

УДК 681.3

Г.Ф. КРИВУЛЯ, О.С. КОРОБКО

*Харьковский национальный университет радиоэлектроники, Украина***ПРОЯВЛЕНИЕ ОШИБОК ВРЕМЕНИ ВЫПОЛНЕНИЯ ПРИ РАЗРАБОТКЕ ПРИЛОЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ .NET-ТЕХНОЛОГИЙ**

В работе проведен обзор ошибок, возникающих при разработке программного обеспечения. Поставлена задача о необходимости тестирования программного обеспечения для ошибок времени выполнения. Рассматриваются основные среды разработки программных продуктов. Приводится пример проявления ошибок времени выполнения в виде исключительных ситуаций, возникающих при функционировании .NET-приложения.

программное обеспечение, ошибка времени выполнения, объектно-ориентированное программирование, платформа .NET, тестирование приложения, среда разработки программного обеспечения

Введение

Одной из наиболее актуальных научно-практических задач программной инженерии является разработка качественного программного обеспечения (ПО), соответствующего не только исходной спецификации разработки, но отвечающего требованиям безопасности и эффективности.

Качество программного продукта характеризуется набором показателей, определяющих, насколько программный продукт соответствует требуемой функциональности, насколько оптимально и эффективно реализованы поставленные задачи, насколько дружелюбен и удобен конечному потребителю представленный пользовательский интерфейс. Немаловажным свойством ПО является его надежное функционирование при неблагоприятных внешних воздействиях, его безопасность и отсутствие «узких мест».

Анализ публикаций. В работе [1] обсуждены проблемы контроля качества разработки программного обеспечения с позиций тестирования. Приведены основные понятия тестирования: терминология тестирования, различия тестирования и отладки, фазы и технология тестирования, проблемы тестирования. Рассмотрены критерии выбора тестов: структурные, функциональные, стохастические,

мутационные, оценки покрытия проекта. Также рассмотрены различные виды тестирования и особенности процесса планирования тестов.

В работе [2] основное внимание уделено приемам и технологиям тестирования. Рассматривается процесс разработки тестов и системы отслеживания обнаруженных проблем. Определяются основные положения процесса анализ и документирования ошибок. Проводится анализ инструментальных средств тестировщика. Предоставляется ряд рекомендаций относительно планирования тестирования ПО.

Работа [3] посвящена обработке ошибок, возникающих при вызове функций Win32 API. Приведены примеры ситуаций, приводящих к возникновению ошибок времени выполнения. Рассматривается задача организации программного кода с целью устранения возможных ошибок времени выполнения.

Основные типы ошибок, возникающие при разработке программного обеспечения

При создании ПО разработчик сталкивается с тремя типами ошибок [1]: ошибками синтаксиса, ошибками времени выполнения и логическими ошибками.

Синтаксическіе ошибки появляются при написании кода. Они обнаруживаются компилятором и могут быть исправлены непосредственно в среде разработки ПО с минимальными затратами времени

Логическіе ошибки рассматриваются как некоторые элементы функциональности программного обеспечения, не соответствующие его спецификации. Они проявляются при использовании программных приложений и являются достаточно трудноустранимыми в связи со сложностью определения причин их возникновения. В качестве примера ошибок логики можно рассматривать выполнение одного оператора вместо другого, обращение не к требуемому ресурсу (например, файлу или блоку памяти), невыполнение перерисовки окна, когда это необходимо, отправку неверного запроса на сервер базы данных и т.п.

Как отдельный вид логических ошибок можно выделить недостатки графического интерфейса пользователя: например, отсутствие полосы прокрутки там, где это необходимо, или неверное позиционирование изображения в окне.

Во многих случаях логические ошибки в программе могут привести к возникновению ошибок времени выполнения.

Ошибки времени выполнения (run-time errors) порождены программным кодом, который может быть синтаксически корректным, но при некоторых условиях не может быть выполнен. В качестве примера можно привести наиболее типичную ошибку времени выполнения: например, программа содержит синтаксически правильный блок кода, открывающий некоторый файл на жестком диске для записи в него информации. В том случае, если файл с указанным именем действительно существует, он не поврежден, не имеет атрибута «только для чтения» - программа будет выполнена корректно, файл будет открыт и запись в него будет выполнена. Но если файл окажется поврежденным или не будет доступен для записи, при попытке выполнить дан-

ный фрагмент кода произойдет сбой в программе, и ее выполнение будет аварийно завершено (разумеется, это утверждение имеет место при отсутствии необходимых проверок условия или блоков обработки исключительных ситуаций).

Постановка задачи. Вопросам тестирования программного обеспечения на предмет его соответствия спецификации уделено достаточно много внимания в литературе [2]. Не менее актуальной является задача проверки программного кода с точки зрения его надежности и безопасности – т.е. устойчивости полученной в результате компиляции данного кода программы к различным воздействиям, которые могут привести к возникновению ошибок времени выполнения [3]. Иначе говоря, для получения качественного, работоспособного и надежного программного обеспечения необходимо не только проверять, делает ли созданная программа то, что должна, согласно ее спецификации, но и не делает ли она то, что не должна. Очевидно, что данную проверку необходимо выполнять не только в условиях нормального функционирования программы; требуется моделировать ситуации, в которых разработанное программное обеспечение может повести себя некорректно.

