

УДК 665.6

**АНАЛИЗ КАЧЕСТВА РАЗРАБОТАННЫХ ПРОГРАММНЫХ
СРЕДСТВ, ИСПОЛЬЗУЕМЫХ В НЕФТЕПЕРЕРАБАТЫВАЮЩЕМ
И НЕФТЕХИМИЧЕСКОМ ПРОИЗВОДСТВЕ**

**ANALYSIS OF QUALITY OF DEVELOPED SOFTWARE USED
IN OIL PROCESSING AND PETROCHEMICAL INDUSTRY**

Фомина В. В., Левина Т. М., Исламгулов Р. Р., Черникова В. О.

**Уфимский государственный нефтяной технический университет,
филиал, г. Салават, Российская Федерация**

V. V. Fomina, T. M. Levina, R. R. Islamgulov, V. O. Chernikova

Ufa State Petroleum Technological University,

Ufa, the Russian Federation

Аннотация. Нефтегазовая отрасль является лидирующей отраслью России по поступлениям денежных средств в бюджет. Информационные технологии активно внедряются в данную сферу для того, чтобы максимально снизить затраты на добычу и переработку нефти. Стоит отметить, что для любого крупного нефтеперерабатывающего предприятия важна не только качественная продукция на выходе, но и хранение и передача больших потоков данных. Прогресс не стоит на месте, развивается наука, технологии, появляются новые задачи. В связи с этим необходимо создание нового программного средства и совершенствование старого.

Каждый программный продукт имеет определенные, присущие только ему, черты и особенности, которые могут обеспечивать те или иные преимущества или недостатки. В связи с резким увеличением сложности и размеров современных программных комплексов при одновременном

росте ответственности выполняемых функций резко повысились требования со стороны заказчиков и пользователей к их качеству и безопасности применения. Для обеспечения высокой эффективности и качества функционирования программ и программных комплексов применяются международные стандарты, разработанные при участии представителей ведущих компаний отрасли разработки программного обеспечения.

Ключевым моментом статьи является проверка разрабатываемых программных средств на семантические ошибки, описание программы в соответствии с ГОСТ 28806-90 «Качество программных средств. Термины и определения». Важность своевременной проверки программного средства на различные ошибки очень велика, так как они могут нанести существенный ущерб, значительно превышающий положительный эффект от их использования.

В статье рассмотрена важность правильного употребления IT терминов и выражений. Рассмотрены статический и динамический анализы исходного кода программного обеспечения. Рассмотрены достоинства и недостатки для каждого из видов анализа, которые представлены в таблице. Проведен статический анализ двух программных продуктов с помощью анализатора PVS-Studio. На основе проведенного анализа сделаны соответствующие выводы.

Abstract. The oil and gas industry in Russia is the leading industry in terms of cash inflows to the budget. Information technologies are actively being introduced into this sphere in order to minimize the cost of oil production. It should be noted that for any large oil refinery, not only quality output is important, but also the storage and transmission of large data flows. Progress does not stand still, science and technology are developing, new tasks are emerging. In this connection, it is necessary to create new software and improve the old one.

Each software product has the right, inherent only to it, features and features that can provide certain advantages or disadvantages. In connection with the sharp advantage and a lot of modern software systems with the simultaneous fulfillment of the requirements imposed on their quality and safety. To ensure high efficiency and quality of software and software complexes, international standards are developed, developed with the participation of representatives of the leading companies in the software development industry.

The key point of the article is checking the developed software tools for semantic errors, the description of programs in accordance with GOST 28806-90 "Quality of software. Terms and Definitions". The importance of timely software verification on various errors, as they can cause significant damage, far exceeding the positive effect of their use.

The article considers the importance of the correct use of IT terms and expressions. Static and dynamic analyzes of the software source code are considered. The advantages and disadvantages for each of the types of analysis that are presented in the table are considered. The advantages and disadvantages for each of the types of analysis that are presented in the table are considered. A static analysis of two software products was performed using the PVS-Studio analyzer. Based on the analysis, the relevant conclusions are drawn.

Ключевые слова: программное обеспечение, семантика, семантические ошибки, семантический анализ, программный код, описание программы.

Key words: software, semantics, semantic error, semantic analysis, program code, program description.

Специалисты, использующие программные средства в отрасли нефтехимии и нефтепереработки сталкиваются с написанием различных программных продуктов для конкретных задач предприятия, чаще всего, не учитывают их основные свойства, например гибкость, надежность,

качество и т.д. [1-13]. На сегодняшний день существует большое множество IT терминов и понятий. Зачастую многие путаются в терминах, понимая под одним термином несколько смысловых значений. В ходе проведения опроса среди специалистов, работающих с IT в нефтехимической отрасли, были получены следующие результаты (рисунок 1). В опросе были такие определения как: приложение, база данных, автоматизированное рабочее место (АРМ), архитектура и другие.

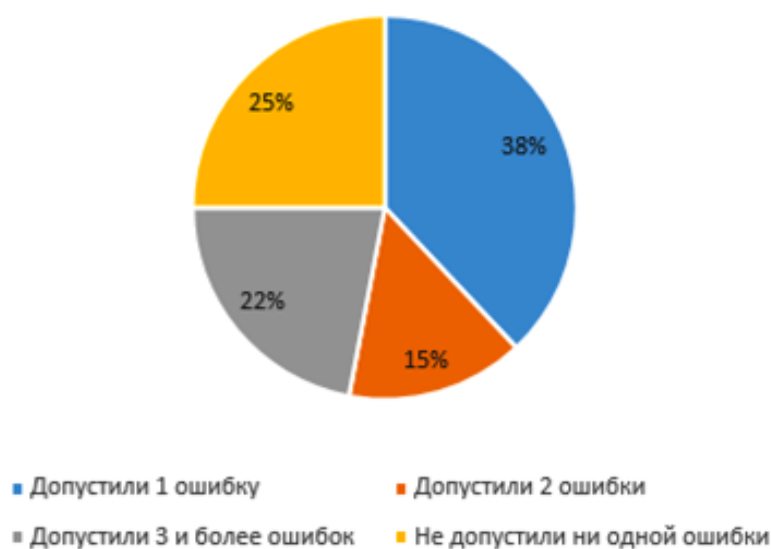


Рисунок 1. Результат опроса

По результатам опроса, можно сделать вывод, что большинство специалистов в какой-то мере ошибаются в определениях. Однако существует множество стандартов, которые однозначно характеризуют тот или иной термин и позволяют создать качественный программный продукт. Основными характеристиками качества программного средства, согласно ГОСТ 28806-90 «Качество программных средств. Термины и определения», являются функциональность, надежность, удобство использования, эффективность, сопровождаемость и мобильность [14].

