

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 08.02.2021 16:51:23
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

МИНОБРАЗОВАНИЯ РОССИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования

«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра информационной безопасности

УТВЕРЖДАЮ
Проректор по учебной работе
О.Г. Локтионова
«А» 08.02.2021
(ЮЗГУ) 2017г.



ОБНАРУЖЕНИЕ ОШИБОК И ОТЛАДКА ПРОГРАММЫ

Методические указания по выполнению лабораторной работы по
дисциплинам «Основы реверсинжиниринга программных средств»,
«Методы защиты программного обеспечения» для студентов
специальности 10.03.01

Курск 2017

УДК 004.056.55

Составитель А.Л. Марухленко

Рецензент

Кандидат технических наук, доцент *И.В. Калуцкий*

Обнаружение ошибок и отладка программы: методические указания по выполнению лабораторных работ по дисциплинам «Основы реверсинжиниринга программных средств», «Методы защиты программного обеспечения» / Юго-Зап. гос. ун-т; сост. А.Л. Марухленко. Курск, 2017. - 9с.

Рассматриваются методы обнаружения ошибок и отладки программы. Указывается порядок выполнения лабораторной работы и содержание отчета.

Методические указания соответствуют требованиям программы, утвержденной учебно-методическим объединением по образованию в области информационной безопасности (УМО ИБ).

Предназначены для студентов специальности 10.03.01.

Текст печатается в авторской редакции

Подписано в печать 01.11.2017. Формат 60x84 1/16.

Усл.печ. л. 0,5. Уч.-изд.л. 0,5. Тираж 30 экз. Заказ _____. Бесплатно.

Юго-Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.

ОГЛАВЛЕНИЕ

1. Цель работы:	3
2. Методические указания.....	3
3. Порядок выполнения работы.....	7
4. Порядок оформления отчета.....	8
5. Контрольные вопросы.	8
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	9

.

1. ЦЕЛЬ РАБОТЫ

Получить начальное представление о возможностях отладчика VBA.

2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Одной из важнейших проблем, связанных с программированием, является проблема своевременного обнаружения и устранения ошибок, возникающих при создании программ. Человечество подходит к разрешению этой проблемы с разных сторон. Во-первых, создаются методы и средства автоматизации программирования, позволяющие уменьшить вероятность возникновения ошибки как таковой. К их числу относятся технологии структурного программирования и само объектно-ориентированное программирование. Во-вторых, совершенствуются сами программные средства и языки программирования, создаются специализированные программы, которые позволяют относительно легко обнаруживать ошибки. Наконец, в-третьих, ведутся статистические исследования и выявляются типовые ошибки, которые делают программисты. На основании полученной статистики выдаются рекомендации разработчикам нового программного обеспечения. Настоящая лабораторная работа посвящена изучению методов обнаружения ошибок, реализованному в VBA.

Обычно выделяют три основных вида ошибок, которые приводят к неправильному выполнению программы или делают ее выполнение просто невозможным. Первый вид ошибок – это ошибки, возникающие на этапе компиляции. Основным смыслом определяемых на этапе компиляции ошибок – это некорректная запись операторов программы с точки зрения правил языка программирования. Как следствие, компилятор не может создать код и требует внести изменения в программу. Компилятор VBA высвечивает строку программы, которая содержит ошибку, красным цветом и выдает дополнительное диагностическое сообщение.

Ошибки этапа компиляции устраняются программистом с относительно небольшими затратами труда, поскольку их поиск автоматизирован, а для уточнения правил языка программист может легко воспользоваться справочной литературой или встроенной в компилятор VBA системой помощи. Ее вызов осуществляется при нажатии клавиши F1. С целью минимизации вероятности возникновения орфографической ошибки при записи класса или метода объекта может быть вызвано специальное контекстное меню вводом команды *Редактирование/Список свойств и методов*. Аналогичное меню может быть вызвано и для списка констант.

Быстрый вызов меню можно осуществить и правой клавишей мыши при наборе текста. Наконец, распознанные операторы языка выделяются цветом, что позволяет уменьшить вероятность ошибки, связанной с неправильным именем переменной. В любом случае ошибки компиляции сопровождаются диагностическим сообщением, из которого, воспользовавшись при необходимости системой помощи, можно установить их причину.

