

Вопрос 5. Проверка и отладка программного кода.

1. Ошибки программирования

Отладка программы — это процесс проверки работоспособности программы и исправления обнаруженных при этом ошибок. Ошибки могут быть связаны с нарушением правил записи программы на конкретном языке программирования. Их программисту помогает найти используемая система программирования; она выдаёт на экран сообщения о выявленных ошибках.

Ошибки возникают в программах, когда строка кода или инструкция конфликтует с другими элементами кода. Такие ошибки или дефекты еще называют **аппаратными ошибками**. Суть в том, что программа не работает должным образом из-за ошибки.

Программирование - сложный процесс. Поскольку это работа человека, часто могут возникать ошибки. Причиной ошибки также может являться конфликт в программном обеспечении при попытке заставить разные приложения работать вместе.

Виды ошибок в программном обеспечении:

- Часть ошибок формального характера, связанных с нарушением правил записи конструкций языка или отсутствием необходимых описаний, обнаруживает транслятор, производя синтаксический анализ текста программы. Транслятор выявляет ошибки и сообщает о них, указывая их тип и место в программе. Такие ошибки называются **синтаксическими**.
- Ошибочные ситуации могут возникнуть и при выполнении программы, например деление на ноль, извлечение корня квадратного из отрицательного числа, попытка открыть несуществующий файл. Такие ошибки называются **семантическими**, они связаны с неправильным содержанием действий или использованием недопустимых значений величин.
- Программа, не имеющая синтаксических и семантических ошибок, может не дать верных результатов из-за **логических** ошибок в алгоритме. Ошибки подобного рода могут возникнуть на этапе постановки задачи, разработки математической модели, разработки алгоритма.

Легенда гласит, что первая компьютерная ошибка была настоящей ошибкой. В 1945 году в Гарварде моль между двумя электрическими реле релейного калькулятора Mark II Aiken отключила всю машину.

Рассмотрим программу вычисления значения функции $f(x)$ при целом x .

$$f(x) = \begin{cases} 3|x|, & x < -2 \\ 9/x, & -2 \leq x \leq 2, x \neq 0 \\ 12, & x = 0 \\ \sin x, & x > 2 \end{cases}$$

```

var x: integer; f: real;
begin
  writeln ('Введите целое x');
  readln (x);
  if x < -2 then f := 3 * abs(x) else
    if (x >= -2) and (x <= 2) then f := 9 / x else
      f := sin(x);
  writeln ('f(' , x, ') = ' , f : 6:2);
end.
        
```

Синтаксическая ошибка

→ writeln ('Введите целое x')

Семантическая ошибка

→ деление на 0 при x=0

Логическая ошибка

→ if (x >= -2) and (x <= 2) then f := 9 / x else

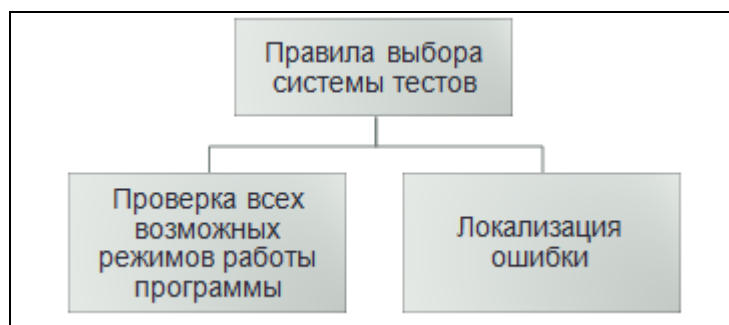
2. Отладка и тестирование

Проверка правильности разработанной программы осуществляется с помощью тестов. **Тест** — это конкретный вариант значений исходных данных, для которого известен ожидаемый результат.

О правильности разработанной программы свидетельствует также соответствие полученных данных экспериментальным фактам, теоретическим положениям и т. д. При этом может возникнуть необходимость уточнить разработанную математическую модель, полнее учесть особенности изучаемого объекта или процесса. По уточнённой математической модели снова составляется программа, анализируются результаты её выполнения. Так продолжается до тех пор, пока полученные результаты не будут достаточно точно соответствовать изучаемому объекту.

