

**\*Технологический  
цикл обработки  
информации на  
компьютере**

Лекция 2

Работа по решению прикладной задачи на компьютере проходит через следующие этапы:

- постановка задачи;
- анализ и исследование задачи, модели;
- разработка алгоритма;
- составление программы на языке программирования;
- тестирование и отладка программы;
- проведение расчетов и анализ полученных результатов.

Эту последовательность называют технологической цепочкой решения задачи на компьютере.

# \* Постановка задачи

На этапе постановки задачи должно быть четко определено, что дано, и что требуется найти. Если задача конкретная, то под постановкой задачи понимают ответ на два вопроса: какие исходные данные известны и что требуется определить. Если задача обобщенная, то при постановке задачи понадобится еще ответ на третий вопрос: какие данные допустимы. Таким образом, постановка задачи включает в себя следующие моменты:

- сбор информации о задаче;
- формулировку условия задачи;
- определение конечных целей решения задачи;
- определение формы выдачи результатов;
- описание данных (их типов, диапазонов величин,

# \* Анализ и исследование задачи, модели

Этот этап включает:

- анализ существующих аналогов;
- анализ технических и программных средств;
- разработка математической модели;
- разработка структур данных.

Математическая модель - это система математических соотношений: формул, уравнений, неравенств и т. д., отражающих существенные свойства объекта или явления. Необходимо отметить, что при построении математических моделей далеко не всегда удастся найти формулы, явно выражающие искомые величины через данные. В таких случаях используются математические методы,

# \* Разработка алгоритма

Разработка алгоритма включает в себя

- выбор метода проектирования алгоритма;
- выбор формы записи алгоритма;
- выбор тестов и метода тестирования;
- проектирование самого алгоритма.

# \* Программирование

Программирование включает в себя следующие виды работ:

- выбор языка программирования;
- уточнение способов организации данных;
- запись алгоритма на выбранном языке программирования.

# \*Отладка и тестирование программы

Отладка программы – это процесс поиска и устранения ошибок в программе, проводимый по результатам ее прогона на компьютере. При отладке происходит локализация и устранение ошибок.

Тестирование – это проверка правильности работы программы в целом или ее составных частей. Тест - это конкретный вариант значений исходных данных, для которого известен ожидаемый результат.

Т.е. тестирование устанавливает факт наличия ошибок, а отладка выясняет ее причину.

# \* Методы отладки

Методы отладки можно разделить на методы "грубой силы" и методы, основанные на обдумывании.

Методы "грубой силы" не являются самыми эффективными, но очень популярны, т.к. не требуют значительного внимания и больших умственных затрат. К ним относятся:

- Отладка в соответствии с общим предложением "расставить операторы печати по всей программе"
- Отладка с использованием автоматических средств

Методы, основанные на обдумывании:



*Метод индукции.* Индукция - это анализ от частного к целому. Просматривая ошибки, установленные тестами и взаимосвязи между ними, можно обнаружить причину ошибки.

*Метод дедукции.* Данный метод позволяет на основе некоторых предпосылок, используя операторы исключения или уточнения, обнаружить место ошибки. Необходимо просмотреть результаты всех тестов об ошибках. Выдвинутые гипотезы поочередно исключаются из рассмотрения.

*Прослеживание логики в обратном порядке.* Отладка начинается в точке программы, где был обнаружен некоторый неверный результат. Мысленно выполняя из данной точки программы возврат в обратном порядке и рассуждая так: "Если бы в этой точке состояние программы было таким, то в другой точке должно быть следующее состояние", можно быстро и точно локализовать ошибку.

# \*Типы ошибок

Различают следующие типы ошибок:

- синтаксические ошибки (ошибки компиляции);
- ошибки выполнения;
- ошибки в алгоритме программы или его реализации (семантические).

*Синтаксические ошибки* возникают при нарушении правил языка. Их обнаруживает компилятор, который и выдает сообщение об ошибке, обычно устанавливая курсор в позицию, где обнаружена ошибка.

Семантические ошибки должен обнаруживать сам программист, проверяя работу программы на заранее подготовленной *системе тестов*, для которых заранее известны результаты. Подготовка такой системы тестов сама по себе представляет серьезную задачу.

