Отладка программ

- Отладка
  - Виды ошибок
  - Тестирование
  - Трассировка
  - Отладочная печать

Отладка

1. Каждая последняя обнаруженная ошибка является предпоследней.

2. Когда программа полностью отлажена она становится ненужной.

Аксиомы отладки.

Виды ошибок

- Синтаксические — нарушен синтаксис языка
- Семантические (логические) — программа выполняет то, что вы написали, а не то, что вы хотели...

- О синтаксических ошибках сообщает компилятор, нередко с указанием строчки. Их сравнительно легко найти и исправить.

Логические ошибки

- Логические ошибки проявляются в виде:
  - зависания (зацикливания) программы;
  - аварийного завершения работы (вылета) программы;
  - неверных выходных данных.
- Зависание программы обычно говорит о наличии в ней цикла, который не заканчивается. Аварийное завершение происходит, например, при обращении к невыделенной вам памяти.
- Логические ошибки значительно труднее в обнаружении и отладке.
- Отладка занимает 60-75% времени на разработку программы.

Тестирование

Самая страшной ошибкой является та, которую обнаружили не вы, а начальник.
Аксиомы отладки.

- Разработанная программа должна тщательно тестироваться.
- При тестировании необходимо проверить все возможные ситуации, возникающие при выполнении программы:
  - в арифметических выражениях учесть особые случаи: деление на ноль, квадратный корень из отрицательного значения и т.д;

Тестирование

- при наличии алгоритмических структур «**развилка**» или «**выбор**» необходимо проверить все пути, задаваемые этой структурой (даже если они ничего не делают);
- При тестировании **развилок** и **выборов**, находящихся внутри **цикла**, необходимо проверить каждый путь развилки в трех ситуациях:
  - на первом шаге цикла;
  - на последнем шаге цикла;
  - на любом промежуточном шаге цикла.

Методы отладки

Пошаговое выполнение (трассировка)

Отладочная печать

Пошаговое выполнение

- Цель трассировки программы — получить представление о том, каким образом изменяются переменные на каждом шаге ее выполнения.
- Единицей (шагом) выполнения программы при трассировке является строка, а не оператор.
- Место начала трассировки задается с помощью точки останова.

Точки останова

- Точка останова — это место в программе, при выполнении которого она приостанавливает свою работу так, чтобы можно было посмотреть значения переменных в этот момент.
- Устанавливаются нажатием кнопки мыши на серое поле слева от текста программы, отображаясь там красными точками ().
- Точка должна ставится на оператор, выполняющий какое-либо действие. Не следует ставить ее на строку комментария, заголовок функции и т.д.

Точки останова

- При изменении программы следите за тем, чтобы она компилировалась без синтаксических ошибок. Иначе может быть запущена старая версия, и точки останова работать не будут.
- Работают только при запуске программы в отладочном режиме (F5, )
- Нажатие F5 после останова продолжает работу программы (до следующей точки останова, если она есть, либо до конца).

Во время останова

- Просмотреть значение переменных и выражений при останове можно, выделяя их в тексте программы, а также с помощью окон отладки (напр. окно *Autos* показывает текущие автоматические переменные, *Watch* — позволяет вводить интересующие вас выражения).
- Команда меню *Stop Debugging* () позволяет немедленно прервать отладку и перейти к изменению текста программы.
- Желтая стрелка показывает *следующую* строчку для выполнения.

Пошаговое выполнение

- Используются следующие команды:
  - *Step Into* (F11, ) - выполняет строку, заходит внутрь функции, если она в строке есть;
  - *Step Over* (F10, ) - выполняет строку за один шаг;
  - *Step Out* (Shift+F11, ) - выполняет текущую функцию до конца и останавливается после возврата.
- Используйте *Step Into*, чтобы пошагово проверить работу своей функции; *Step Over* — с библиотечными функциями или чтобы быстро проверить, верно ли работает ваша функция.
- Используйте *Step Out*, если вы нечаянно вошли в функцию, в которую не хотели входить.

Отладочная печать

- Состоит в выводе на экран значений интересующих вас переменных.
- Применяется, если:
  - значение переменных сложных структурированных типов сложно посмотреть во время останова программы;
  - вы хотите получить быстрый «снимок» состояний переменных в нескольких контрольных точках, чтобы определить, в какой части программы ошибка;
  - программа не может быть остановлена, т.к. она взаимодействует в реальном времени с другой программой или физическим устройством.