

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Д09.03.04-ИИ.22-20/6251.КП

**Кафедра** прикладной математики и  
искусственного интеллекта

КУРСОВОЙ ПРОЕКТ

по дисциплине «Архитектура и проектирование программного обеспечения»

Тема: «Шаблон проектирования Iterator для BFT»

Руководители:

\_\_\_\_\_ ст. преп. О.А.Гудаев  
(дата, подпись)

Нормоконтроль:

\_\_\_\_\_ асс. Н.П. Пулинец  
(дата, подпись)

Исполнитель:

\_\_\_\_\_ ст. гр. ПИ-20г В.А. Пустовой  
(дата, подпись)

**ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Факультет:** *Интеллектуальных систем и программирования*  
**Направление:** *Программная инженерия*  
**Профиль:** *Искусственный интеллект*  
**Кафедра:** *«Прикладной математики и искусственного интеллекта»*

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ**

по дисциплине «Архитектура и проектирование программного обеспечения»

Студенту Пустовому Вадиму Александровичу группы ПИ-20г  
(фамилия, имя, отчество)

Тема проекта: «Шаблон проектирования Iterator для BFT»

Исходные данные к проекту: Спецификация шаблонов проектирования, спецификация графического языка Unified Modeling Language, спецификация нотации Brain Fogic Tag

Перечень искомых результатов: Диаграммы UML, структура шаблона Iterator, редактор BFT меток.

Рекомендуемая литература: Н. Вирт. Алгоритмы и структуры данных. / Вирт, Никлаус. - М.: СПб: Невский Диалект; 2-е изд., 2018. - 352 с.

Крэг Ларман. Применение UML 2.0 и шаблонов проектирования / 3-е изд. - М.: Вильямс, 2006. – 736 с.

М. Прайс. C# 7 и .NET Core. / Прайс, Марк. – СПб.: Питер; 3-е изд.; 2018. - 640 с.

Дата выдачи задания 02.09.2022

Дата защиты проекта 28.12.2022

**Руководители** \_\_\_\_\_  
(подпись)

**Разработчик** \_\_\_\_\_  
(подпись)

ст. преп. Гудаев О. А.  
(должность, Ф.И.О.)

Пустовой В. А.  
(Ф.И.О.)

## РЕФЕРАТ

Пояснительная записка: страниц 48, рис. 21, источников 6, прил. 2.

Целью работы является демонстрация работы заданного шаблона на примере создания приложения для считывания и редактирования BFT меток.

Для построения диаграмм используется язык диаграмм Unified Modelling Language (UML). Для реализации проекта используется язык программирования высокого уровня C#.

В ходе проектирования получены следующие результаты: исходный код протоколов классов на языке программирования высокого уровня C# и UML-диаграмма классов.

Для тестирования классов на языке высокого уровня C# была использована среда разработки «Visual Studio 2019». Для создания диаграмм были использованы программа проектирования диаграмм «StarUML».

Проектируемый редактор нотации BrainFogicTag может быть использован в учебных заведениях в качестве наглядного пособия при обучении приемам и правилам использования нотации BrainFogicTag, а также поведенческого шаблона проектирования Iterator.

ШАБЛОН ПРОЕКТИРОВАНИЯ, ПАТТЕРН, ИТЕРАТОР,  
РЕДАКТОР ТЕКСТА, BrainFogicTag, UML ДИАГРАММЫ, ДАННЫЕ

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	5
1 ОСНОВЫ ШАБЛОННОГО ПРОЕКТИРОВАНИЯ ПО.....	6
1.1 Методы проектирования и программирования.....	6
1.2 Описание шаблонов проектирования .....	7
1.3 Нотация BrainFogicTag .....	8
2 ХАРАКТЕРИСТИКА ШАБЛОНА ПРОЕКТИРОВАНИЯ ITERATOR.....	9
2.1 Общие сведения.....	9
2.2 Структура шаблона Iterator.....	10
3 ОПИСАНИЕ UML ДИАГРАММ.....	11
3.1 Диаграмма классов.....	11
3.2 Диаграмма последовательности.....	12
3.3 Диаграмма деятельности.....	13
3.4 Диаграмма компонентов.....	14
3.5 Диаграмма вариантов использования .....	15
4 РАЗРАБОТКА ПРОГРАММЫ НА ОСНОВЕ ШАБЛОНА ITERATOR.....	16
4.1 Описание классов.....	16
5 ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ .....	17
ЗАКЛЮЧЕНИЕ .....	23
ПЕРЕЧЕНЬ ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	24
Приложение А .....	25
Приложение Б.....	28

## ВВЕДЕНИЕ

Паттерн представляет определенный способ построения программного кода для решения часто встречающихся проблем проектирования. В данном случае предполагается, что есть некоторый набор общих формализованных проблем, которые довольно часто встречаются, и паттерны предоставляют ряд принципов для решения этих проблем.

Хотя идея паттернов как способ описания решения распространенных проблем в области проектирования появилась довольно давно, но их популярность стала расти во многом благодаря известной работе четырех авторов Эриха Гаммы, Ричарда Хелма, Ральфа Джонсона, Джона Влиссидеса, которая называлась: "Приемы объектно-ориентированного проектирования. Паттерны проектирования", и которая вышла в свет в 1994 году. Данная книга являлась первой масштабной попыткой описать распространенные способы проектирования программ. И со временем применение паттернов стало считаться хорошей практикой программирования.

При написании программ мы можем формализовать проблему в виде классов и объектов и связей между ними. И применить один из существующих паттернов для ее решения. У нас уже есть готовый шаблон, и нам только надо его применить в конкретной программе.

Также мышление паттернами упрощает групповую разработку программ. Зная применяемый паттерн проектирования и его основные принципы другому программисту будет проще понять его реализацию и использовать ее. В то же время не стоит применять паттерны ради самих паттернов.

Хорошая программа предполагает использование паттернов. Однако не всегда паттерны упрощают и улучшают программу. Неоправданное их использование может привести к усложнению программного кода, уменьшению его качества. Паттерн должен быть оправданным и эффективным способом решения проблемы.

## 1 ОСНОВЫ ШАБЛОННОГО ПРОЕКТИРОВАНИЯ ПО

### 1.1 Методы проектирования и программирования

Проектирование алгоритмов и программ – наиболее ответственный этап жизненного цикла программных продуктов, определяющий, насколько создаваемая программа соответствует спецификациям и требованиям со стороны конечных пользователей.

Затраты на создание, сопровождение и эксплуатацию программных продуктов, научно-технический уровень разработки, время морального устаревания и многое другое – все это также зависит от проектных решений.

Методы проектирования алгоритмов и программ очень разнообразны, их можно классифицировать по различным признакам, важнейшими из которых являются:

- степень автоматизации проектных работ;
- принятая методология процесса разработки.

По степени автоматизации проектирования алгоритмов и программ можно выделить:

- методы традиционного (неавтоматизированного) проектирования;
- методы автоматизированного проектирования.

Неавтоматизированное проектирование алгоритмов и программ преимущественно используется при разработке небольших по трудоемкости и структурной сложности программных продуктов, не требующих участия большого числа разработчиков.

Автоматизированное проектирование алгоритмов и программ возникло с необходимостью уменьшить затраты на проектные работы, сократить сроки их выполнения, создать типовые «заготовки» алгоритмов и программ, многократно тиражируемых для различных разработок, координации работ большого коллектива разработчиков, стандартизации алгоритмов и программ.

## 1.2 Описание шаблонов проектирования

Паттерн проектирования — это часто встречающееся решение определённой проблемы при проектировании архитектуры программ.

В отличие от готовых функций или библиотек, паттерн нельзя просто взять и скопировать в программу. Паттерн представляет собой не какой-то конкретный код, а общую концепцию решения той или иной проблемы, которую нужно будет ещё подстроить под нужды вашей программы.

Паттерны часто путают с алгоритмами, ведь оба понятия описывают типовые решения каких-то известных проблем. Но если алгоритм — это чёткий набор действий, то паттерн — это высокоуровневое описание решения, реализация которого может отличаться в двух разных программах.

Паттерны отличаются по уровню сложности, детализации и охвата проектируемой системы. Проводя аналогию со строительством, вы можете повысить безопасность перекрёстка, поставив светофор, а можете заменить перекрёсток целой автомобильной развязкой с подземными переходами.

Самые низкоуровневые и простые паттерны — идиомы. Они не универсальны, поскольку применимы только в рамках одного языка программирования.

Самые универсальные — архитектурные паттерны, которые можно реализовать практически на любом языке. Они нужны для проектирования всей программы, а не отдельных её элементов.

Кроме того, паттерны отличаются и предназначением. Основные группы:

- Порождающие паттерны беспокоятся о гибком создании объектов без внесения в программу лишних зависимостей.
- Структурные паттерны показывают различные способы построения связей между объектами.
- Поведенческие паттерны заботятся об эффективной коммуникации между объектами.

### 1.3 Нотация BrainFogicTag

Общей целью создания искусственной нотации BFT является улучшение коммуникации между носителями различных формальных систем. Перечислим интегральные характеристики такого процесса коммуникации: неоднозначность понимания констант формальной системы; трудность перевода между искусственными языками, процессов вывода; сложность описания процесса конструирования формальных выражений.