- Только после того как появится уверенность, что программа обеспечивает получение правильных результатов, можно приступить непосредственно к ее эксплуатации (расчётам).
- После завершения расчётов полученные результаты используются в практической деятельности.

Тестирование программы — это выполнение программы на наборах исходных данных (тестах), для которых известны результаты, полученные другим методом.



При тестировании программы простой и действенный метод дополнительного контроля над ходом её выполнения — **получение контрольных точек**, т.е. контрольный вывод промежуточных результатов.

Эффективная отладка очень важна для улучшения качества системы. Отладка не гарантирует, что приложение полностью соответствует его требованиям. Вот почему необходимо всегда тестировать программное обеспечение после исправления ошибки.

Для программиста отладка более трудна для сложных систем, особенно если разные подсистемы тесно связаны, потому что изменения в одном приложении или интерфейсе в другой системе могут вызвать ошибки, которые трудно отследить. Иногда для отладки программы требуется больше времени, чем для написания кода. Отладка и тестирование принадлежат друг другу.

Программные модули проходят множество строгих тестов во время разработки программного обеспечения. Различные типы тестов должны определять проблемы и ошибки до того, как программное обеспечение будет запущено в производство. **Есть модульные тесты, системные тесты, интеграционные тесты, тестирование производительности, приемочные испытания и так далее.** Во время этих тестов всегда находятся ошибки и недочеты. Программный код может иметь потенциальные ошибки, которые могут привести к

неожиданному поведению приложения или даже к сбою. Эти ошибки возвращаются программисту, который затем обычно удаляет их с помощью отладки.

В процессе отладки полные программы регулярно кодируются и выполняются специально для диагностики проблем. Большие приложения, содержащие миллионы строк программного кода, делятся на небольшие части. Эффективно сначала тестировать и отлаживать каждую часть по отдельности, а затем приложение в целом.

2. Процесс отладки

Основные методы отладки:

- 1) отладка вручную;
- 2) использование программы-отладчика;
- 3) включенный в программу режим отладки.

При **отладке программы вручную**, программист должен начать с обнаруженной ошибки. Затем он должен изолировать исходный код с ошибкой и исправить ошибку. Программист должен знать, как исправить ошибку. Это требует знания анализа и устранения проблем. Когда ошибка устранена, программное обеспечение готово к использованию.

Чтобы отладить программу вручную:

1. Воспроизведите ситуацию ошибки.
2. Опишите ошибку. Постарайтесь получить как можно больше информации от пользователя, чтобы определить точную причину.
3. Запишите состояние программы при появлении ошибки. Постарайтесь получить все значения переменных и состояния программы в это время.
4. Проанализируйте статус на основе переменных и действий в данный момент. Исходя из этого, ищите причину ошибки.
5. Исправьте сообщенную ошибку, но также проверьте, не была ли введена новая ошибка.

При отладке вручную, программист должен настроить программу, чтобы останавливать программу в определенных точках и читать переменные. С большим приложением это невозможно, в этих случаях применяется программа-отладчик.

Отладчик — это программный инструмент для программистов, используемый для тестирования и обнаружения ошибок (багов) в других программах.

Большинство отладчиков используют имитаторы набора команд. В отличие от запуска программы непосредственно на процессоре, симулятор обеспечивает более высокий уровень контроля над выполнением. Программист может использовать симулятор, чтобы остановить программу отладки или позволить ей работать в зависимости от конкретных обстоятельств. Программист может сам предопределять и устанавливать эти настройки.

Программист может отслеживать выполнение программы шаг за шагом, оценивая значение переменных. Выполнение может быть остановлено в любом месте, чтобы получить значение переменных.

Если программа выйдет из строя, отладчик укажет на местоположение ошибки в программе. Помимо остановки в определенных точках, большинство отладчиков также могут выполнять программы шаг за шагом. Часто программист может одновременно изменять переменные и статусы программ, чтобы программа пошла по другому пути. Тогда можно будет проверить, что произошло.

Поскольку это промежуточный уровень, использование тренажеров замедляет скорость работы программы.