Более сложный класс ошибок – это ошибки, возникающие на этапе выполнения программы. Эти ошибки в том или ином виде связаны с обрабатываемыми данными и, как следствие, не могут быть определены на этапе компиляции, поскольку конкретные значения данных в этот момент неизвестны. При возникновении подобных ошибок на экран выдается диагностическое сообщение с указанием кода ошибки и его кратким описанием. Составляя алгоритм, программист обязан предусмотреть возможность их появления и принять дополнительные меры по их локализации и, если это требуется, перехвату. Список некоторых ошибок этапа выполнения приведен в таблице 1.

Наиболее сложным видом ошибок при программировании являются алгоритмические ошибки. Причина таких ошибок двояка – с одной стороны они возникают из-за неправильного составления алгоритма, с другой из-за неправильного кодирования (записи операторов программы не в соответствии с составленным алгоритмом). К сожалению, единственным способом обнаружения алгоритмических ошибок является тестирование. Под тестированием обычно понимают испытание программы при условии подачи на нее заведомо известных данных (теста) и проверки результатов ее работы (они должны быть определены совместно с подготовкой теста). Особенностью тестирования является то обстоятельство, что если тест обнаруживает факт существования алгоритмической ошибки (программа выполняется неверно), то ошибка существует и должна быть устранена. В то же время, если тест не находит ошибки, то это обстоятельство не является доказательством того, что ошибка отсутствует. Как следствие, созданная программа должна быть подвергнута максимально возможному тестированию. Однако исчерпывающее тестирование программы, как правило, является невозможным из-за чрезвычайно большого числа возможных вариантов данных, в связи с чем приходится использовать методы программирования, уменьшающие вероятность возникновения ошибки, и рассчитывать на искусство программиста. Если в процессе тестирования была обнаружена ошибка, программист должен начать процесс определения конкретных операторов программы, вызвавших появление ошибки, обычно называемый отладкой. Для автоматизации процесса тестирования и отладки созданы специальные программы, которые получили название программ-отладчиков. Подобная программа есть и в составе редактора VBA.

Таблица 1 – Ошибки этапа выполнения

Код ошибки	Диагностическое сообщение
5	Приложение не запущено
6	Переполнение
7	Не хватает памяти
9	Выход индекса за границы диапазона
11	Деление на нуль
13	Несоответствие типов данных
18	Произошло прерывание, вызванное пользователем
52	Неправильное имя файла или идентификатора
53	Файл не найден
54	Неправильный режим работы с файлом
55	Файл уже открыт
56	Ошибка ввода-вывода
61	Переполнение диска
68	Устройство не доступно
71	Диск не готов
72	Повреждена поверхность диска
335	Невозможен доступ к системным ресурсам
368	Истек срок действия данного файла
482	Ошибка принтера

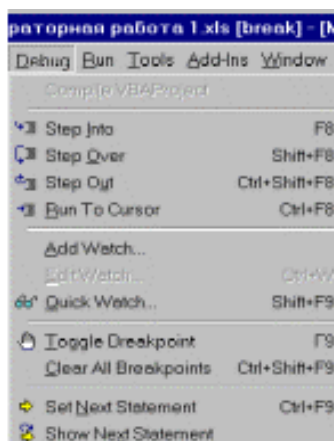


Рисунок 1 – Меню отладчика

Режим отладки включается из главного меню при активном окне редактора при выполнении команды *Отладка/Шаг с заходом*. Выключение режима обеспечивает выполнение команды *Выполнить/Сброс*. Внешний вид меню показан на рисунке 1. Обратите внимание на то, что основные команды отладчика вызываются комбинациями клавиш F8 и F9. Команда *Шаг с заходом* позволяет оператор за оператором выполнить тестируемую программу, включая вызываемые программой функции и процедуры. Команда *Шаг с обходом* исключает пошаговое выполнение вызываемых модулей. Команда *Шаг с выходом* завершает пошаговое выполнение вызванного модуля. Команда *Выполнить до текущей позиции* выполняет программу до оператора, на котором установлен курсор. Команда *Точка останова* задает и снимает точку останова в тексте программы, причем конкретный оператор предварительно выбирается курсором. Команда *Снять все точки останова* удаляет все установленные в программе точки останова. Выполнение команд *Задать следующую инструкцию* и *Показать следующую инструкцию* позволяет найти следующий выполняемый оператор в окне редактирования.