Таблица 1.1 – Список базовых конструкций метаязыка BFT

Литерал	Логика	Описание
1[]	<i>false</i>	Антиномия
0[]	<i>false</i>	Ложь
0[0]	<i>false</i>	Пустой список
0[.]	<i>false</i>	Не объявленная конструкция
1[1]	<i>true</i>	Истина
1[.] X	<i>true</i>	Ключевое слово X
1[X] Y	<i>true</i>	Память с адресом X и значением Y
0[X] Z	<i>true</i>	Конвертер с именем предиката X и списком термов Z, разделенных удвоенной точкой

Тег 1[.]X, интерпретируется как истина существования имени, где X – это тег, метаописание, десигнат слота фреймовой модели, онтология связанных данных. Ярлык, в виде точки, обозначает обобщение, а имя класса – это надпись X. Так как X – это символьный поток, то необходимо показать его завершение. Для указания завершения команд BFT подходит тег 0[0]. Например, ключевое слово гипертекстовой разметки 1[.] html 0[0] экранировано комбинацией {1[.], 0[0]}.

Ячейка памяти 1[X]Y, интерпретируется как истина, где X – это идентификатор, а Y – это символьный текст, размещённый в памяти.

Конвертер 0[X]Z, интерпретируется как истинный предикат. Где X – это идентификатор, а Z – это список тегов метаописания, разделенных символом удвоенной точки "..". Если Z с одним тегом, то это атомарный предикат или конвертор косвенной адресации.



## 2 ХАРАКТЕРИСТИКА ШАБЛОНА ПРОЕКТИРОВАНИЯ ITERATOR

### 2.1 Общие сведения

Iterator – поведенческий шаблон проектирования. Представляет собой объект, позволяющий получить последовательный доступ к элементам объекта-агрегата без использования описаний каждого из агрегированных объектов. Например, такие элементы как дерево, связанный список, хеш-таблица и массив могут быть пролистаны (и модифицированы) с помощью объекта Итератор. Перебор элементов выполняется объектом итератора, а не самой коллекцией. Это упрощает интерфейс и реализацию коллекции, а также способствует более логичному разделению обязанностей.

Назначение паттерна:

- Предоставляет способ последовательного доступа ко всем элементам составного объекта, не раскрывая его внутреннего представления.
- Абстракция в стандартных библиотеках C++ и Java, позволяющая разделить классы коллекций и алгоритмов.
- Придает обходу коллекции "объектно-ориентированный статус".
- Полиморфный обход.

Многие языки высокого уровня (C#, C++, F#, PHP) имеют собственную поддержку Итераторов.

Итераторы можно разделить на два вида:

- Внешние – клиент получает экземпляр Итератора и сам запрашивает у него элементы коллекции.
- Внутренние – клиент передает Итератору метод, который вызывается для каждого элемента коллекции.

## 2.2 Структура шаблона Iterator

Структура шаблона Iterator показана на рисунке 2.1.

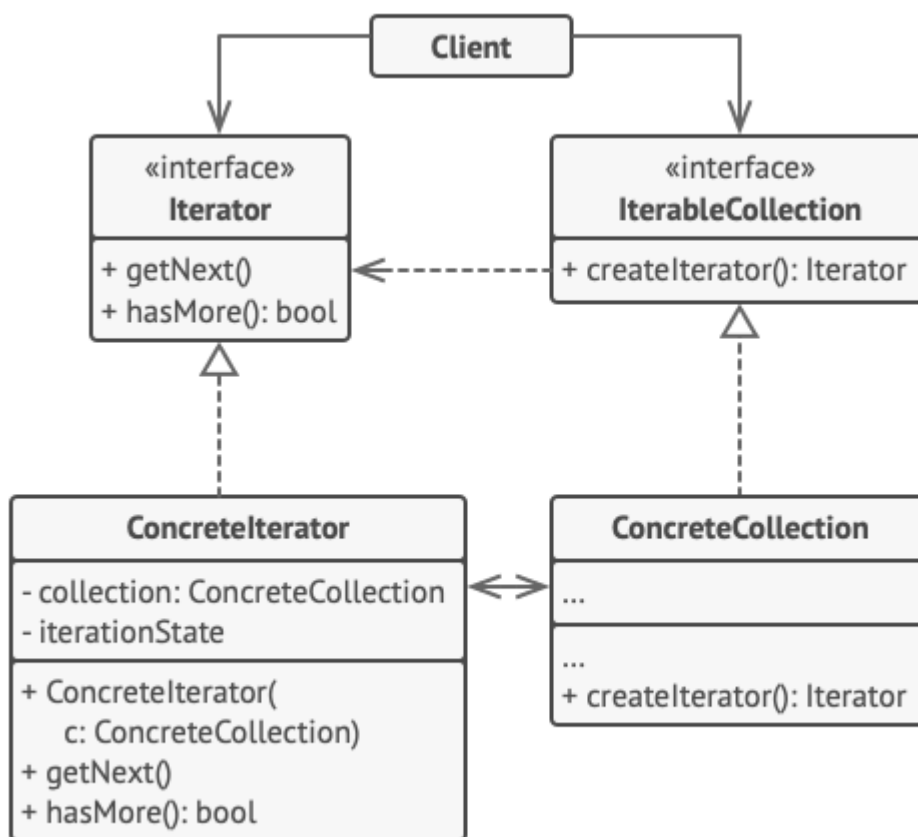


Рисунок 2.1 – Структура шаблона Iterator

**Iterator** – описывает интерфейс для доступа и обхода элементов коллекции.

**ConcreteIterator** – реализует алгоритм обхода какой-то конкретной коллекции. Объект итератора должен сам отслеживать текущую позицию при обходе коллекции, чтобы отдельные итераторы могли обходить одну и ту же коллекцию независимо.

**IterableCollection** описывает интерфейс получения итератора из коллекции.

**ConcreteCollection** - возвращает новый экземпляр определённого конкретного итератора, связав его с текущим объектом коллекции.

**Client** – работает со всеми объектами через интерфейсы коллекции и итератора.

### 3 ОПИСАНИЕ UML ДИАГРАММ

UML (англ. Unified Modeling Language – унифицированный язык моделирования) – язык графического описания для объектного моделирования в области разработки программного обеспечения, моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

UML является языком широкого профиля, это – открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML-моделей возможна генерация кода.

#### 3.1 Диаграмма классов

Диаграмма классов приведена на рисунке 3.1. В UML диаграмма классов является типом диаграммы статической структуры. Она описывает структуру системы, показывая её классы, их атрибуты и операторы, а также взаимосвязи этих классов. Взаимосвязь – это особый тип логических отношений между сущностями, показанных на диаграммах классов и объектов.

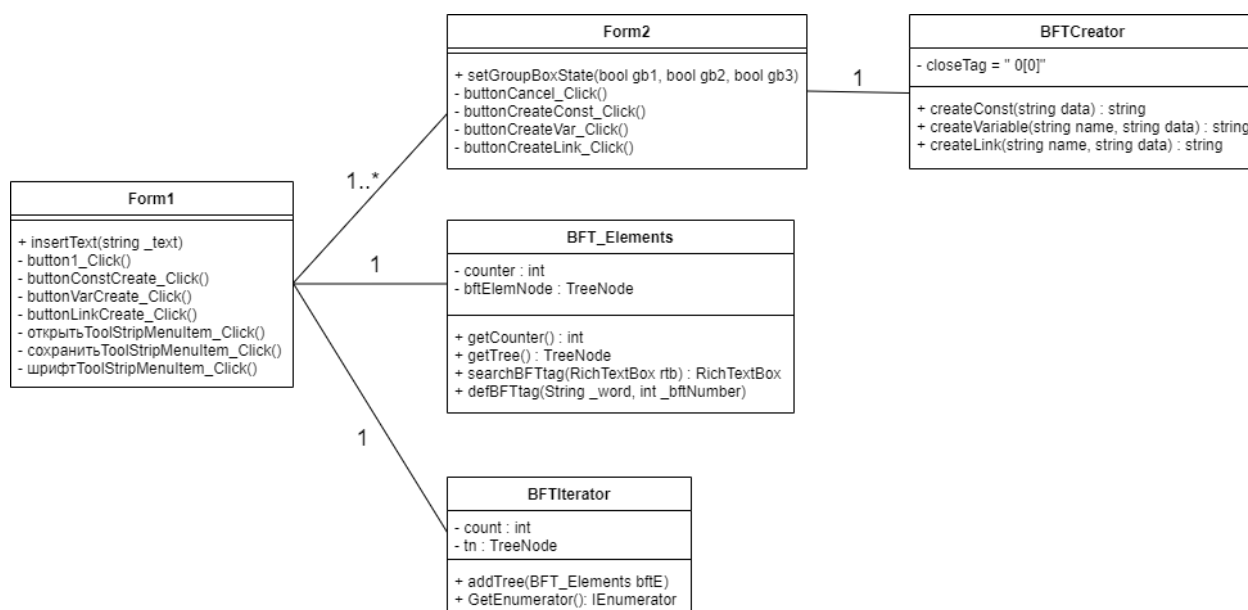


Рисунок 3.1 – Диаграмма классов

### 3.2 Диаграмма последовательности

Диаграмма состояний приведена на рисунке 3.2. Диаграмма последовательности отражает динамические аспекты поведения системы. По существу, эта диаграмма представляет собой блок-схему, которая наглядно показывает, как поток управления переходит от одной деятельности к другой.

