

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

Отладка и тестирование программы

1. Отладка программы
2. Тестирование программы

Отладка программы

Отладка программы

Ошибки анализа

Неполный учет возможных ситуаций

Появление отрицательных значений переменных, малых и больших величин.

Неверное решение задачи (логические ошибки)

Отсутствие начальных значений переменных.

Неверное условие окончания цикла.

Неверная индексация цикла.

Отсутствие условий инициирования цикла.

Неправильный выбор ветви алгоритма для продолжения процесса решения задачи.

Отладка программы

Ошибки общего характера

Этап программирования

Недостаточное знание или понимание программистом языка программирования.

Использование в программе команд, не обеспечивающих последовательности событий, установленной алгоритмом.

Пропуск операторов.

Отсутствие необходимых данных.

Непредусмотренные данные.

Неверный формат данных.

Большое значение для успешной отладки программы имеют простота и рациональность ее кодирования.

Отладка программы

Синтаксические ошибки

Нарушение правильности программы связано с неверной синтаксической конструкцией программы и/или выдачей неверных результатов.

Пропуск необходимого знака пунктуации.

Несогласованность скобок.

Пропуск нужных скобок.

Неправильная структура оператора.

Неправильное образование имени переменной.

Неправильное использование арифметических операторов.

Неверное написание зарезервированных слов.

Отладка программы

Процесс обнаружения ошибок

Точка обнаружения — место в программе, где ошибка себя проявляет или становится очевидной. Выявляется первой и служит отправным пунктом для поиска точки происхождения.

Точка происхождения — место в программе, где возникают условия для появления ошибки.

**Действительная ошибка исходит не из точки обнаружения,
а из точки происхождения.**

Отладка программы

Ситуации и их причины

Синтаксических ошибок нет, но программа не скомпилирована

Причина: принципиальная ошибка в программе.

Программа скомпилирована, работает, но не выдает результатов

Причина: логические или синтаксические ошибки.

Программа скомпилирована, работает, но происходит преждевременный останов

Причина: ошибки, приводящие к преждевременному прекращению работы программы («взрыв», «воронка»).

Программа скомпилирована, работает, но выдает неправильные результаты

Причина: логические или синтаксические ошибки.

Программа зациклилась

Отладка программы

Технология

Нельзя отлаживать все сразу

На каждом этапе проверяется отдельный фрагмент. Программа должна проходить по уже протестированным частям.

Отладку программы нужно начинать на простых тестовых данных

Входные данные для отладки лучше не вводить, а задавать в виде последовательностей в массивах или в файлах.

Набирать статистику для анализа

Проследить выполнение программы на различных комбинациях входных данных.

Отладка программы

Технология

Модульному программированию соответствует модульное тестирование

Отдельные модули следует вызывать из головной программы и отлаживать на тестовых данных.

Вместо ненаписанных модулей можно использовать «заглушки», дающие фиксированный результат

Нисходящему программированию соответствует нисходящее тестирование

Внутренние части программы могут быть заменены «заглушками», позволяющими отладить внешние части программы.

Методы отладки

Отладка с помощью операторов печати

Работа методом проб и ошибок.

Тестирование большого числа данных.

Изменения в программе, которые могут скрыть ошибку или внести новую.

Высокая трудоемкость для больших программ или систем.

Отладка с использованием автоматических средств

Метод индукции

Метод дедукции

Прослеживание логики в обратном порядке

Метод локализации для небольших ошибок. Отладка начинается в точке программы, где был обнаружен некоторый результат. Для этой точки следует установить, какими должны быть значения переменных.

Автоматизированные средства отладки программ

Средства отладчика

Трассировка программы.

Просмотр изменения значений контролируемых переменных и выражений в процессе трассировки.

Модификация значений переменных для тестирования программы.

Просмотр значений фактических параметров, вызванных подпрограмм.

Работа с точками прерывания.

Просмотр выходной информации.