Функциональность – совокупность свойств программного средства, определяемая наличием и конкретными особенностями набора функций, способных удовлетворять заданные или подразумеваемые потребности.

Надежность – совокупность свойств, характеризующая способность программного средства сохранять заданный уровень пригодности в заданных условиях в течение заданного интервала времени.

Удобство использования – совокупность свойств программного средства, характеризующая усилия, необходимые для его использования, и индивидуальную оценку результатов его использования заданным или подразумеваемым кругом пользователей программного средства.

Эффективность – совокупность свойств программного средства, характеризующая те аспекты его уровня пригодности, которые связаны с характером и временем использования ресурсов, необходимых для заданных условий функционирования.

Сопровождаемость – совокупность свойств программного средства, характеризующая усилия, которые необходимы для его модификации.

Мобильность – совокупность свойств программного средства, характеризующая приспособленность для переноса из одной среды функционирования в другие. Также для достижения качества программного средства необходимо уделять внимание семантике кода. Семантика в программировании – это система правил определения поведения отдельных языковых конструкций. Семантика определяет смысловое значение предложений алгоритмического языка [15]. Рассмотрим виды семантического анализа.

Статический анализ используется в практике программных разработок почти столь же долго, как существует сама разработка программного обеспечения. На этапе статического анализа обнаруживаются и описываются области исходного кода со слабыми местами, включая скрытые уязвимости, логические ошибки, дефекты реализации, некорректности при выполнении параллельных операций, редко возникающие граничные условия и многие другие проблемы [16].

Инструменты динамического анализа обнаруживают программные ошибки в коде, запущенном на исполнение. При этом разработчик имеет

возможность наблюдать или диагностировать поведение приложения во время его исполнения, в идеальном случае – непосредственно в целевой среде.

Технология динамического анализа включает в себя:

Размещение вставок в исходный код на этапе препроцессорной обработки – в исходный текст приложения до начала компиляции вставляется специальный фрагмент кода для обнаружения ошибок. Размещение вставок в объектный код – для такого инструмента динамического анализа необходимо обладать достаточными знаниями относительно среды исполнения, чтобы иметь возможность вставлять код непосредственно в исполняемые файлы и библиотеки. Вставка кода во время компиляции – разработчик использует специальные ключи (опции) компилятора для внедрения в исходный код. Специализированные библиотеки этапа исполнения – для обнаружения ошибок в передаваемых параметрах разработчик использует отладочные версии системных библиотек.

Таблица 1. Виды семантического анализа

Виды	Достоинства	Недостатки
Статический	Используется на ранних этапах жизненного цикла ПО; Можно анализировать существующие базы кодов, которые уже прошли тестирование; Средства могут быть интегрированы в среду разработки в качестве части компонента; Низкие затраты.	Возможно обнаружение программных ошибок и уязвимостей, которые не обязательно приводят к отказу программы или воздействию на ее поведение; Ненулевая вероятность "ложного срабатывания".
Динамический	Редко возникают "ложные срабатывания" – высокая продуктивность по нахождению ошибок; Для отслеживания причины ошибки может быть произведена полная трассировка стека и среды исполнения; Захватываются ошибки в контексте работающей системы.	Происходит вмешательство в поведение системы в реальном времени; Полнота анализа ошибок зависит от степени покрытия кода.

Для проведения семантического анализа программного кода на языках C/C++/C# существуют специальные программы, такие как PVS-Studio, Blast, Coverity.

Berkeley Lazy Abstraction Software Verification Tool (BLAST) – программа проверки моделей для языка Си. Задача, решаемая инструментом BLAST — это проверка того, что программа удовлетворяет поведенческим требованиям к ней. BLAST реализует подход абстракции и уточнения по контрпримерам для конструирования абстрактной модели, которая затем проверяется на свойства безопасности. Абстракция строится по ходу анализа и только до требуемой точности, устанавливаемой в ходе анализа.

Coverity – пакет программного обеспечения, состоящий из статического и динамического анализаторов кода, принадлежащий компании Synopsys. Программное обеспечение ищет ошибки и недочёты в безопасности исходных кодов программ, написанных на Си, C++, Java, C# и JavaScript.

PVS-Studio может обнаруживать большое количество ошибок. Среди них: арифметическое переполнение, потеря значимости, выход за границу массива, неинициализированные переменные, неиспользуемые переменные, неправильная работа с типами, неправильное представление о работе функции/класса, опечатки, не совпадение оформления кода с логикой его работы, ошибки из-за copy-paste, ошибки при работе с исключениями, проблемы безопасности, путаница с приоритетом операций, разыменование нулевого указателя/нулевой ссылки, разыменование параметров без предварительной проверки, ошибки синхронизации, неточное сравнение, ошибки при использовании WPF, целочисленное деление на 0. Проведен анализ качества двух разработанных программных средств, применяемых на производствах нефтехимии и нефтепереработки, с помощью программы PVS-Studio который выполняет статический анализ кода и генерирует отчёт,

помогающий программисту находить и устранять ошибки, и интегрируется в среду разработки Visual Studio 2010-2017.

Первое программное средство применяется для определения характеристик волоконно-оптических элементов, которые позволяют создать современные высокочувствительные измерительные системы для контроля широкого круга физических величин (электрический ток, давление, температура и так далее) химико-технологических процессов нефтехимии и нефтепереработки. Архитектура разработанного программного средства представлена на рисунке 2.