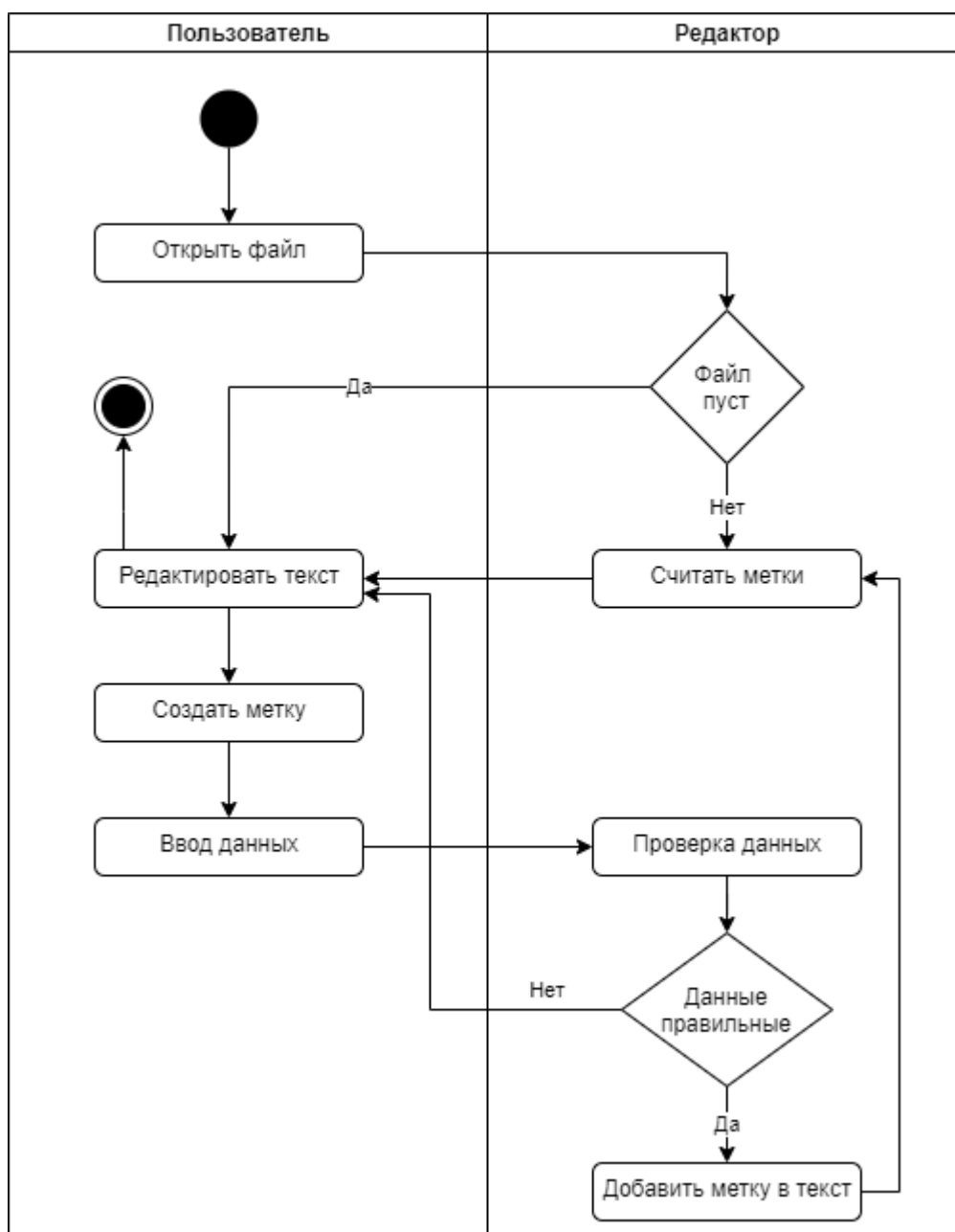


Рисунок 3.2 – Диаграмма последовательности

### 3.3 Диаграмма деятельности

Диаграмма деятельности приведена на рисунке 3.3. Для моделирования процесса выполнения операций в языке UML используются диаграммы деятельности. Каждое состояние на диаграмме деятельности соответствует выполнению некоторой элементарной операции, а переход в следующее состояние срабатывает только при завершении этой, операции в предыдущем состоянии. Графически диаграмма деятельности представляется в форме графа деятельности, вершинами которого являются состояния действия, а дугами – переходы от одного состояния действия к другому.

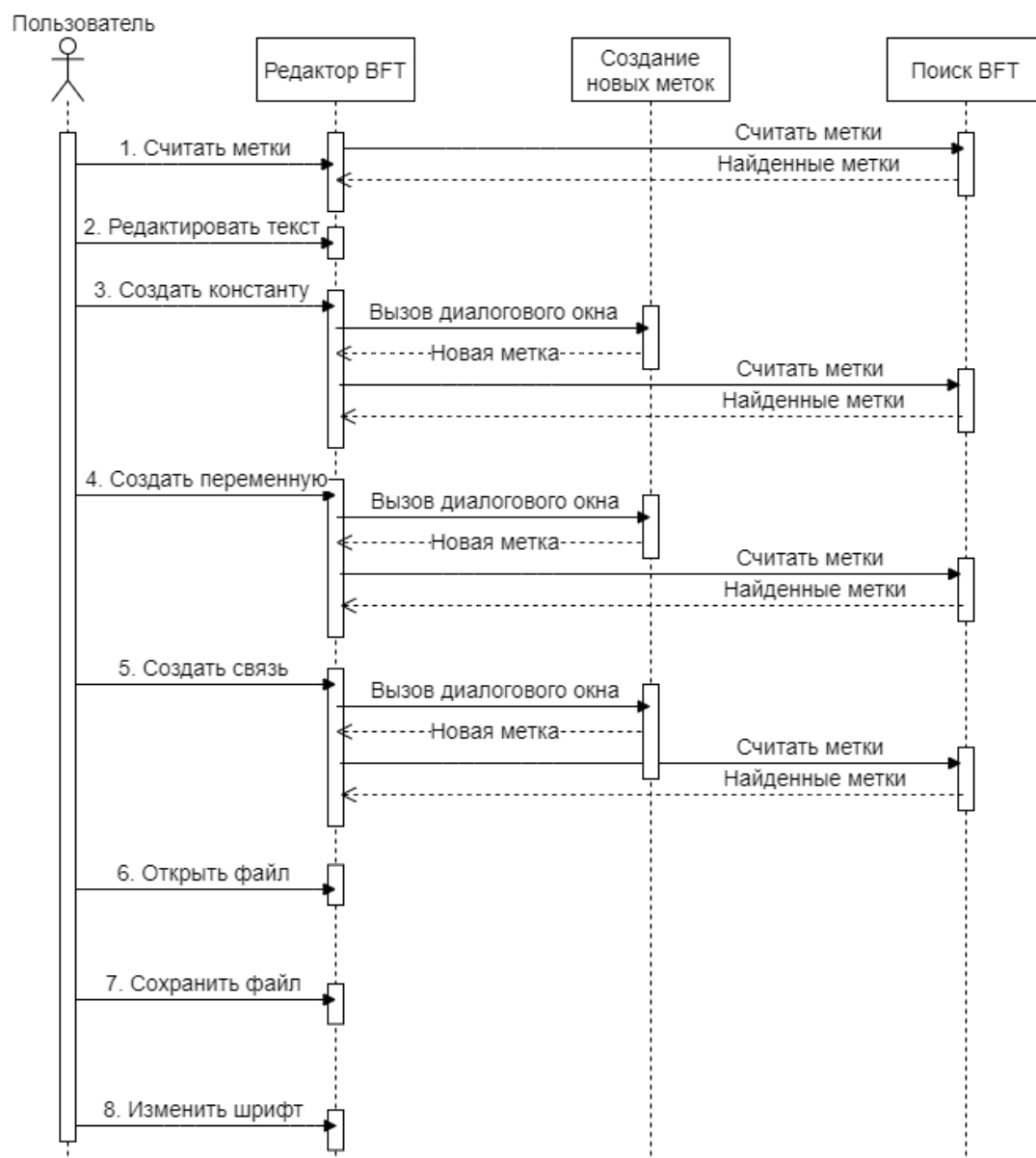


Рисунок 3.3 – Диаграмма деятельности

### 3.4 Диаграмма компонентов

Диаграмма компонентов приведена на рисунке 3.4.

Диаграмма компонентов описывает особенности физического представления системы. Она позволяет определить архитектуру 13 разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код. Во многих средах разработки модуль или компонент соответствует файлу.

Пунктирные стрелки, соединяющие модули, показывают отношения взаимозависимости, аналогичные тем, которые имеют место при компиляции исходных текстов программ. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними.

