

Вопрос № 1

Структурное программирование при разработке программ

На протяжении 70х годов произошел переход с ламповой элементарной базы построения ЭВМ на полупроводниковую, а затем, на базу интегральных схем. Это было связано с повышением ее компактности, удешевлением, серьезным повышением эффективности. В результате этого перехода потребовались сложные программные комплексы для решения новых проблем и задач. Появилась проблема, которая выражалась в том, что используемые ранее методы конструкторского программирования оказались неэффективными. С появлением необходимости разработки программируемых систем резко встает вопрос о повышении производительности труда программиста, об облегчении этапов отладки и тестирования программ, о понимании большими коллективами программистов не только своих участков работы, но и всей решаемой проблемы, а также облегчении решения этапа объединения отдельных частей в большой комплекс.

Одним из пионеров новой методики стал Дейкстра, в 1968 году заявил о необходимости исключить из состава языков программирования высокого уровня оператора goto. Позднее была сформулирована четкая система правил, названная структурным программированием.

Эта система представляет собой ряд ограничений и правил, которые обеспечивают соответствие любой программы строгому образцу, исключая запутанность, порождающую ошибки и затрудняющую тестирование. Одним из главных вопросов. Побудивших Дейкстру к новой методике стала необходимость облегчения тестирования программных комплексов в наиболее полном объеме. Необходимость в этой процедуре вызвана следующим:

- 1) трудоемкость и стоимость тестирования больших программных комплексов возрастает экспоненциально с увеличением их размеров. При этом есть мнение, что стоимость проверки любого изменения в сложном комплексе более чем в сто раз превышает стоимость внесения этого изменения.
- 2) В сложных системах, требующих постоянного сопровождения и развития, ошибки присутствуют всегда
- 3) Издержки от испытания сложных комплексов постоянно возрастают, по мере того, как люди возлагают на ЭВМ все более ответственные функции.

Одним из выходов может стать разработка сложных и дорогостоящих методов тестирования программ (их эффективность является спорной). Дейкстра заложил в своей методике автоматическое доказательство правильности создаваемой программы, то есть требование правильности программы выполняется на любом этапе ее конструирования. Структурное программирование обладает тем дополнительным преимуществом, что повышает читаемость программы.

Поведение неструктурированной программы сравнивают с броуновским движением. Структурированная программа обладает структурной организацией и исполнением. Такие программы читаются последовательно, так как в них не содержится передача управления.

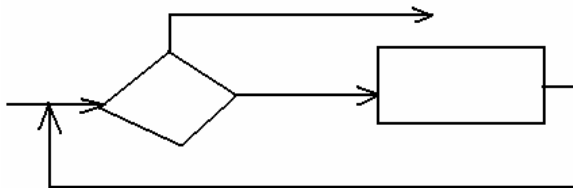
Помимо всего прочего было доказано, что разработка структурированной программы на языках высокого уровня производится гораздо эффективнее. После разработки идей структурного программирования появилась возможность разрабатывать компиляторы с глобальной оптимизацией кода, так как анализ в таких программах производится автоматически.

В области практического применения структурного программирования основные исследования были направлены на решение проблемы разработки программы таким способом, чтобы ее правильность подавалась доказательству. Все усилия прилагались для анализа программы, организованной по нисходящей схеме, где вся программа рассматривалась как независимый вызываемый модуль. На следующем этапе исходная программа разбивается на комбинацию модулей 1го уровня. Общий процесс декомпозиции продолжается до тех пор, пока проектировщик не получит легкокодируемые модули. Необходимость легкого тестирования любой структуры $k+1$ го независимо от смысла, вкладываемого в не на каком уровне декомпозиции, привела к требованию, чтобы любой модуль программы изначально проектируемой с одним входом и выходом, что приводит к представлению о программе, как о множестве вложенных друг в друга модулей. В классических работах по структурному программированию доказано, что такая реализация программы может быть воплощена с использованием всего двух управляющих конструкций, конкретная реализация которых может быть различной в разных языках программирования. Этот базовый принцип часто называют структурной теоремой. Согласно ей для построения программы необходимы всего три функциональных блока:

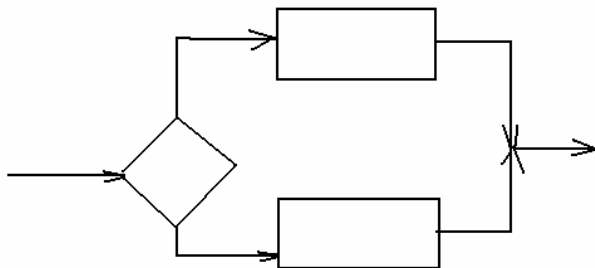
1) представляет собой отдельный оператор или логически связанную последовательность операторов, имеющую один вход и выход.