Текущие значения переменных можно наблюдать задавая их имена в окне контрольных значений (*Отладка/Добавить контрольные значения*, *Отладка/Контрольные значения*, *Вид/Окно локальных переменных*). При выполнении команды *Вид/Проверка* можно задать дополнительный оператор VBA, или изменить значение любой переменной оператором присваивания.

3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Включите компьютер. Загрузите *Windows*. Загрузите *Excel* и выполните команду *Файл/Открыть*. При стандартной настройке на экране появится окно *Открытие документа*, а в окне *Папка* появится название основной рабочей папки *Мои документы*. Если по каким-то причинам установилась другая папка, нажмите кнопку меню *Папка* и выберете папку *Мои документы*. Ниже будет показано содержимое папки *Мои документы*. Найдите папку с номером вашей группы. Откройте эту папку и найдите файл, соответствующий вашей фамилии, установите на него курсор. Нажмите кнопку *Открыть* и убедитесь, что в верхней левой области экрана после текста *Microsoft Excel* появилось название рабочей папки с вашей фамилией.
2. Перейдите на следующий лист рабочего поля. Если свободные листы отсутствуют, выполните команду *Вставить/Лист*.
3. Последовательно командам *Сервис/Макрос/Редактор Visual Basic* запустите редактор VBA.
4. Вернитесь к макросу, записанному вами при выполнении предыдущей лабораторной работы.
5. Запустите отладчик и выполните программу по шагам.
6. Введите точки останова и выполните программу с точками останова. Удалите точки останова.
7. Включите окно контрольных значений, введите в него несколько переменных и проконтролируйте в нем изменения значений переменных при пошаговом выполнении программы.
8. Повторите пошаговое выполнение программы, контролируя значения в окне локальных переменных.
9. Воспользуйтесь окном проверка для оперативного изменения текста выполняемой программы, а также для изменения значений переменных.

4. ПОРЯДОК ОФОРМЛЕНИЯ ОТЧЕТА

Отчетом о лабораторной работе является файл с именем, совпадающим с фамилией студента с результатами работы в папке *Мои документы/номер группы*.

5. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Перечислите виды ошибок, возникающих в процессе создания и эксплуатации программного обеспечения.
2. Как диагностируются ошибки, выявляемые компилятором VBA? В чем причина этих ошибок?
3. Какие ошибки могут возникнуть на этапе выполнения программы? В чем причина возникновения этих ошибок?
4. Что такое тест и как выполняется тестирование?

5. Каковы виды пошаговых режимов работы отладчика?
6. Какие существуют способы контроля над значениями переменных?
7. Как можно изменить значения переменных в процессе отладки программы?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бычков М.И. Основы программирования на VBA для Microsoft Excel:[Текст] учебное пособие, 2010 - 99с.
2. Кукушкина Е.В. Начальные сведения о языке программирования Visual Basic for Application: [Текст] учебная литература для ВУЗов, 2014 – 111с.
3. Абрамян М. Э. Практикум по параллельному программированию с использованием электронного задачника Programming Taskbook for MPI: [Текст] учебное пособие, 2010 – 172с.
4. Паппас, К. Х. Отладка в С++ [Текст] : руководство для разработчиков / Пер. с англ.; Под ред. В. В. Тимофеева. - М. : БИНОМ, 2001. - 512 с.
5. Этапы создания программы на языке Ассемблер. Изучение интегрированной среды отладчика Turbo Debug [Электронный ресурс] : методические указания к лабораторной работе №1 / Юго-Зап. гос. ун-т ; сост.: Е. А. Титенко, Д. И. Родионов, Е. А. Петрик. - Курск : ЮЗГУ, 2010. - 14 с.