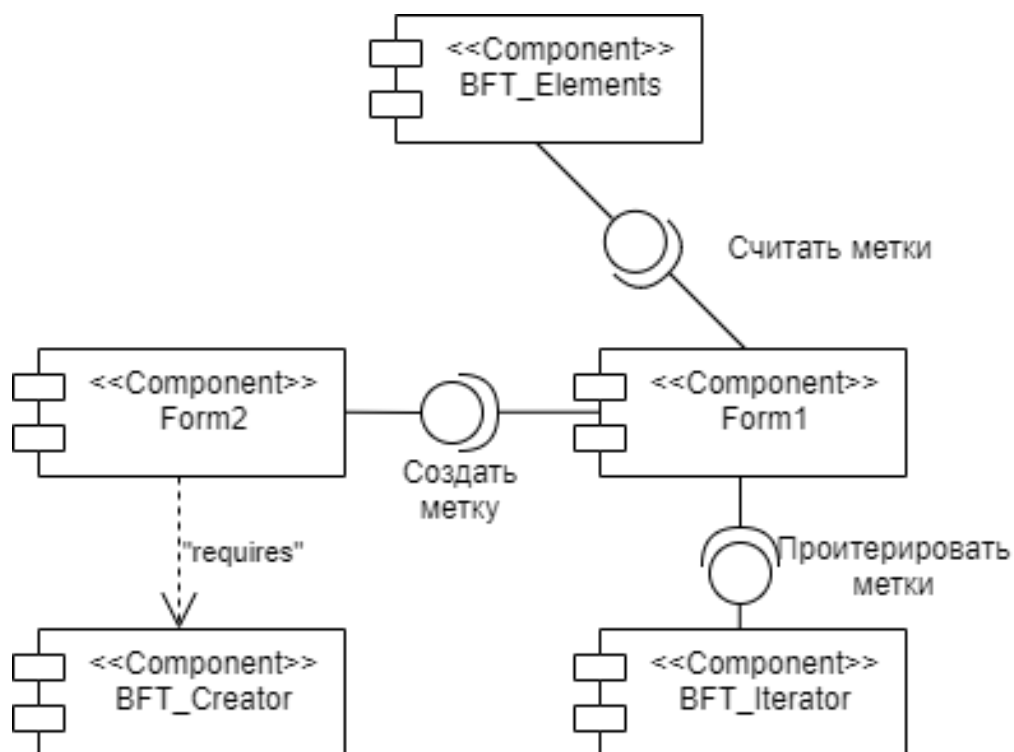


Рисунок 3.4 – Диаграмма компонентов

### 3.5 Диаграмма вариантов использования

Диаграмма вариантов использования приведена на рисунке 3.5.

Диаграмма вариантов использования в UML – диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Диаграмма вариантов использования рассматривается как главное средство для первичного моделирования динамики системы, используется для выяснения требований к разрабатываемой системе, фиксации этих требований в форме, которая позволит проводить дальнейшую разработку.



Рисунок 3.5 – Диаграмма использования

## 4 РАЗРАБОТКА ПРОГРАММЫ НА ОСНОВЕ ШАБЛОНА ITERATOR

### 4.1 Описание классов

Для реализации BFT редактора были созданы классы: BFT\_Elements, BFT\_Creator и BFT\_Iterator.

Класс BFT\_Elements содержит в себе свойство counter, хранящее кол-во BFT меток в тексте, и дерево tags, содержащее в себе сами BFT метки. Метод searchBFTtags() совершает обход текста, размечая цветом метки в тексте, и отправляя их в метод isBFTtag(), определяющий тип меток.

Класс BFT\_Creator содержит в себе 3 метода для составления новых меток по введенным данным: константы, переменной и связи.

Класс BFT\_Iterator наследуется от IEnumerable, что позволяет совершать обход всех BFT меток и возвращать содержащийся в них текст.

Form1 содержит richTextBox1 для ввода и редактирования текста, treeView1 для вывода на экран считанных меток. С помощью кнопки «Файл» пользователь может открыть файл формата “.rtf” для считывания из него данных или сохранить уже введенные данные в том же формате. При нажатии на кнопку “Шрифт” будет вызвано диалоговое окно, в котором можно будет выбрать шрифт и его размер. При нажатии на кнопку “Считать метки” текст из richTextBox1 будет передан в класс BFT\_Elements, где будут найдены BFT метки и выведены в древо treeView1. В форме определен метод insertText(), который нужен для взаимодействия между первой и второй формами.

При нажатии на кнопки создания будет открыта Form2, в которой можно ввести параметры выбранной метки. Если данные были введены правильно, то в позиции пользовательского курсора в редакторе будет создана новая метка, используя класс BFT\_Creator.



## 5 ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

На рисунке 5.1 представлено открытое приложение:

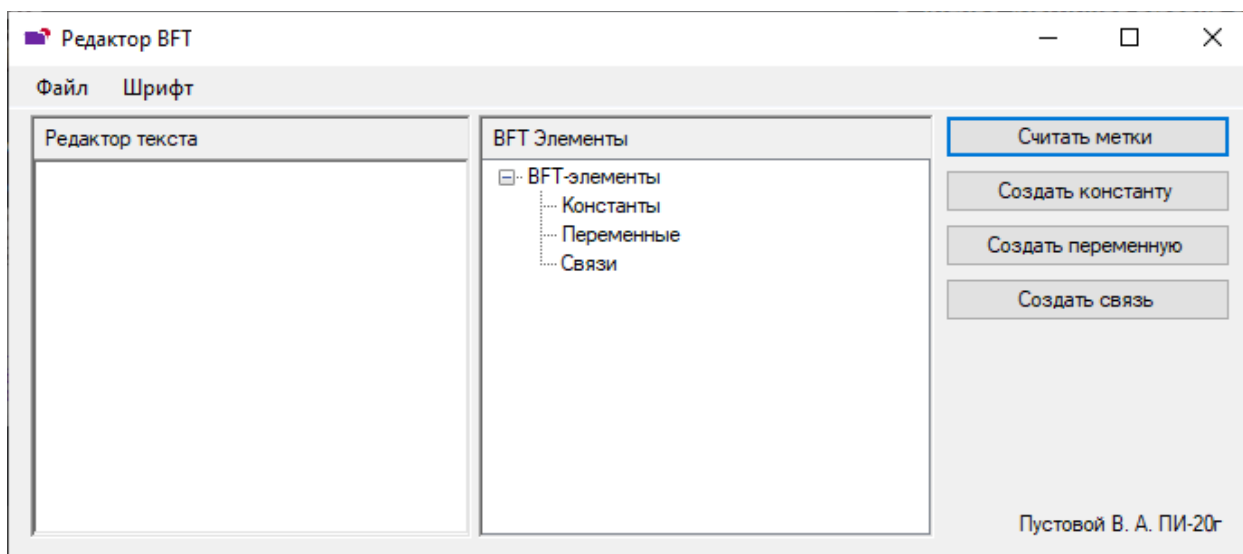


Рисунок 5.1 – Главная форма программы

С помощью ползунка между элементами, можно изменять их размер.

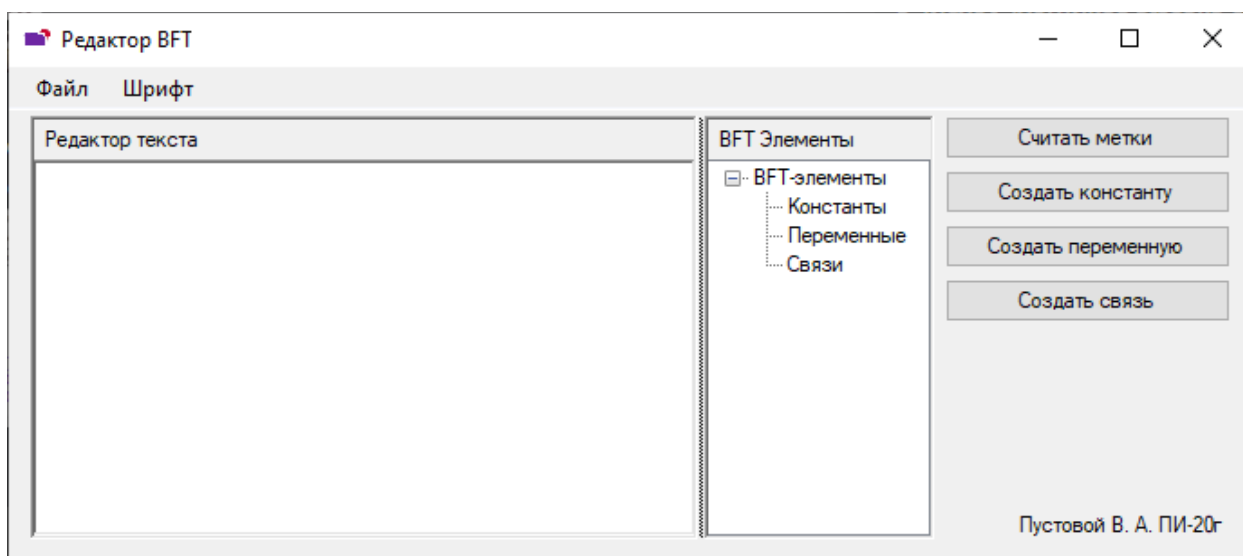


Рисунок 5.2 – Главная форма программы

Нажав на “Файл” => ”Открыть” будет открыто диалоговое окно с рисунка 5.3, в котором нужно выбрать файл для открытия.