Обзор основных сред разработки ПО

На сегодняшний день существует большое многообразие языков программирования высокого уровня, связанных с использованием различных технологий и спецификой реализуемых задач.

Так как современные тенденции в сфере программной инженерии разработки программного обеспечения ведут к увеличению роли платформы .NET в области разработки программного обеспечения, актуальной является задача тестирования программных продуктов, созданных именно с использованием данной платформы и .NET-совместимых языков программирования.

Спеціально для платформи .NET Microsoft был разработан новый язык программирования: C# [4], сфера использования которого значительно расширилась в течение последних пяти лет.

Проявления ошибок времени выполнения

Рассмотрим на примере программное приложение, написанное на языке C# для демонстрации ошибок времени выполнения.

Пусть необходимо реализовать графическое приложение для просмотра графических изображений. Созданная программа должна позволять открывать файлы в формате .jpg, отображать их в главном окне программы и сохранять рисунки в формате bmp.

Данный программный продукт своей спецификации соответствует. Главное меню программы имеет пункты Open и Save для открытия и сохранения файлов соответственно. При выборе данных пунктов меню открывается диалоговое окно, позволяющее выбрать необходимый файл.

При выборе опции Save приложение сохранит рисунок в формате bmp в указанном месте на диске.

Однако, хотя данное приложение выполняет свои задачи, в его программном коде присутствует потенциальный источник ошибок времени выполнения.

Если пользователь выберет в главном меню пункт Save, когда ни один рисунок не открыт в главном окне программы, он получит следующее сообщение:

```
System.NullReferenceException: Object reference not set to an instance of an object.
```

Причиной появления такой ошибки является просчет программиста, не предусмотревшего возникновение подобной ситуации и некорректные действия пользователя, попытавшегося сохранить несуществующий объект.

Данная ошибка порождена следующим фрагментом программного кода:

```
SaveFileDialog saveDlg = new SaveFileDialog();
saveDlg.Filter = "bmp|*.bmp";
saveDlg.ShowDialog();

if (saveDlg.FileName != null && saveDlg.FileName != "")
{
    loadedImage.Save (saveDlg.FileName);
}
```

Возникновения приведенной ошибки можно избежать путем внесения следующих исправлений в коде:

```
SaveFileDialog saveDlg = new SaveFileDialog();
saveDlg.Filter = "bmp|*.bmp";

if (loadedImage != null)
{
    saveDlg.ShowDialog();
}

if (saveDlg.FileName != null && saveDlg.FileName != "")
{
    if (loadedImage != null)
    {
        loadedImage.Save(saveDlg.FileName);
    }
}
```

В том случае, если пользователь вместо того, чтобы открыть в предложенном ему диалоговом окне выбора файлов рисунок, выберет некоторый текстовый файл, произойдет следующая ошибка:

```
System.ArgumentException: Parameter is not valid.
```

Как и в предыдущем случае, данная ошибка была порождена синтаксически верным, но неустойчивым к подаче некорректных воздействий кодом:

```
if (openDlg.ShowDialog() == DialogResult.OK)
{
    loadedImage = new Bitmap(openDlg.FileName);
    Invalidate();
}
```

Устранить данную ошибку можно также путем внесения в код некоторых изменений. Например, такой код на языке C# уже не приведет к возникновению ошибки времени выполнения:

```
if (openDlg.ShowDialog() == DialogResult.OK)
{
    try
    {
        loadedImage = new
        Bitmap(openDlg.FileName);
        Invalidate();
    }
    catch (System.ArgumentException)
    {
        MessageBox.Show("The file
        extension is not correct", "Выбран неверный
        файл");
    }
}
```

Выводы

В данной работе был проведен обзор основных видов ошибок, возникающих в процессе разработке программного обеспечения.

Был приведен пример появления ошибки времени выполнения в .NET-приложении с графическим интерфейсом.

На основании данного примера можно сделать вывод о том, что ошибки времени выполнения являются результатом плохо продуманного, не содержащего проверок и блоков обработки исключительных ситуаций, программного кода. Данные ошибки могут быть устранены путем внесения требуемых изменений в код. Однако обнаружить

такую ошибку можно лишь путем подачи на тестируемую программу некоторых непредусмотренных спецификацией, нежелательных тестовых воздействий.

Приведенный обзор позволяет сделать вывод о необходимости решения задачи тестирования программных средств компьютерной техники, разработанных на основе платформы .NET для выявления ошибок времени выполнения.

Литература

1. Основы современного тестирования программного обеспечения, разработанного на C#: Учебное пособие / Под научной редакцией проф. В.П. Котлярова. – Санкт-Петербург, 2004. – 340 с.
2. Cem Kaner, Jack Falk, Hung Quoc Nguyen. Testing Computer Software. International Thomson Computer Press, 2001. – 430 с.
3. Биллиг В. А. Курс лекций «Основы офисного программирования и язык VBA»: WinApi. Интернет-Университет Информационных Технологий, INTUIT.ru.
4. Andrew Troelsen. Pro C# 2005 and the .NET 2.0 Platform. Berkeley, Apress, 2006. – 986 p.

Поступила в редакцию 2.03.2006

Рецензент: д-р техн. наук, проф. Г.Н. Жолткевич, Харьковский национальный университет им. В.Н. Каразина, Харьков.