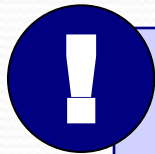


Структурное программирование на языке Си

Этапы разработки программ

1. Постановка задачи

- определить **цель** и **категорию** программы (системная, прикладная)
- определить **исходные данные** и требуемый **результат**
- проверить, является ли задача **хорошо поставленной** (должны быть определены все связи между исходными данными и результатом)



Плохо поставленные задачи:

- не хватает исходных данных
 - заданы не все связи между исходными данными и результатом
 - задача не имеет решения
 - задача имеет множество решений
- зафиксировать требования к программе **в письменной форме**

Этапы разработки программ

2. Разработка модели данных

- формальная модель
- типы данных (массивы, структуры, ...)
- взаимосвязь между данными

3. Разработка алгоритма

- выбор существующего или разработка нового
- возможен возврат к шагу 2

4. Разработка программы

Языки: C, C++, Visual Basic, Delphi (Паскаль), ...

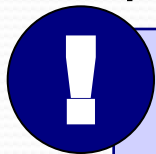
5. Отладка программы (поиск и исправление ошибок)

debug – извлечение жучков (*bug*), 1945, MARK-I

- **отладчик** (точки останова, пошаговый режим, просмотр переменных)
- **профайлер** (сколько выполняется каждая из процедур)

Этапы разработки программ

- 6. Тестирование программы** (проверка на исходных данных, для которых известен результат)
- **альфа-тестирование:** внутри фирмы (тестеры)
 - **бета-тестирование:** в других организациях, распространение через Интернет



Тестирование может показать наличие ошибок, но не их отсутствие.

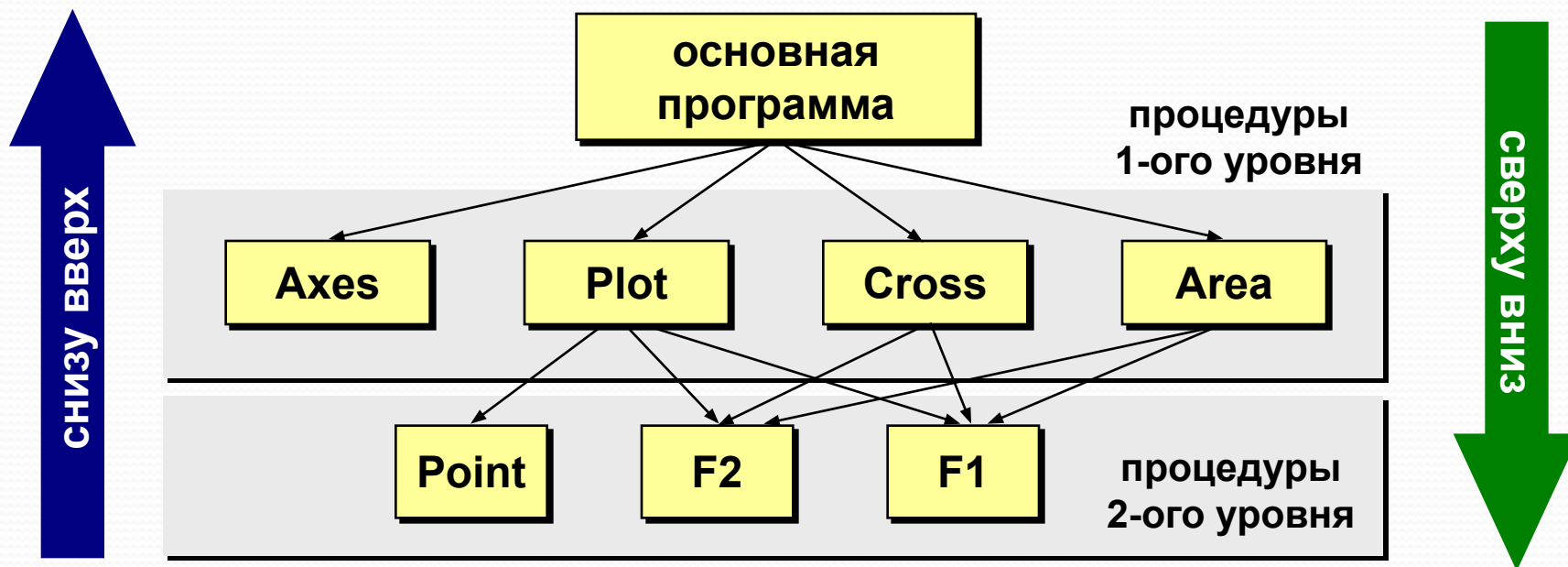
7. Разработка документации

- справочная система
- руководство пользователя (*User Manual*)
- руководство разработчика

8. Сопровождение (техническая поддержка)

- исправление ошибок, найденных заказчиком
- обучение и консультирование заказчика
- новые версии по льготной цене

Методы проектирования программ



Проектирование «снизу вверх»

сначала составляются процедуры нижнего уровня, из которых затем «собираются» процедуры более высокого уровня.



- **легче начать** программировать
- более **эффективные** процедуры



- процедуры необходимо связывать с основной задачей («**держат в голове**»)
- при окончательной сборке может **не хватить** «кубиков»
- часто программа получается **запутанной**
- сложно распределить работу **в команде**

Проектирование «сверху вниз»

метод последовательного уточнения:

- 1) начинаем с **основной программы**;
- 2) она разбивается на подзадачи, для каждой из которых пишется процедура-«**заглушка**»;
- 3) реализуем каждую из процедур тем же способом.



- меньше вероятность **принципиальной ошибки** (начали с главного)
- проще **структура** программы
- удобно **распределять** работу в команде



- в разных блоках могут быть реализованы похожие операции (можно было решить одной **общей процедурой**), особенно в команде

Структурное программирование

Существовавшие проблемы:

- увеличилась **сложность** программ
- сократилось **время** на разработку

Цели:

- **повысить надежность**
- **уменьшить время и стоимость** разработки
- **облегчить тестирование и отладку**
- **возможность переделки** одного модуля
- **улучшить читабельность**
 - без переходов на другую страницу
 - избегать трюков и запутанных приемов

Структурное программирование

Принципы:

- **абстракции:** программу можно рассматривать на любом уровне без лишних подробностей
- **модульности:** программа разбивается на отдельные модули, которые могут отлаживаться независимо друг от друга
- **подчиненности:** связь между модулями «сверху вниз»
- **локальности:** каждый модуль использует только свои локальные переменные, глобальные переменные только в крайних случаях

Модуль

Модуль – это программный блок (процедура или функция), отделенный от кода других модулей, который полностью решает самостоятельную задачу своего уровня.

- работа модуля не зависит от того, **откуда** он вызывается, и от того, сколько раз он вызывался **до этого**
- размер модуля не более **50-60 строк** (1 страница)
- модуль имеет **один вход** и **один выход**
- модуль начинается с «шапки»-комментария (входные данные, результаты, какие модули использует)
- имена переменных – **смысловые**
- в одной строке – один оператор
- «трюки» – долой

Оформление текста программы

Шапка – комментарий в начале процедур и функций.

```
//-----  
// Sum сумма элементов массива  
// Вход: A[] - массив целых чисел  
//        n - размер массива  
// Выход: S = A[0]+A[1]+...+A[n-1]  
// Вызывает: -  
//-----  
int Sum ( int A[], float n )  
{  
...  
}
```