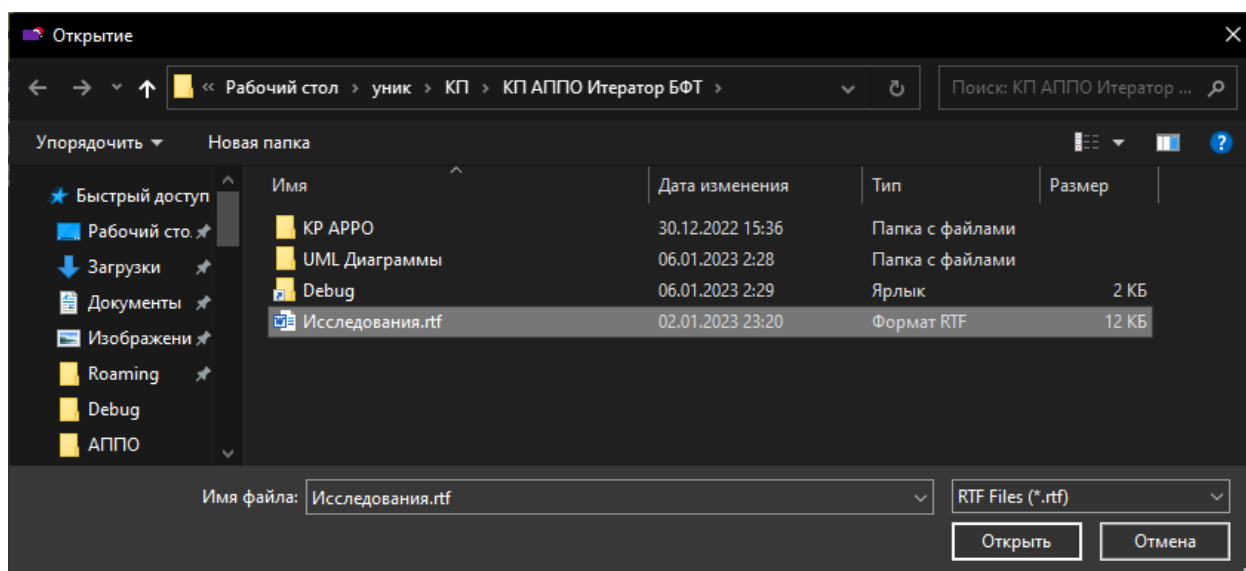


Рисунок 5.3 – Выбор файла для открытия

При считывании файла, найденные метки будут выведены на экран:

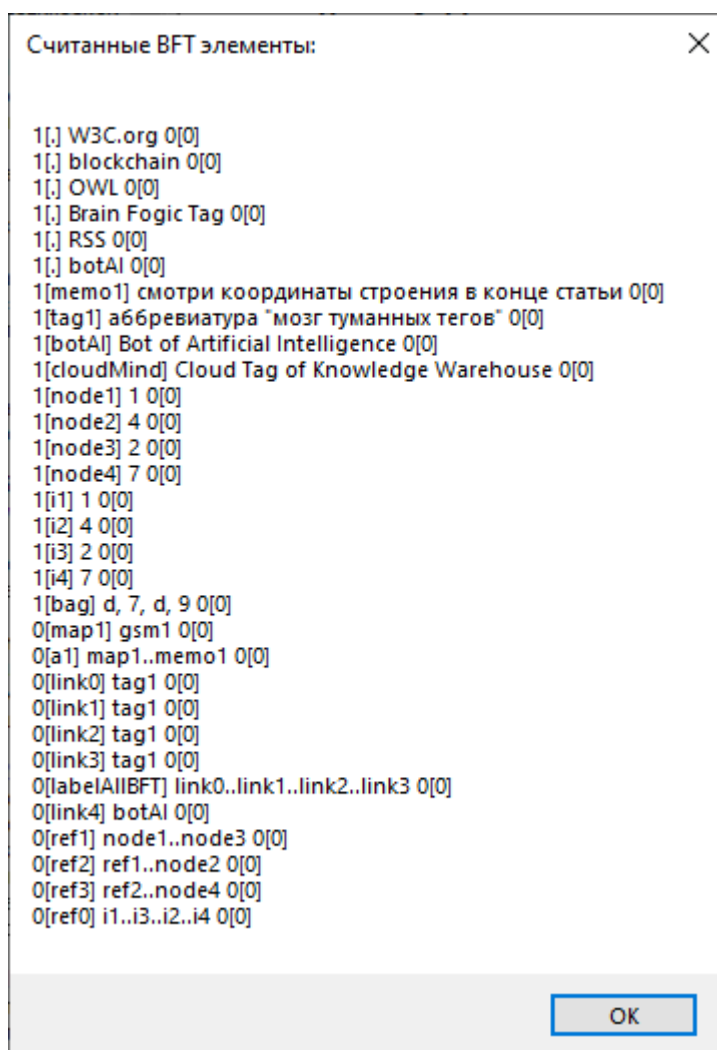


Рисунок 5.4 – Считанные из файла метки

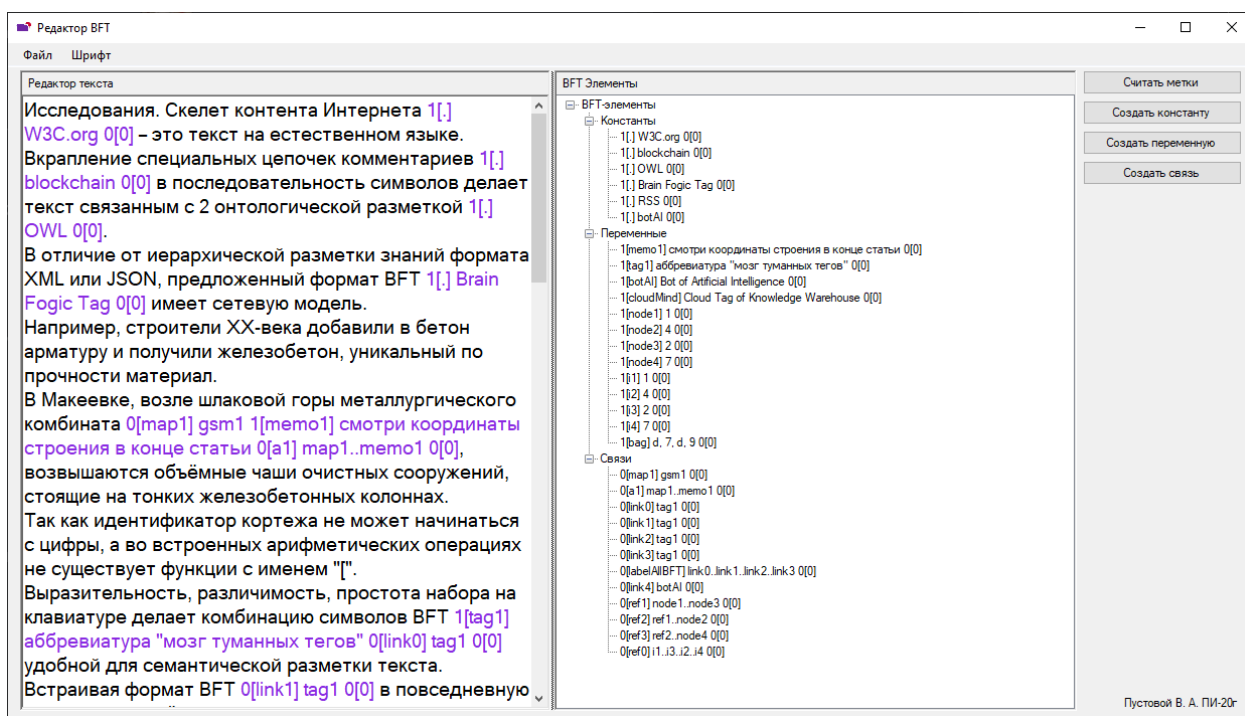


Рисунок 5.5 – Редактор текста после считывания

Для тестирования программы, введём набор меток и текст после них:

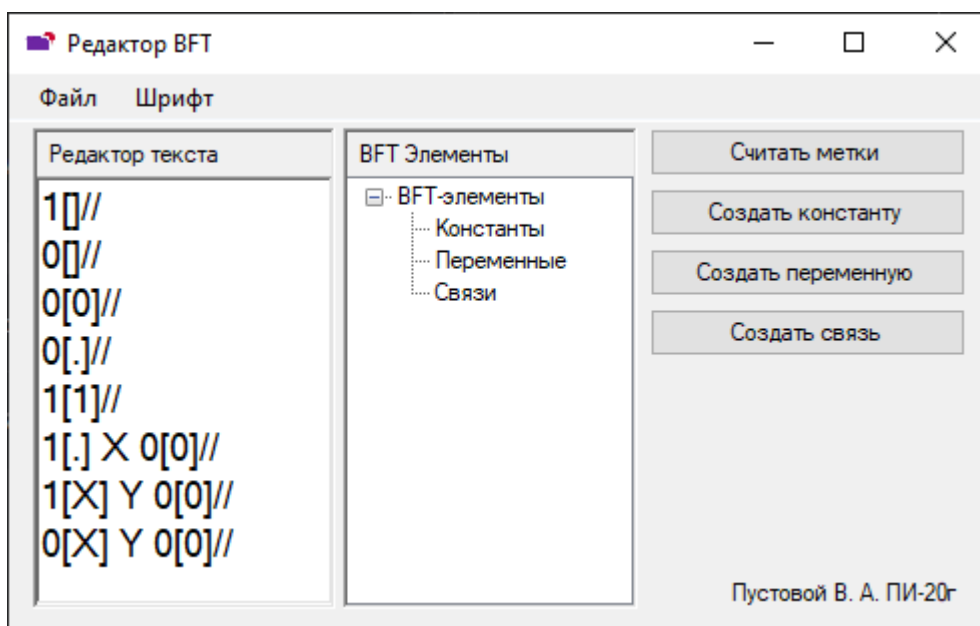


Рисунок 5.6 – Набор BFT меток

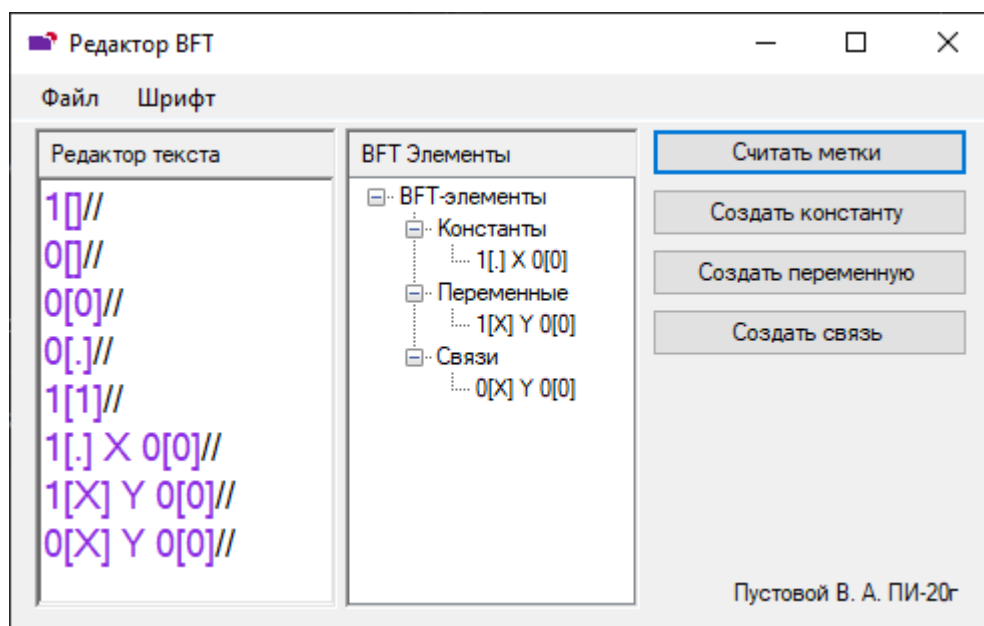


Рисунок 5.7 – Подсветка найденных BFT меток

Создадим несколько меток:

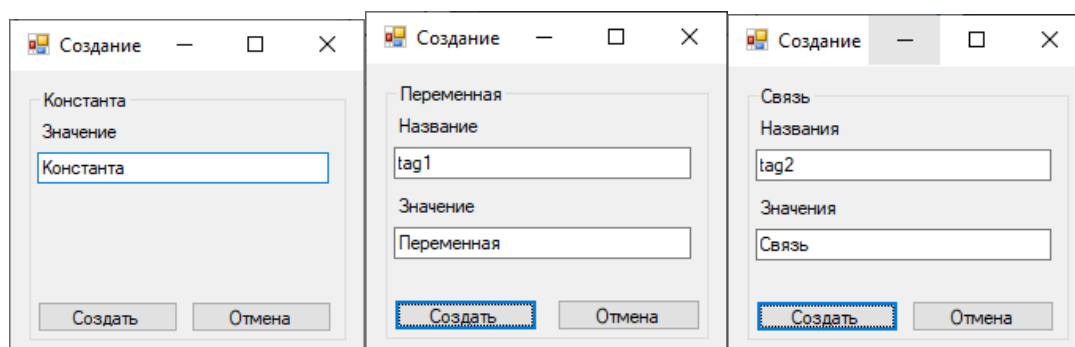


Рисунок 5.8 – Окна создания новой метки

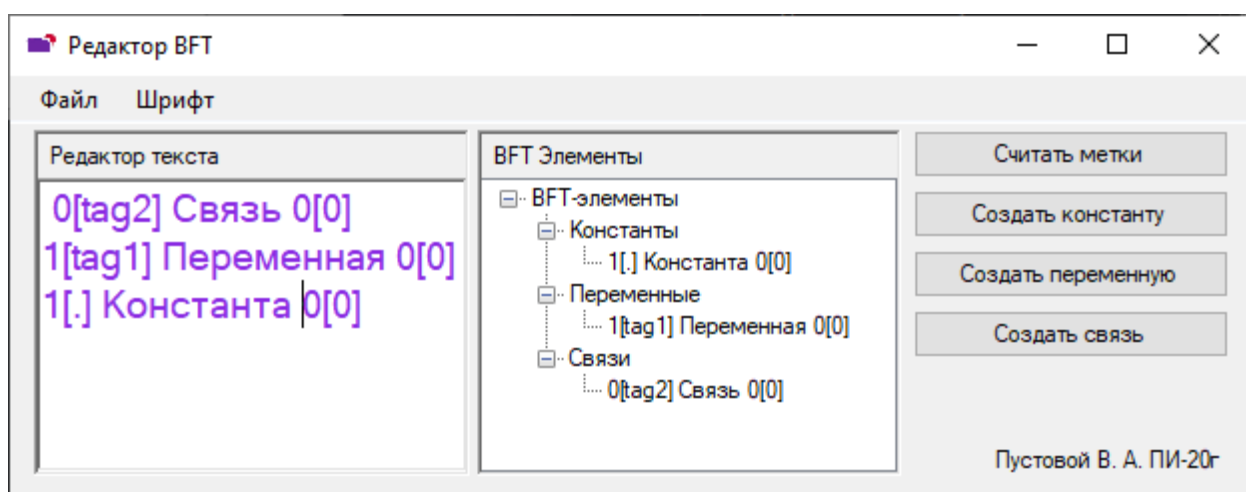


Рисунок 5.9 – Созданные метки

Попробуем создать метку в положении курсора:

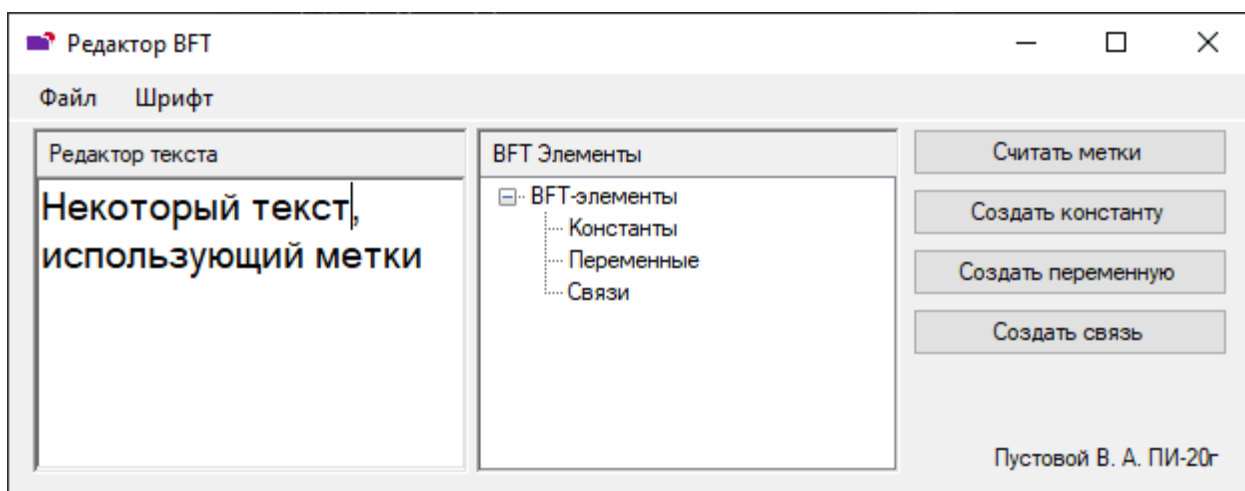


Рисунок 5.10 – Положение курсора в richTextBox

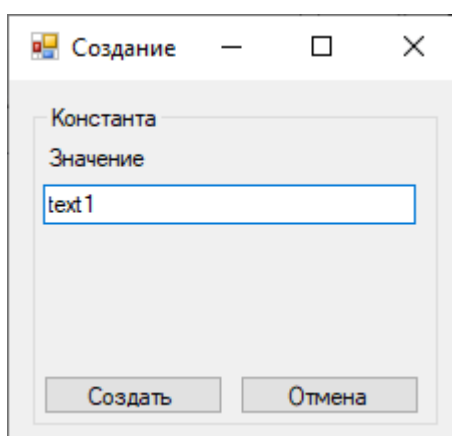


Рисунок 5.11 – Окно создания новой константы

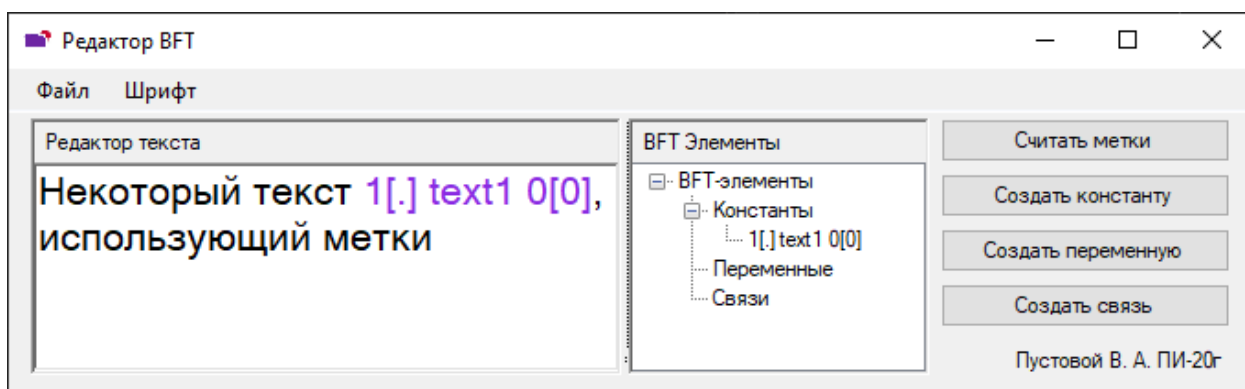


Рисунок 5.12 – Полученная метка

Если оставить поля пустыми или ввести в них знаки “[” “]”, то:

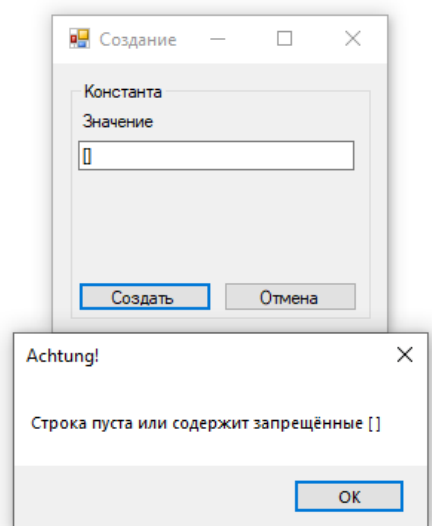


Рисунок 5.13 – Ошибка при создании новой константы

Попробуем сохранить файл:

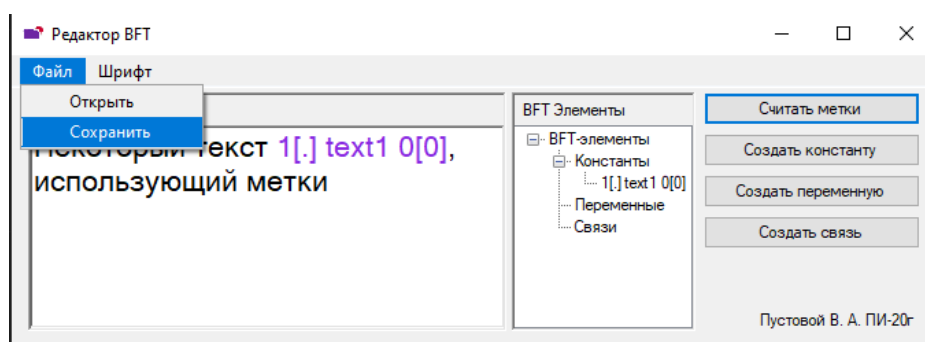


Рисунок 5.14 – Контекстное меню Файл

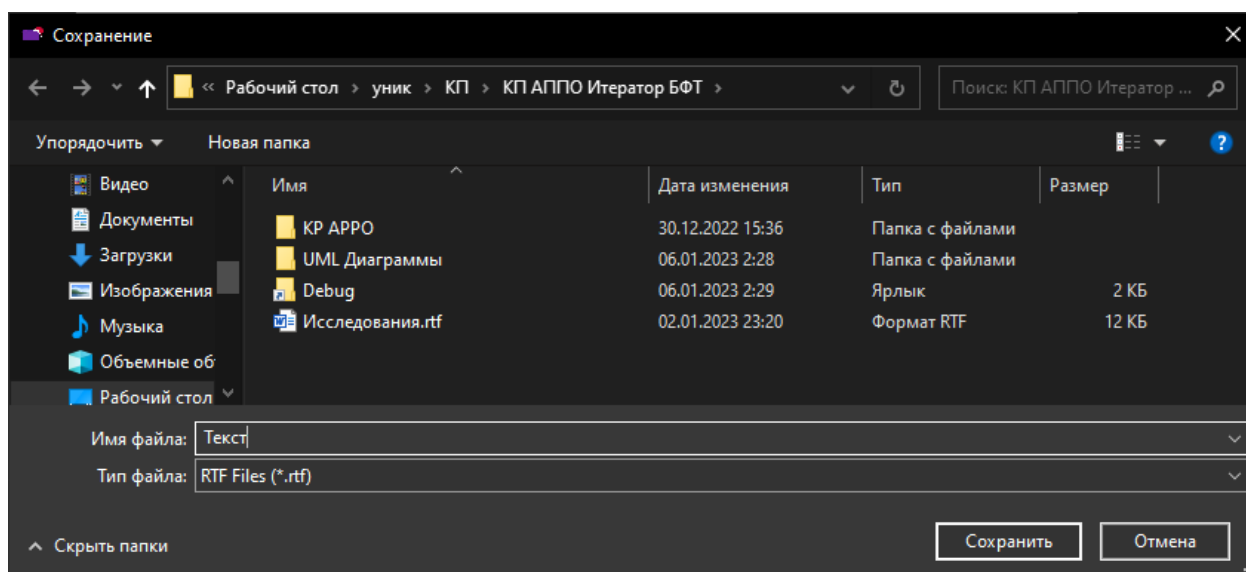


Рисунок 5.15 – Выбор файла для сохранения

## ЗАКЛЮЧЕНИЕ

В ходе курсового проекта была разработана программа для работы с нотацией BrainFogicTag. Для ее проектирования был использован графический язык моделирования UML, который позволил описать и сформулировать четкую структуру создаваемого программного продукта.

В ходе разработки был применен шаблон проектирования Iterator. Данный паттерн – один из двадцати трех хорошо известных шаблонов проектирования GoF, которые описывают, как решать повторяющиеся проблемы проектирования для разработки гибкого и многократно используемого объектно-ориентированного программного обеспечения, то есть, объекты, которые легче реализовать, изменить, протестировать и повторно использовать.

Для реализации был выбран объектно-ориентированный подход на языке программирования C# с использованием фреймворка .Net. Благодаря этому стала возможна простая необходимых классов и создание удобного и понятного интерфейса с использованием инструментов Windows Form.

Для построения диаграмм было использовано приложение Draw.io.

Работа была успешно протестирована на разнообразный пользовательский ввод и были добавлены обработчики всевозможных исключений. В результате была получена стабильная версия программы, корректно обрабатывающая любой ввод и отображающая верные данные.

Также был реализован ввод данных с клавиатуры и файла.

## ПЕРЕЧЕНЬ ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Вирт Н. Алгоритмы и структуры данных. — М.: Мир, 1989. — С. 272—286.
2. Д. Кнут. Искусство Программирования. Сортировка и Поиск. — С. 460.
3. Гамма, Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. — СПб.: Питер, 2011. — 400 с.
4. Емельянова Н.З. Основы построения автоматизированных информационных систем: Учебное пособие для среднего проф. образования / Н. З. Емельянова. — М.: ФОРУМ ГРУПП ИНФРА-М, 2015. — 416 с.
5. Босуэлл, Дастин. Читаемый код, или Программирование как искусство / Дастин Босуэлл, Тревор Фаучер; [пер. с англ. И. Афанасьева]. — М.: ТИД «ДС», 2012. — 432 с.
6. Канер С. Тестирование программного обеспечения: фундаментальные концепции менеджмента бизнес-приложений / С. Канер, Д.Фолк, Енг Кек Нгуен; [пер. с англ.] — К.: ДияСофт, 2011. — 544 с.
6. Нотация связанных данных BrainFogicTag 1[.] BFT 0[0] [Электронный ресурс]. — Режим доступа:  
[https://www.academia.edu/39589256/Нотация\\_связанных\\_данных\\_BrainFogicTag\\_1\\_BFT\\_0\\_0\\_дополняющих\\_контекст\\_естественного\\_текста](https://www.academia.edu/39589256/Нотация_связанных_данных_BrainFogicTag_1_BFT_0_0_дополняющих_контекст_естественного_текста)



## Приложение А

### ТЕХНИЧЕСКОЕ ЗАДАНИЕ

#### А.1 Общие сведения

Тема курсового проекта: «Шаблон проектирования Iterator для BFT».