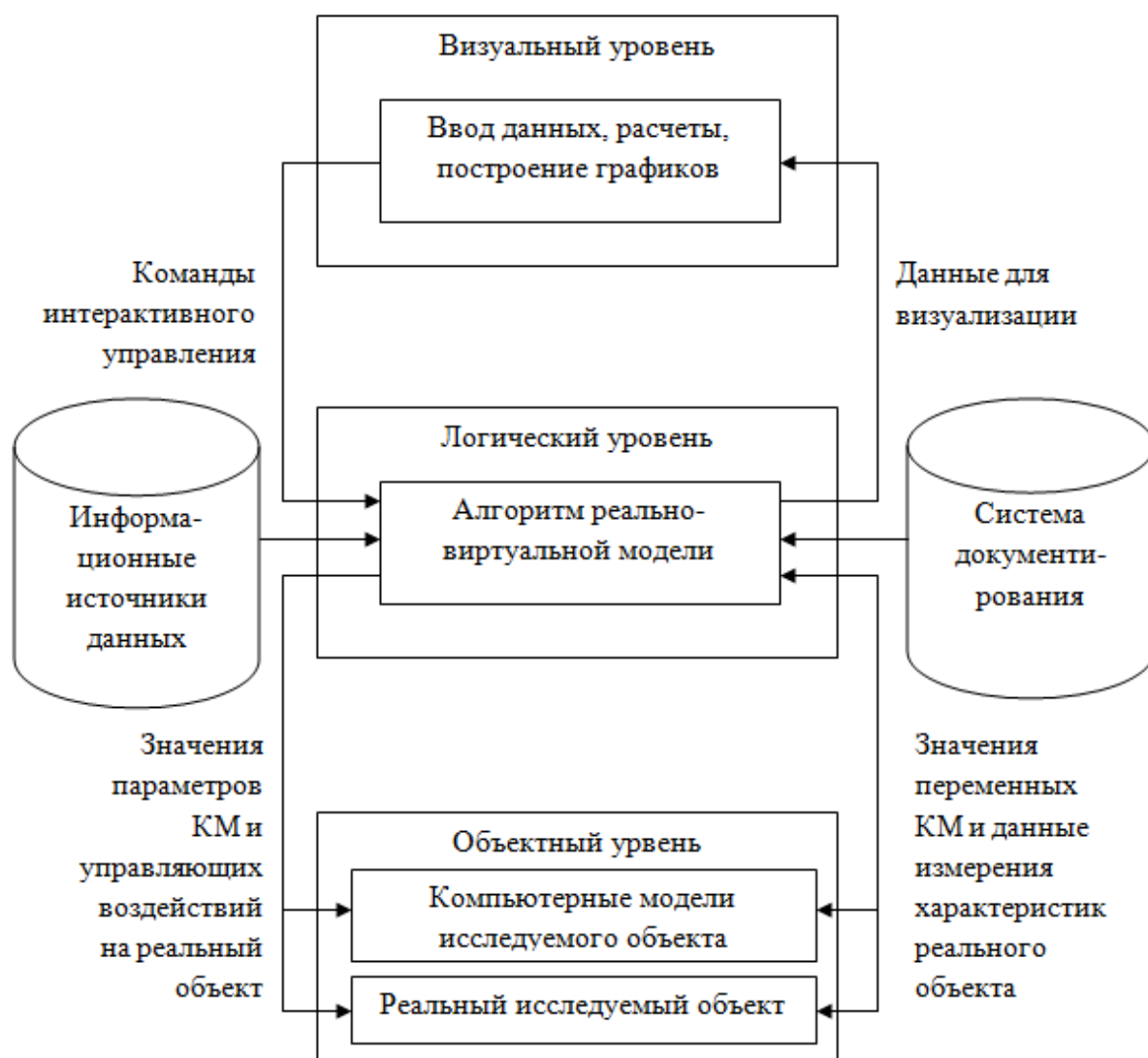


Рисунок 2. Архитектура программного средства для определения характеристик волоконно-оптических элементов

Визуальный уровень включает в себя средства визуализации результатов расчетов и экспериментов, а также средства интерактивного изменения варьируемых параметров. На логическом уровне формируются алгоритмы выполнения реально-виртуальных экспериментов с использованием значений, поступающих из информационных источников данных, и возможностью формирования документов на основе интерактивных отчетных форм. На объектном уровне располагаются средства интеграции с реальным техническим объектом и его компьютерной моделью.

Принцип работы программного средства заключается в следующем. При запуске программы на экран выходит главная форма, одним компонентом которой являются четыре ячейки для ввода или выбора данных. Они позволяют определить оптимальные условия движения света в волоконно – оптическом световоде. Известно, что основными периметрами являются сердцевина или внутренняя поверхность световода, которая имеет высокий показатель преломления, однородную в направлении z , а также окружающую ее оболочку с меньшим показателем преломления. При этом свет распространяется по сердцевине в направлении z [17].

Сечение волновода в плоскости xz представлено на рисунке 3. Оптический луч, как видно из рисунка, распространяется по световоду, испытывая многократное полное отражение от границы раздела между сердцевиной и оболочкой, однако, когда угол Θ становится слишком велик, полного отражения не происходит и свет проникает в оболочку [18].

