} .

Содержательная форма представления алгоритма

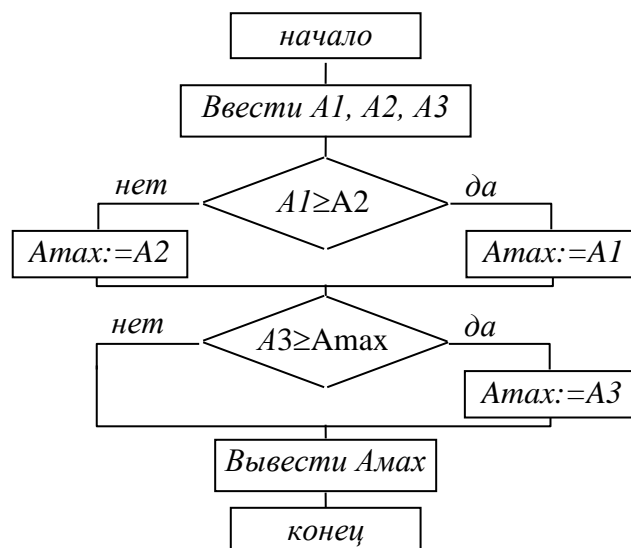
Шаг 1. Сравнить значения чисел $A1$ и $A2$; если $|A1| \geq |A2|$, то числу A_{\max} присвоить значение $|A1|$, иначе числу A_{\max} присвоить значение $|A2|$.

Шаг 2. Сравнить значения чисел $A3$ и A_{\max} . Если $|A3| \geq |A_{\max}|$, то числу A_{\max} присвоить значение $|A3|$.

Шаг 3. Конец алгоритма.

Приведенное описание алгоритма является достаточно строгим и позволяет однозначно решить поставленную задачу. Однако, для достаточно сложных алгоритмов описание становится слишком громоздким и ненаглядным. Поэтому содержательную форму используют на начальных стадиях разработки алгоритмов, когда определяются основные этапы решения поставленной задачи.

Графическая форма представления алгоритма



Программная форма представления алгоритма

```
Program max_3;  
Var A1,A2,A3,Amax:integer;  
Begin  
  Writeln('Введите три числа');  
  Readln(A1,A2,A3);  
  if A1>=A2 then Amax:=A1  
    else Amax:=A2;  
  if A3>=Amax then Amax:=A3;  
  Writeln('Максимальное число ',Amax);  
End.
```

Программирование решения задачи

Алгоритм и его программная реализация тесно взаимосвязаны.

- **Программа** – упорядоченная последовательность инструкций компьютера (команд) для решения задачи.

- **Программное обеспечение** – совокупность программ обработки данных и необходимых для их эксплуатации документов.

- **Программирование** – теоретическая и практическая деятельность, связанная с созданием программ.

Программирование является собирательным понятием и может рассматриваться и как наука, и как искусство; на этом основан научно-практический подход к разработке программ.

Программа – результат интеллектуального труда, для которого характерно творчество, поэтому в любой программе присутствует индивидуальность ее разработчика, программа отражает определенную степень искусства программиста. Вместе с тем программирование предполагает и рутинные работы, которые могут и должны иметь строгий регламент выполнения и соответствовать стандартам.

Программирование базируется на комплексе научных дисциплин, направленных на исследование, разработку и применение методов и средств разработки программ. При разработке программ используются ресурсоемкие и наукоемкие технологии, высококвалифицированный интеллектуальный труд. По некоторым данным в середине 90-х годов в мире было занято программированием $\approx 2\%$ трудоспособного населения. Совокупный оборот в сфере создания программных средств достигает нескольких сот миллиардов долларов в год. В связи с этим весьма актуальным становится вопрос о разработке и применении эффективных технологий программирования.

Стадии разработки программного продукта

Этот этап представляет собой выполнение четырех взаимосвязанных задач:

- ◆ реализация алгоритма на языке программирования;
- ◆ отладка программы;
- ◆ тестирование программы;
- ◆ модификация и сопровождение программы.

- **Написание программы** - запись разработанного алгоритма на каком-либо языке программирования.

- **Отладка.** На этом этапе с помощью специальной программы (транслятора) в программе устраняются *синтаксические* ошибки (связанные с неправильной записью или употреблением языковых конструкций).

- **Тестирование.** На этом этапе устраняются *семантические* (смысловые) ошибки. Необходимо проводить тестирование в два этапа: на заведомо правильных, а затем на заведомо неправильных данных. В любом случае программа должна выдавать соответствующие результаты. Как правило, в программе должны быть отражены все особые случаи. Для сложных алгоритмов такая информация, как правило, неизвестна.

- **Модификация.** Программа постоянно модифицируется как на этапе разработки, так и в период опытной эксплуатации. Программы, рассчитанные на долгий срок применения, требуют периодического усовершенствования. Это может быть связано с дополнительными требованиями к ее возможностям, сменой оборудования и т.п. Поэтому необходимо, чтобы программа предоставляла возможность своей модификации без полной повторной реализации.

- **Сопровождение программного продукта** - исправление обнаруженных ошибок, а также поддержка работоспособности программного продукта при переходе на новые версии и внесении изменений.

Схема взаимодействия специалистов, связанных с созданием и эксплуатацией программ

Основная категория специалистов, занятых разработкой программ – это программисты. Программисты неоднородны по уровню квалификации, а также по характеру своей деятельности. Наиболее часто программисты делятся на системных и прикладных.

Системный программист занимается разработкой, эксплуатацией и сопровождением системного программного обеспечения, поддерживающего работоспособность компьютера и создающего сферу для выполнения программ, обеспечивающих реализацию функциональных задач.

Прикладной программист осуществляет разработку и отладку программ для решения функциональных задач.

В процессе создания программ на начальной стадии работ участвуют и специалисты – постановщики задач. При создании больших по масштабам и функциям обработки программ (например, операционных систем) – нужен программист-аналитик для анализа и проектирования комплекса взаимосвязанных программ.

Большинство информационных систем основано на работе с базами данных (БД). Если база данных является сложной и используется многими программистами, то возникает проблема организационной поддержки БД, которая выполняется администратором БД.

Основным потребителем программ служит конечный пользователь, который относится к категории пользователей-непрограммистов. Он должен иметь элементарные знания и навыки работы с компьютером.