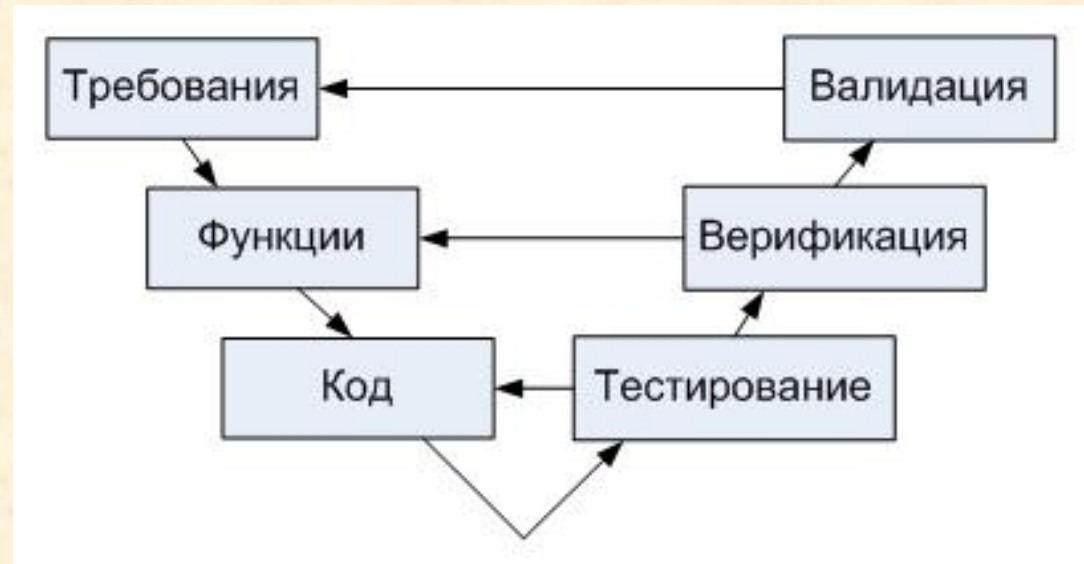
Тестирование программы

Тестирование, верификация, валидация

Тестирование — вид деятельности в процессе разработки, который связан с выполнением процедур, направленных на обнаружение (доказательство наличия) ошибок (несоответствий, неполноты, двусмысленностей и т.д.)

Целью **верификации** является достижение гарантии того, что объект (требования или программный код) соответствует требованиям, реализован без непредусмотренных функций и удовлетворяет проектным спецификациям и стандартам.

Валидация — это проверка соответствия системы ожиданиям заказчика.



Тестирование программы

Причины необходимости тестирования

Сложность программы.

Возможность отклонения от правильного понимания спецификации программы.

Внесение изменений в постановку задачи.

Данные для тестирования

Экстремальные значения. Например, если целая величина должна находиться в диапазоне от a до b , то следует проверить граничные значения, некоторые внутренние значения, значения за пределами диапазона.

Специальные значения. Константы, 0, 1, пустая строка, пустой файл, строка из одного символа и т.д.

Для циклов проверить выполнение 0, 1 и максимальное число раз.

Проверка устойчивости системы

Проверка на значениях, близких к предельным

Значение внутри диапазона

Минимальное значение

Максимальное значение

Проверка на граничных значениях

Значение внутри диапазона

Минимальное значение

Минимальное значение + 1

Максимальное значение

Максимальное значение - 1

Пример

Если в функцию

```
char sum(char a, char b){ return a + b;}
```

вычисляющую сумму чисел a и b, будут переданы значения 255 и 255, то в случае отсутствия специальной обработки ситуации переполнения сумма будет вычислена неверно.

Проверка робастности

Выход за границы диапазона

Робастность системы — это степень ее чувствительности к факторам, не учтенным на этапах проектирования. Например, неточность основного алгоритма, приводящего к ошибкам округления при вычислениях, сбои во внешней среде, данные, значения которых находятся вне допустимого диапазона.

Проверка выхода за границы диапазона

Минимальное значение - 1

Максимальное значение + 1,

Классы эквивалентности

Разбиение на классы эквивалентности особенно полезно, когда на вход системы может быть подано большое количество различных значений. Тестирование каждого возможного значения увеличит объем тестирования.

Пример классов эквивалентности

Значение из середины интервала.

Граничные значения.

Недопустимые значения за границами интервала.

Правила определения классов эквивалентности

Всегда будет, по меньшей мере, два класса: корректный и некорректный.

Если входное условие определяет диапазон значений, то будет три класса: меньше, внутри и больше диапазона.

Если элементы диапазона обрабатываются по-разному, то каждому варианту обработки будут соответствовать разные требования.

Уровни тестирования

Модульное тестирование. Тестируется минимально возможный для тестирования компонент, например, отдельный класс или функция.

Интеграционное тестирование. Тестируются интерфейсы между компонентами, подсистемами или системами.

Системное тестирование. Тестируется интегрированная система на ее соответствие требованиям.

Альфа-тестирование. Имитируется реальная работа с системой штатными разработчиками, либо реальная работа с системой потенциальными пользователями/заказчиком. Цель: функциональное наполнение кода.

Бета-тестирование. Выполняется распространение предварительной версии (иногда с ограничениями по функциональности или времени работы) для некоторой большой группы лиц. Цель: исправление ошибок.

Стратегии упорядочения маршрутов для тестирования ПО

Стратегия 1. Учет числа строк текста программы в выделенных маршрутах или расчетной длительности их исполнения при функционировании программы.

Стратегия 2. Анализ числа альтернатив (условных переходов), определяющих образование каждого маршрута.

Стратегия 3. Использование вероятности исполнения маршрутов при реальном функционировании программы.