Максимальный угол Θ_c , при котором происходит полное отражение, определяется формулой:

$$\Theta_c = \arccos\left(\frac{n_2}{n_1}\right) = \arcsin\sqrt{\frac{n_1^2 - n_2^2}{n_1^2}} = \arcsin\sqrt{2\Delta}, \quad (1)$$

где Δ - удельная разность показателей преломления и определяется через показатель преломления сердцевины и оболочки:

$$\Delta = \frac{n_1^2 - n_2^2}{2n_1^2} \cong \frac{n_1 - n_2}{n_1}.$$

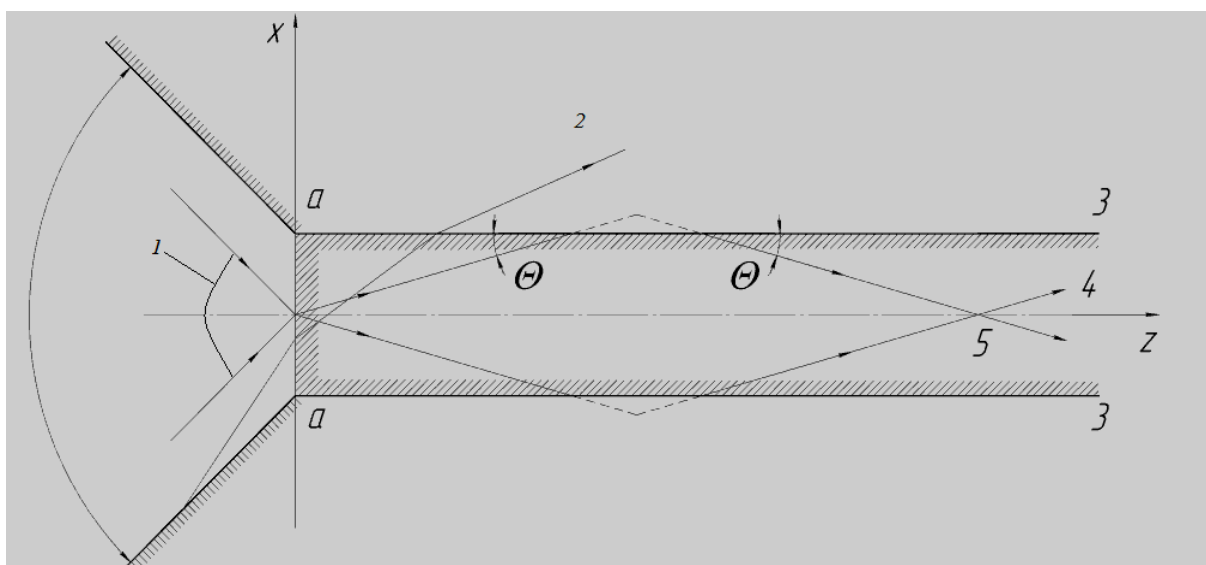


Рисунок 3. Распространение света, падающего на световод:
 1 – угол приема лучей $2\Theta_{max}$; 2 – свет, который не передается по световоду ($\Theta > \Theta_c$); 3 – оболочка n_2 ; 4 – сердцевина n_1 ;
 5 – распространяемый свет

Луч света, который распространяется в световоде, отражаясь от границы раздела под максимальным углом Θ_c (формула 1), при вводе в световод падает на его торец под еще большим углом Θ_{max} ,

$$\Theta_{max} = \arcsin(n_1 \sin \Theta_c) = \arcsin \sqrt{n_1^2 - n_2^2} \quad (2)$$

Этот угол является максимальным углом, при котором возможен ввод света в световод или вывод из него. В формуле (2) величина, стоящая в скобках, является числовой апертурой

$$NA = n_1 \sqrt{2\Delta}$$

Оптический луч распространяется в световоде, испытывая многократное полное отражение от границы раздела сердцевины и оболочки. Однако из этого не следует, что угол Θ может быть практически произвольным. Наоборот, угол распространения луча Θ является особенным, характеристическим углом [19].

После того, как пользователь вводит данные и нажимает на кнопку, на экране выводится графики функций:

- зависимости коэффициента преломления n от длины волны λ ;
- зависимости угла падения от коэффициента преломления n_1 ;
- зависимости угла поворота плоскости поляризации от длины волны λ .

Скриншот окна главной формы представлен на рисунке 4.