Мощность отладчика может сильно различаться от языка к языку. Как правило, языки программирования на основе компилятора имеют лучший отладчик, чем языки, основанные на интерпретаторах. Известные языки с компилятором: C, Java и Cobol. Основанные на интерпретаторе языки: HTML, PHP и Javascript.

Некоторые языки программирования предлагают отладчик для проверки кода на наличие ошибок, возникающих во время выполнения.

Разработчик часто создает дополнительные возможности отладки в своих приложениях. Это часто происходит с SaaS и веб-приложениями. Затем приложению предоставляется дополнительная настройка, так называемый **«режим отладки»**. Это может быть включено и выключено при необходимости.

Если режим отладки включен, программа показывает:

- Переменные и их содержание.
- Статусы, которые применяются в то время.
- Обширные сообщения об ошибках.
- Иногда пользователь имеет больше прав для запроса или изменения данных.
- То, что программист также считает важным показать.

С одной стороны, преимуществом режима отладки является то, что для отладки не требуется дополнительная программа. С другой стороны, его применение может быть опасным. Содержание отладочной информации очень интересно для хакеров. Показанные данные, такие как имена переменных и их значения, многое говорят о том, как работает приложение. Это облегчает обнаружение слабых мест, и порой этого достаточно, чтобы войти в систему незамеченным.

Поэтому программисты не должны забывать снова выключать режим отладки после своей работы. Это может предотвратить многие проблемы.

Для отладки программ обычно применяют три способа:

1. Пошаговая отладка программ с заходом в подпрограммы;
2. Пошаговая отладка программ с выполнением подпрограммы как одной операции;
3. Выполнение программы до точки остановки.

Пошаговая отладка программ заключается в том, что выполняется одна операция программы и, затем контролируются те переменные, на которые она должна была воздействовать.

Если в программе имеются уже отлаженные подпрограммы, то подпрограмму можно рассматривать, как одну операцию программы и воспользоваться вторым способом отладки программ.

Если в программе существует достаточно большой участок программы, уже отлаженный ранее, то его можно выполнить, не контролируя переменные, на которые он воздействует. Использование точек остановки позволяет пропускать уже отлаженную часть программы. Точка остановки устанавливается в местах, где необходимо проверить содержимое переменных или просто проконтролировать, передаётся ли управление данной операции.

Практически во всех отладчиках поддерживается это свойство (а также выполнение программы до курсора и выход из подпрограммы). Затем отладка программы продолжается в

пошаговом режиме с контролем локальных и глобальных переменных, а также внутренних регистров микроконтроллера и напряжений на выводах этой микросхемы.

Контроль данных при пошаговой отладке. При использовании встроенного отладчика программ для контроля переменных можно воспользоваться окном Watch. В большинстве случаев это намного выгоднее, чем использовать просмотр памяти данных. Переменные в этом окне отображаются в том формате, в котором они были объявлены в программе. Преобразование данных из одной формы представления в другую осуществляется встроенными средствами отладчика. Если же отладочная информация по какой либо причине была потеряна или программирование ведется на языке ассемблер, то состояние переменных можно проконтролировать просмотром содержимого внутренней и внешней памяти микроконтроллера. Для определения конкретной ячейки памяти микроконтроллера, занятой под переменную можно воспользоваться листингом, создаваемым редактором связей компилятора или самим компилятором, если не используется многомодульное программирование.

Литература:

1. Лягинова О.Ю., Можеева М.Г. Этапы решения задач на ЭВМ: онлайн-презентация / Кафедра математики и информатики ЧГУ / <https://en.ppt-online.org/593552> (стр. 24-31)
2. Босова Л.Л., Босова А.Ю. Информатика. 9 класс: учебник / Л. Л. Босова, А. Ю. Босова. — 6-е изд. — М. : БИНОМ. Лаборатория знаний, 2016. — 208 с.
3. Отладка и отладка программного обеспечения / <https://ru.itpedia.nl/2019/01/30/debuggen-en-foutopsporing-van-software/>
4. <https://www.computer-museum.ru/technlgy/proclact/progr/otladka.htm>