2) конструкция обобщенного цикла



3) конструкция двоичного решения



Вопрос № 2

Понятие критерия оптимального проектирования и его связь с варьируемыми переменными через уравнения математической модели. Постановка задачи оптимального проектирования.

Задачей проектирования новых объектов на современном этапе является получение не просто работоспособных, но и оптимальных с точки зрения заданного критерия объекта.

Для постановки и решения задачи оптимального проектирования необходимо два фактора:

1. Должны быть варьируемые переменные, изменяя которые проектировщик получает различные проектные решения
2. Критерий, по которому сравниваются проектные решения

Для связи варьируемых переменных с критерием используется математическая модель (ММ).

Математическая модель – это система булевских, алгебраических, дифференциальных, интегральных уравнений, связывающих варьируемые переменные с критерием.

В качестве критериев могут использоваться:

1. Технологические переменные: производительность, концентрация на выходе, температура на выходе и т.д.
2. Конструкционные характеристики: объем, длина и др. габаритные размеры.
3. Техничко-экономические показатели: себестоимость, прибыль, технологическая себестоимость и др.

В случае отсутствия варьируемых переменных задача оптимизации вырождается в задачу расчета.

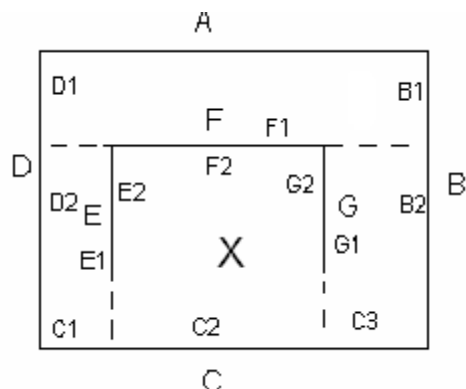
Постановка задача:

Найти значения варьируемых переменных X_1^* , X_2^* , ..., X_n^* , при которых критерий достигает экстремального значения при выполнении ограничений, наложенные на входные и выходные координаты, и выполнении связей, заданных в виде уравнений мат. модели.

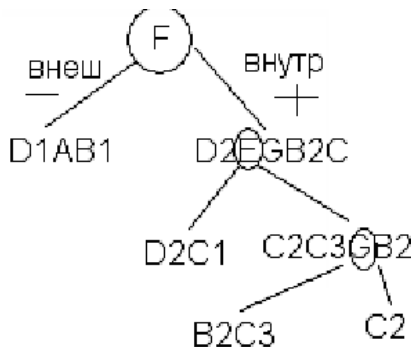
Вопрос № 3

Показать пример построения BSP-дерева в алгоритмах удаления невидимых граней.

Построим BSP-дерево для данного помещения



Возьмем в качестве корневого элемента сторону F



Для показа BSP-дерева можно воспользоваться алгоритмом:

Например, мы находимся в точке X. Обращаемся в корневой элемент BSP-дерева. Если мы находимся внутри по отношению к F, то мы идем по дереву во внешнюю ветвь. Выписываем элементы ветви в отдельный список D1AB1. Далее поднимаемся в F и выписываем значение F. Идем в правую ветвь в узел E. Если мы находимся внутри по отношению к E, то идем во внешнюю ветвь. Выписываем D2C1EB2C3GC2.

Определение понятия внешней стороны:

Пусть требуется определить расположение полигона 2 и 3 по отношению к полигону 1.

1. Для этого используют тестовую функцию:

$$T = Ax + By + Cz + D$$

Если в T полигона 1 подставить узловые точки полигона 2 и все значения будут положительны, то полигон 2 относится к правой ветви BSP-дерева, если отрицательны – то к левой. Если знаки меняются, то полигоны пересекаются.

Алгоритм визуализации BSP-дерева

Если камера находится в положительном подпространстве сцены по отношению к корневому элементу, то выводим отрицательную ветвь, далее корневой элемент и в конце выводим правую ветвь и это делается для каждого узла дерева рекурсивно.