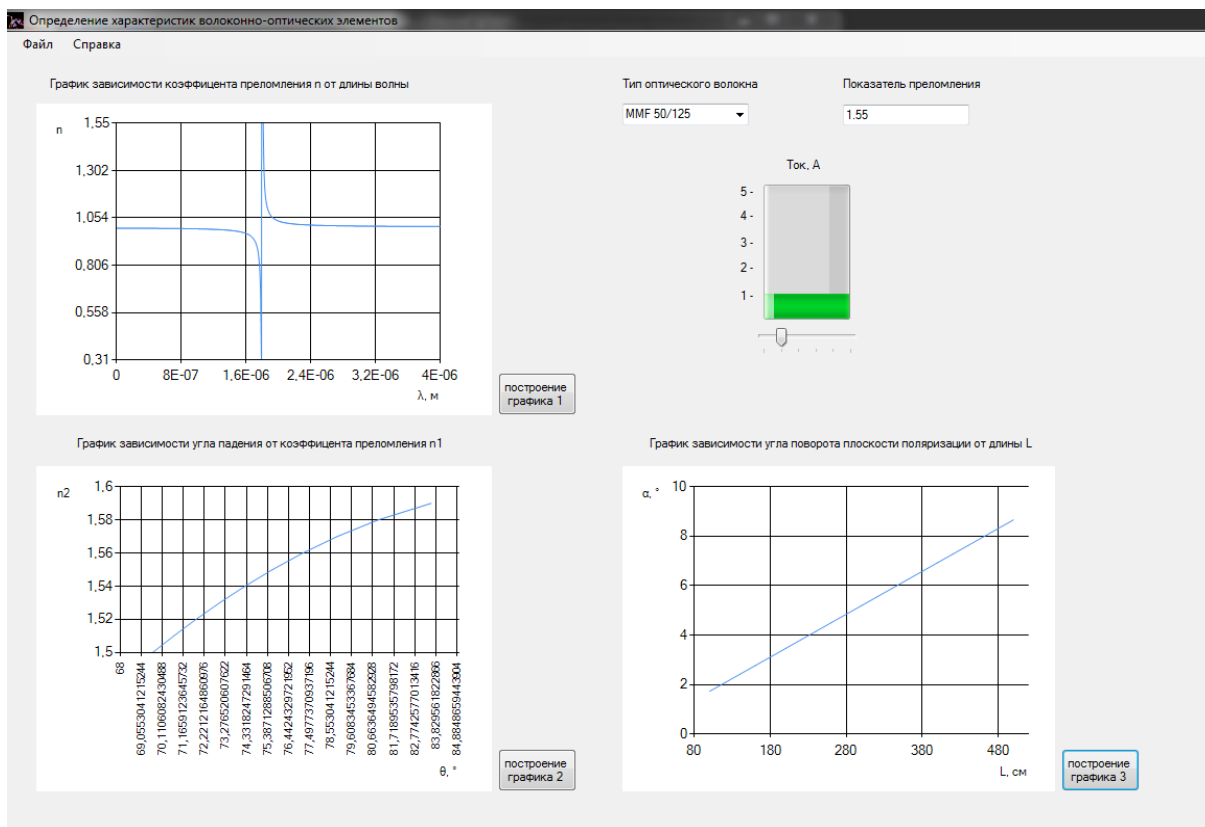


Рисунок 4. Скриншот окна главной формы

Часть кода данного программного средства представлена ниже.

```

if (n1 == 1.6)
{
    for (double i = 1.5; i < 1.6; i = i + 0.01)
    {
        chart2.ChartAreas[0].AxisX.Maximum = 85;
        chart2.ChartAreas[0].AxisX.Minimum = 68;
        chart2.ChartAreas[0].AxisY.Maximum = 1.6;
        chart2.ChartAreas[0].AxisY.Minimum = 1.5;

        chart2.Series[0].Points.AddXY(((Asin(i / n1)) * 180 / PI), i);
    }
}
else if (n1==1.65)
{
    for (double i = 1.55; i < 1.65; i = i + 0.01)
    {

```

```

        chart2.ChartAreas[0].AxisX.Maximum = 85;
        chart2.ChartAreas[0].AxisX.Minimum = 68;
        chart2.ChartAreas[0].AxisY.Maximum = 1.65;
        chart2.ChartAreas[0].AxisY.Minimum = 1.55;
        chart2.Series[0].Points.AddXY(((Asin(i / n1)) * 180 / PI), i);
    }
}
else if (n1 == 1.7)
{
    for (double i = 1.5; i < 1.7; i = i + 0.01)
    {
        chart2.ChartAreas[0].AxisX.Maximum = 84;
        chart2.ChartAreas[0].AxisX.Minimum = 60;
        chart2.ChartAreas[0].AxisY.Maximum = 1.7;
        chart2.ChartAreas[0].AxisY.Minimum = 1.5;
        chart2.Series[0].Points.AddXY(((Asin(i / n1)) * 180 / PI), i);
    }
}
else if (n1 == 1.55)
{
    for (double i = 1.45; i < 1.55; i = i + 0.02)
    {
        chart2.ChartAreas[0].AxisX.Maximum = 82;
        chart2.ChartAreas[0].AxisX.Minimum = 69;
        chart2.ChartAreas[0].AxisY.Maximum = 1.55;
        chart2.ChartAreas[0].AxisY.Minimum = 1.45;
        chart2.Series[0].Points.AddXY(((Asin(i / n1)) * 180 / PI), i);
    }
}
}
}

```

В результате анализа исходного кода первого программного средства с помощью PVS-Studio, найдены пять ошибок неточного сравнения (рисунок 5). В результате выявления этих ошибок в программу вносятся изменения. Затем процедура проверки качества программы повторяется до тех пор, пока все ошибки не будут устранены [20-23].

Code	Message	File	Line
V3024	An odd precise comparison: n1 == 1.6. Consider using a comparison with defined precision: Math.Abs(A - B) < Epsilon.	Form1.cs	123
V3024	An odd precise comparison: n1 == 1.65. Consider using a comparison with defined precision: Math.Abs(A - B) < Epsilon.	Form1.cs	135
V3024	An odd precise comparison: n1 == 1.7. Consider using a comparison with defined precision: Math.Abs(A - B) < Epsilon.	Form1.cs	146
V3024	An odd precise comparison: n1 == 1.55. Consider using a comparison with defined precision: Math.Abs(A - B) < Epsilon.	Form1.cs	157
V3024	An odd precise comparison: n1 == 1.5. Consider using a comparison with defined precision: Math.Abs(A - B) < Epsilon.	Form1.cs	168

Рисунок 5. Результат анализа первого программного продукта

Второе программное средство применяется для работы с показаниями различных датчиков физических величин.

Проведен анализ клиентской части программного средства для мониторинга показаний датчиков. Клиентская часть разработанного программного средства отвечает за принятие данных с сервера, сохранение

в локальной базе данных и представление их в удобочитаемом виде для пользователя. Программное средство написано на языке программирования С#. В случае некорректной работы программного средства могло иметь место следующие ошибки: потеря данных, неправильное отображение данных и т.д. Скриншот окна главной формы программы представлен на рисунке 6.