Для облегчения поиска семантических ошибок используются специальные программы, называемые *отладчиками*. Отладчик позволяет выполнять программу по шагам, просматривая при этом промежуточные результаты каждого шага. Этот процесс называется *трассировкой*. Если на очередном шаге результат оказывается не таким, на какой рассчитывал разработчик, значит, ошибка локализована. Осталось только понять ее причину и исправить.

*Ошибки выполнения (run-time errors)* не нарушают синтаксис языка. Однако они приводят к ошибочным операциям в процессе выполнения программы, таким например, как попытка деления на ноль или извлечение квадратного корня из отрицательного числа. При возникновении такой ошибки процессор вынужден выполнить аварийную остановку программы, на экран при этом выводится сообщение **Run-time error** и сообщается код ошибки, по которому можно установить ее причину.

*Семантические ошибки* программы при верных исходных данных и внешне безошибочной работе программы приводят к неверным результатам. Этот тип ошибок наиболее коварен и труден для исправления, т. к. пользователь, получая ошибочный результат, может посчитать его верным, поскольку никаких сообщений об ошибках не было.

Таким образом, тестирование и отладка включают в себя

- синтаксическую отладку;
- отладку семантики и логической структуры программы;
- тестовые расчеты и анализ результатов тестирования;
- совершенствование программы.

# \* Методы тестирования

- Метод "черного ящика". Основан на принципе "вход-выход". Программе подаются некоторые данные на вход и проверяются результаты, в надежде найти несоответствия. При этом как именно работает программа, считается несущественным.
- Метод "белого ящика". В этом методе тестовые данные получают путем анализа логики программы.

# \* Этапы процесса тестирования

1. **Проверка в нормальных условиях.** Предполагает тестирование на основе данных, которые характерны для реальных условий функционирования системы.
2. **Проверка в экстремальных условиях.** Тестовые данные включают граничные значения области изменения входных переменных, граничные объемы данных.
3. **Проверка в исключительных ситуациях.** Проводится с использованием данных, значения которых лежат за пределами допустимой области. Программа должна отвергать любые данные, которые она не в состоянии обработать правильно. Наихудшая ситуация возникает тогда, когда программа принимает данные, выходящие за допустимые пределы и выдает неверный результат.

# \* Характерные ошибки программирования

**Неправильная постановка задачи.** Неверно трактуется условие задачи.

**Неверный алгоритм.** Выбран алгоритм, приводящий к неточному или неэффективному решению задачи.

**Ошибка анализа.** Неполный учет всех ситуаций, которые могут возникнуть.

**Синтаксические ошибки.** Нарушение правил, определяемых языком программирования.

**Семантические ошибки.** Возникают при незнании порядка выполнения операторов.

**Ошибки выполнения.** Недопустимые операции.



**Ошибки в данных.** Неудачное определение возможного диапазона изменения данных.

**Опечатки.** Перепутаны близкие по написанию символы.

**Ошибки ввода-вывода.**

# \* Проведение расчетов и анализ полученных результатов

Последний этап - это использование уже разработанной программы для получения искомых результатов. Производится анализ результатов решения задачи и в случае необходимости - уточнение математической модели (с последующей корректировкой алгоритма и программы). Программы, имеющие большое практическое или научное значение, используются длительное время. Иногда даже в процессе эксплуатации программы могут исправляться, дорабатываться.

# \*Сопровождение программы

- Доработка программы для решения конкретных задач;
- Составление документации к решенной задаче, к математической модели, к алгоритму, к программе, к набору тестов, к использованию.

# \*Единая система программной документации

Единая система программной документации (ЕСПД) — комплекс государственных стандартов, устанавливающих взаимосвязанные правила разработки, оформления и обращения программ и программной документации.

Стандарты ЕСПД обеспечивают возможность:

- унификации программных изделий для взаимного обмена программами и применения ранее разработанных программ в новых разработках;
- снижения трудоемкости и повышения эффективности разработки, сопровождения, изготовления и эксплуатации программных изделий;
- автоматизации изготовления и хранения программной документации.

# \* Домашнее задание

1. Составить опорный конспект лекции по теме «Технологический цикл обработки информации на компьютере» на основе презентации.
2. Проектирование программного обеспечения экономических информационных систем. Вендров А.М. М.: Финансы и статистика, 2000, с. 7-55