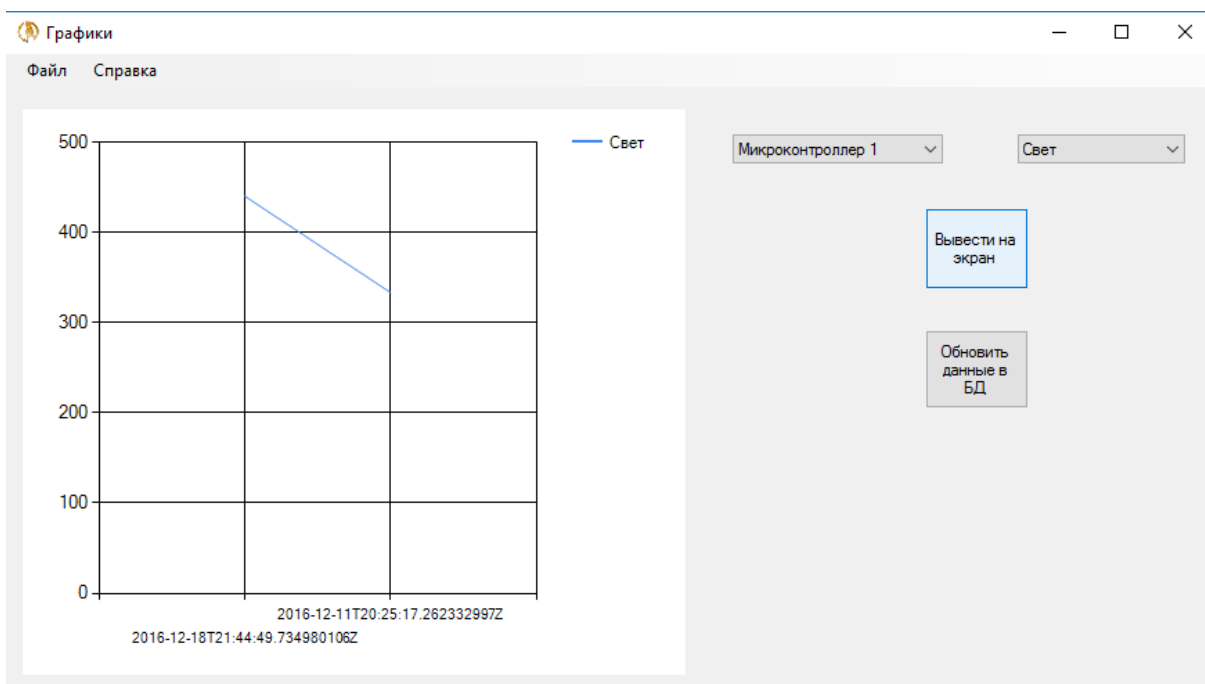


Рисунок 6. Скриншот главной формы программного средства применяющегося для работы с показаниями различных датчиков физических величин

Часть кода данного комплекса представлена ниже.

```
private void button3_Click(object sender, EventArgs e)
{
    string jsonString;
    string site = "http://78.155.219.175/get";

    HttpWebRequest req = (HttpWebRequest)WebRequest.Create(site);
    HttpWebResponse resp = (HttpWebResponse)req.GetResponse();

    using (StreamReader stream = new StreamReader(
        resp.GetResponseStream(), Encoding.UTF8))
    {
        jsonString = stream.ReadToEnd();
    }

    JArray a = JArray.Parse(jsonString);

    foreach (JObject o in a.Children<JObject>())
    {
```

```
foreach (JProperty p in o.Properties())
{
    if (p.Name == "id")
        id = (int)p.Value;
    if (p.Name == "mic")
        mic = p.Value.ToString();
    if (p.Name == "time")
        time = p.Value.ToString();
    if (p.Name == "type")
```

В результате анализа исходного кода второго программного средства с помощью PVS-Studio ошибки не обнаружены (рисунок 7).

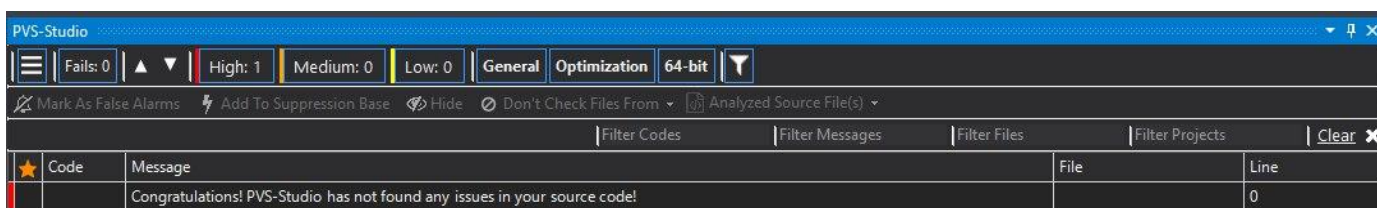


Рисунок 7. Результат анализа второго программного продукта

По результатам анализа качества исходных кодов двух разработанных программных средств, используемых в нефтеперерабатывающем и нефтехимическом производстве можно сделать вывод, что любое создаваемое программное средство должно быть хорошо продуманно на ранних этапах разработки и протестировано по завершении разработки, дабы избежать ошибок, которые скажутся в дальнейшем на работе программы в целом.

Список используемых источников

1 Fomina V. V., Zharinov Yu. A., Abyzgildin A.Yu. Building up the hybrid graph of the oil chemical production on the basis of the graphical model of the process // Нефтегазовое дело. 2016. Т. 14, №1. С.157-164.

2 Фомина В. В., Абызгильдин А. Ю. Метод оптимизации нефтехимических процессов с применением графических моделей технологических систем // Нефтегазовое дело. 2016. Т. 14, № 3. С. 141-144.

3 Fomina V. V., Zharinov Yu. A., Abyzgildin A. Yu. Graphical modeling method application in construction of petrochemical plant bloc-diagram // Oil and Gas Business. 2014. № 1, pp. 263-272. URL: http://ogbus.ru/eng/authors/FominaVV/FominaVV_1.pdf.

4 Фомина В. В., Катков А. Н., Абызгильдин А. Ю. Анализ структуры и построение графической модели сложного нефтехимического комплекса // Нефтепереработка и нефтехимия. Научно-технические достижения и передовой опыт. 2010. №2. С. 33-39.

5 Катков А. Н., Фомина В. В., Абызгильдин А. Ю. Анализ и оптимизация технологического процесса с применением метода графического моделирования // Нефтепереработка и нефтехимия. Научно-технические достижения и передовой опыт. 2010. №3. С. 30-33.

6 Фомина В. В., Абызгильдин А. Ю. Представление блок-схемы завода нефтехимического профиля методом графического моделирования // Нефтепереработка и нефтехимия. Научно-технические достижения и передовой опыт. 2008. №3. С. 38-42.

7 Ферлюдин Ю. П., Фомина В. В., Абызгильдин А. Ю. Блок-схема ОАО «Салаватнефтеоргсинтез», построенная методом графического моделирования // Нефтепереработка и нефтехимия. Научно-технические достижения и передовой опыт. 2008. №4-5. С. 117-120.

8 Абызгильдин А. Ю., Фомина В. В. Графическая модель блок-схемы нефтеперерабатывающего завода ОАО «Салаватнефтеоргсинтез» // Нефтегазовое дело. 2008. Т. 6, № 1. С. 44-46.

9 Абызгильдин А. Ю., Фомина В. В., Рахимова З. Ф. Графическое представление сложного нефтехимического комплекса методом графических моделей // Нефтегазовое дело. 2008. Т. 6, № 2. С. 137-140.

10 Фомина В. В. Анализ структуры нефтехимического комплекса и оптимизация технологических схем с применением метода графических моделей: дис... канд. техн. наук . Уфа, 2009.

11 Фомина В. В. Анализ структуры нефтехимического комплекса и оптимизация технологических схем с применением метода графических моделей: автореф. дис. канд. техн. наук. Уфа, 2009.

12 Абызгильдин А. Ю., Фомина В. В. Графические модели процессов первичной переработки, термических, термокаталитических и термогидрокаталитических процессов переработки нефти и газа. Уфа: УГНТУ, 2016. 87 с.

13 Абызгильдин А. Ю., Фомина В. В. Графические модели процессов переработки масел. Уфа: УГНТУ, 2016. 88 с.

14 ГОСТ 28806-90 «Качество программных средств. Термины и определения». М.: Стандартинформ, 2005.

15 Лунева Н. Н., Левина Т. М. Программный инструмент для принятия управленческих решений // Вестник экономики и менеджмента. 2016. №3. С.31-36

16 Ураксеев М. А., Левина Т. М., Шамаев Ф. Ф., Кулябин А. С. Разработка волоконно-оптических систем для учета, мониторинга и прогнозирования работы высоковольтного оборудования в СУБД с web-интерфейсом // Электротехнические и информационные комплексы и системы. 2015. Т 11, №1. С. 97-103

17 Ураксеев М. А., Левина Т. М. Применение магнитооптического элемента фарадея в информационно-измерительных системах контроля магнитного поля и электрического тока // Прикаспийский журнал: управление и высокие технологии. 2008. № 2. С. 24-31.

18 Ураксеев М. А., Левина Т. М. Оптоволоконные трансформаторы как элементы современных электротехнических комплексов и систем // Электротехнические и информационные комплексы и системы. 2013. Т 9. №2. С. 23-29.

19 Filippov V. N., Scharafiev R., Galiakbarova E., Sultanova E., Efimenko E. Systems clearing constructions enterprises of oil refining and petrochemicals the republic of bashkortostan at the present stage of planning of protection water resources // Oriental Journal of Chemistry. 2015. Т. 31, № 5, pp. 157-163.

20 Левина Т. М., Бажанова Т. В. Удаленное предоставление производственной информации по запросу пользователя //Сборник тезисов докладов VI науч.-техн. конф. молодых специалистов ООО "БашНИПИнефть". 2016. С. 115-116.

21 Лунева Н. Н., Левина Т. М. Применение информационных технологий в управлении и организации проведения ремонтов на предприятиях нефтехимии и нефтепереработки //Вестник экономики и менеджмента. 2016. № 3. С. 26-31.

22 Левина Т. М., Каримов Р. Р. Алгоритм учета некорректных данных оборудования как способ повышения техносферной безопасности // Инновационные научные исследования: теория, методология, практика. 2016. С. 40-43 с.

23 Ураксеев М. А., Левина Т. М., Бажанова Т. В. Алгоритм работы цифрового устройства интегрирования в измерительном комплексе контроля электрического тока и магнитного поля // Электротехнические и информационные комплексы и системы. 2016. Т. 12. № 2. С. 80-84.

References

1 Fomina V. V., Zharinov Yu. A., Abyzgildin A.Yu. Building up the hybrid graph of the oil chemical production on the basis of the graphical model of the process // Neftegazovoe delo. 2016. Т. 14, №1. S.157-164. [in Russian].

2 Fomina V. V., Abyzgil'din A. Ju. Metod optimizacii neftehimicheskikh processov s primeneniem graficheskikh modelej tehnologicheskikh sistem // Neftegazovoe delo. 2016. Т. 14, № 3. S. 141-144. [in Russian].

3 Fomina V. V., Zharinov Yu. A., Abyzgil'din A. Yu. Graphical modeling method application in construction of petrochemical plant bloc-diagram // Oil and Gas Business. 2014. № 1, pp. 263-272. URL: http://ogbus.ru/eng/authors/FominaVV/FominaVV_1.pdf. [in Russian].

4 Fomina V. V., Katkov A. N., Abyzgil'din A. Ju. Analiz struktury i postroenie graficheskoy modeli slozhnogo neftehimicheskogo kompleksa // Neftepererabotka i neftehimija. Nauchno-tehnicheskie dostizhenija i peredovoj opyt. 2010. №2. S. 33-39. [in Russian].

5 Katkov A. N., Fomina V. V., Abyzgil'din A. Ju. Analiz i optimizacija tehnologicheskogo processa s primeneniem metoda graficheskogo modelirovkanija // Neftepererabotka i neftehimija. Nauchno-tehnicheskie dostizhenija i peredovoj opyt. 2010. №3. S. 30-33. [in Russian].

6 Fomina V. V., Abyzgil'din A. Ju. Predstavlenie blok-shemy zavoda neftehimicheskogo profilja metodom graficheskogo modelirovanija // Neftepererabotka i neftehimija. Nauchno-tehnicheskie dostizhenija i peredovoj opyt. 2008. №3. S. 38-42. [in Russian].

7 Ferljudin Ju. P., Fomina V. V., Abyzgil'din A. Ju. Blok-shema OAO «Salavatnefteorgsintez», postroennaja metodom graficheskogo modelirovanija // Neftepererabotka i neftehimija. Nauchno-tehnicheskie dostizhenija i peredovoj opyt. 2008. №4-5. S. 117-120. [in Russian].

8 Abyzgil'din A. Ju., Fomina V. V. Graficheskaja model' blok-shemy neftepererabatyvajushhego zavoda OAO «Salavatnefteorgsintez» // Neftegazovoe delo. 2008. T. 6, № 1. S. 44-46.

9 Abyzgil'din A. Ju., Fomina V. V., Rahimova Z. F. Graficheskoe predstavlenie slozhnogo neftehimicheskogo kompleksa metodom graficheskikh modelej // Neftegazovoe delo. 2008. T. 6, № 2. S. 137-140. [in Russian].

10 Fomina V. V. Analiz struktury neftehimicheskogo kompleksa i optimizacija tehnologicheskikh shem s primeneniem metoda graficheskikh modelej: dis... kand. tehn. nauk . Ufa, 2009. [in Russian].

11 Fomina V. V. Analiz struktury neftehimicheskogo kompleksa i optimizacija tehnologicheskikh shem s primeneniem metoda graficheskikh modelej: avtoref. dis. kand. tehn. nauk. Ufa, 2009. [in Russian].

12 Abyzgil'din A. Ju., Fomina V. V. Graficheskie modeli processov pervichnoj pererabotki, termicheskikh, termokataliticheskikh i termogidrokataliticheskikh processov pererabotki nefti i gaza. Ufa: UGNTU, 2016. 87 s. [in Russian].

13 Abyzgil'din A. Ju., Fomina V. V. Graficheskie modeli processov pererabotki masel. Ufa: UGNTU, 2016. 88 s. [in Russian].

14 GOST 28806-90 «Kachestvo programmnyh sredstv. Terminy i opredelenija». M.: Standartinform, 2005. [in Russian].

15 Luneva N. N., Levina T. M. Programmnyj instrument dlja prinjatija upravlencheskikh reshenij // Vestnik jekonomiki i menedzhmenta. 2016. №3. S.31-36. [in Russian].

16 Urakseev M. A., Levina T. M., Shamaev F. F., Kuljabin A. S. Razrabotka volokonno-opticheskikh sistem dlja ucheta, monitoringa i prognozirovaniya raboty vysokovol'tnogo oborudovaniya v SUBD s web-interfejsom // Jelektrotehnicheskie i informacionnye kompleksy i sistemy. 2015. T 11, №1. S. 97-103. [in Russian].

17 Urakseev M. A., Levina T. M. Primenenie magnitoopticheskogo jelementa faradeja v informacionno-izmeritel'nyh sistemah kontrolja magnitnogo polja i jelektricheskogo toka // Prikaspijskij zhurnal: upravlenie i vysokie tehnologii. 2008. № 2. S. 24-31. [in Russian].

18 Urakseev M. A., Levina T. M. Optovolokonnye transformatory kak jelementy sovremennyh jelektrotehnicheskikh kompleksov i sistem // Jelektrotehnicheskie i informacionnye kompleksy i sistemy. 2013. T 9. №2. S. 23-29. [in Russian].

19 Filippov V. N., Scharafiev R., Galiakbarova E., Sultanova E., Efimenko E. Systems clearing constructions enterprises of oil refining and petrochemicals the republic of bashkortostan at the present stage of planning of protection water resources // Oriental Journal of Chemistry. 2015. T. 31, № 5, pp. 157-163. [in Russian].

20 Levina T. M., Bazhanova T. V. Udalennoe predostavlenie proizvodstvennoj informacii po zaprosu pol'zovatelja //Sbornik tezisov dokladov VI nauch.-tehn. konf. molodyh specialistov OOO "BashNIPIneft". 2016. S. 115-116. [in Russian].

21 Luneva N. N., Levina T. M. Primenenie informacionnyh tehnologij v upravlenii i organizacii provedenija remontov na predpriyatijah neftehimii i neftepererabotki //Vestnik jekonomiki i menedzhmenta. 2016. № 3. S. 26-31. [in Russian].

22 Levina T. M., Karimov R. R. Algoritm ucheta nekorrektnyh dannyh oborudovanija kak sposob povyshenija tehnosfernoj bezopasnosti // Innovacionnye nauchnye issledovanija: teorija, metodologija, praktika. 2016. S. 40-43 s. [in Russian].

23 Urakseev M. A., Levina T. M., Bazhanova T. V. Algoritm raboty cifrovogo ustrojstva integrirovaniya v izmeritel'nom komplekse kontrolja jelektricheskogo toka i magnitnogo polja // Jelektrotehnicheskie i informacionnye kompleksy i sistemy. 2016. T. 12. № 2. S. 80-84. [in Russian].

Сведения об авторах

About the authors

Фомина В. В., канд. техн. наук, доцент кафедры «Химико-технологические процессы» ФГБОУ ВО «УГНТУ», филиал, г. Салават, Российская Федерация.

V. V. Fomina, Candidate of Engineering Sciences, Associate Professor of the Chair «Chemical and Technological Processes», FSBEI HE «USPTU», Branch, Salavat, Russian Federation.

e-mail: asterhttp@mail.ru

Левина Т. М., канд. техн. наук, доцент кафедры «Общенаучные дисциплины», ФГБОУ ВО «УГНТУ», филиал, г. Салават, Российская Федерация.

T. M. Levina, Candidate of Engineering Sciences, Associate Professor of the Chair «General Scientific Disciplines», FSBEI HE «USPTU», Branch, Salavat Russian Federation.

e-mail: tattin76@mail.ru

Исламгулов Р. Р., студент, ФГБОУ ВО «УГНТУ», филиал, г. Салават, Российская Федерация.

R. R. Islamgulov, student, FSBEI HE «USPTU», Branch, Salavat, Russian Federation.

Черникова В. О., студент, ФГБОУ ВО «УГНТУ», филиал, г. Салават, Российская Федерация.

V. O. Chernikova, student, FSBEI HE «USPTU», Branch, Salavat Russian Federation.