Курсовой проект разрабатывается студентом 3-го курса Донецкого национального технического университета, факультета ИСП, группы ПИ-20г, Пустовым В.А.

Основанием для разработки КП является задание, выданное кафедрой прикладной математики и искусственного интеллекта. Плановый срок начала работы по выполнению задания курсового проекта: 02.09.2022, срок окончания: 28.12.2022. Курсовой проект должен выполняться согласно графику, приведенному в таблице А.1

Таблица А.1 – Этапы, результаты и сроки разработки программного продукта.

№	Этап работы	Срок выполнения (№ недели)
1	Получение задания на КП	1
2	Выявление требований к разрабатываемому программному продукту	2
3	Изучение паттерна проектирования	3
4	Разработка UML диаграмм	4-6
5	Разработка программного продукта	7-11
5	Тестирования и отладка	12-13
6	Оформление пояснительной записки	14-17
7	Защита курсового проекта	18

#### А.2 Назначения и цели создания проекта

Целью создания проекта является приобретение практических навыков в использовании паттерна проектирования Iterator на примере создания программного продукта на языке С#. Проект предназначен для демонстрации работы паттерна Iterator и тестового примера BFT.

### А.3 Требования к курсовому проекту

Необходимо спроектировать программу при помощи графического языка UML, и разработать программный продукт для работы с нотацией BrainFogicTag, использующий паттерн Iterator.

### А.4 Требования к задачам и функциям программного продукта

В процессе работы необходимо обеспечить выполнение следующих функций:

- ввод текста с файла или клавиатуры;
- проверка текста на наличие BFT меток;
- создание правильных BFT меток;
- работа программы на различных данных для демонстрации работы программного продукта;
- вывод полученных данных на экран.

### А.5 Требования к техническому обеспечению

- К техническому обеспечению предъявляются следующие требования:
- процессор – 32-битный x86-совместимый (уровня Pentium и выше);
- объем оперативной памяти – не менее 1 Гб;
- свободное дисковое пространство – около 50 Мб. Не менее 5 Мб свободного дискового пространства для временных файлов;
- графический адаптер – VGA-совместимый;
- монитор – VGA-совместимый;
- клавиатура.

## А.6 Требования к программному обеспечению

В курсовом проекте должны быть соблюдены следующие требования к программному обеспечению:

- программный продукт должен проектироваться как текстовый редактор по работе с нотацией BFT;
- при разработке проекта должен использоваться любой объектно-ориентированный язык программирования;
- программный продукт должен обладать интуитивно понятным и легким в использовании интерфейсом.

## А.7 Требования к организационному обеспечению

В программную документацию должны входить:

- пояснительная записка;
- приложения:
  - техническое задание;
  - листинг программных модулей.

## Приложение Б

### ЛИСТИНГ ПРОГРАММНЫХ МОДУЛЕЙ

#### Б.1 Файл Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace KP_APP0
{
    static class Program
    {
        /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

#### Б.2 Файл App.config

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>
</configuration>
```

#### Б.3 Файл BFT\_Creator

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace KP_APP0
{
    class BFT_Creator
    {
        const string closeTag = " 0[0]";

        public string createConst(string data)
        {
            return (" 1[.] " + data + closeTag);
        }

        public string createVariable(string name, string data)
        {
            return (" 1[" + name + "]" + data + closeTag);
        }
    }
}
```

```

        public string createLink(string name, string data)
        {
            return (" 0[" + name + "]" + data + closeTag);
        }
    }
}

```

#### Б.4 Файл BFT\_Iterator

```

using System;
using System.Collections;
using System.Linq;
using System.Collections.Generic;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace KP_APP0
{
    class BFT_Iterator : IEnumerable
    {
        private int count;
        private TreeNode tn = new TreeNode();

        public void addTree(BFT_Elements bftE)
        {
            count = bftE.getCounter();
            tn = bftE.getTree();
        }
        public IEnumerator GetEnumerator()
        {
            foreach(TreeNode tag in tn.Nodes[0].Nodes)
            {
                yield return tag.Text;
            }
            foreach (TreeNode tag in tn.Nodes[1].Nodes)
            {
                yield return tag.Text;
            }
            foreach (TreeNode tag in tn.Nodes[2].Nodes)
            {
                yield return tag.Text;
            }
        }
    }
}

```

#### Б.5 Файл BFT\_Elements

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Drawing;
using System.Windows.Forms;

namespace KP_APP0
{
    class BFT_Elements
    {
        // кол-во BFT элементов
    }
}

```

```

private int counter;
// дерево, хранящее метки
private TreeNode tags;
public BFT_Elements()
{
    counter = 0;
    // заполнение дерева при создании объекта
    tags = new TreeNode("BFT-элементы");
    // 1[.] value1 0[0]
    tags.Nodes.Add(new TreeNode("Константы"));
    // 1[type] value2 0[0]
    tags.Nodes.Add(new TreeNode("Переменные"));
    // 0[type] value1..value2 0[0]
    tags.Nodes.Add(new TreeNode("Связи"));
}
public int getCounter()
{
    return counter;
}
public TreeNode getTree()
{
    return tags;
}
// поиск тэгов по тексту
public RichTextBox searchBFTtag(RichTextBox rtb)
{
    rtb.SelectAll();
    rtb.SelectionColor = Color.Black;

    // начало BFT-элемента
    bool flag = false;
    // начало новой метки
    int buffer = 0;
    // Cursor position место курсора перед обыском
    int cp = rtb.SelectionStart;
    // отсчёт начинаем с первого вхождения скобки
    if (rtb.Text.IndexOf('[') > 0)
        for (int i = rtb.Text.IndexOf('[') - 1; i < rtb.Text.Length; i++)
        {
            if (!flag)
            {
                // поиск начала метки
                if ((rtb.Text[i] == '1' || rtb.Text[i] == '0') && rtb.Text[i + 1]
== '[' )
                {
                    if (rtb.Text[i + 2] == ']')
                    {
                        // 1[] || 0[]
                        rtb.Select(i, 3);
                        rtb.SelectionColor = Color.BlueViolet;
                    }
                    else if (rtb.Text[i] == '0' && (rtb.Text[i + 2] == '.' ||
rtb.Text[i + 2] == '0') && rtb.Text[i + 3] == ']')
                    {
                        // 0[.] || 0[0]
                        rtb.Select(i, 4);
                        rtb.SelectionColor = Color.BlueViolet;
                    }
                    else if (rtb.Text[i] == '1' && rtb.Text[i + 2] == '1' &&
rtb.Text[i + 3] == ']')
                    {
                        // 1[1]
                        rtb.Select(i, 4);
                        rtb.SelectionColor = Color.BlueViolet;
                    }
                }
            }
            else
            {

```

```

        flag = true;
        buffer = i;
        i += 4;
    }
}
else
{
    // поиск конца метки
    if (rtb.Text[i] == '0' && rtb.Text[i + 1] == '[' && rtb.Text[i +
2] == '0' && rtb.Text[i + 3] == ']')
    {
        counter++;
        this.isBFTtag(rtb.Text.Substring(buffer, i + 4 - buffer),
counter);

        // выделяем тэг подсветкой
        rtb.Select(buffer, i + 4 - buffer);
        rtb.SelectionColor = Color.BlueViolet;
        flag = false;
        // после конечной метки всегда пробел
        i += 4;
    }
}
}
rtb.Select(cp, 0);
rtb.SelectionStart = cp;
return rtb;
}
// определение типа тэга
public void isBFTtag(String _word, int _bftNumber)
{
    // строка очищается слева направо до тех пор, пока не останется только тэг
закрытия
    while (!_word.Equals("0[0]"))
    {
        if (_word.StartsWith("1[.]"))
        {
            // убираем открывающий тэг, иначе IndexOf найдет не ту [
            _word = _word.Substring(5);
            tags.Nodes[0].Nodes.Add(_bftNumber.ToString(), "1[." +
_word.Substring(0, _word.IndexOf('[') - 2) + " 0[0]");
        }
        else if (_word.StartsWith("1[")
        {
            _word = _word.Substring(2);
            tags.Nodes[1].Nodes.Add(_bftNumber.ToString(), "1[" +
_word.Substring(0, _word.IndexOf('[') - 2) + " 0[0]");
        }
        else // if (_word.StartsWith("0[")
        {
            _word = _word.Substring(2);
            tags.Nodes[2].Nodes.Add(_bftNumber.ToString(), "0[" +
_word.Substring(0, _word.IndexOf('[') - 2) + " 0[0]");
        }
        // убираем пробел перед началом закрывающего или следующего открывающего
тэга
        _word = _word.Substring(_word.IndexOf('[') - 1);
    }
}
}
}
}

```

## Б.6 Файл Form1.cs

```

using System;
using System.IO;
using System.Drawing;
using System.Windows.Forms;

namespace KP_APP0
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            BFT_Elements bftE = new BFT_Elements();
            bftE.searchBFTtag(richTextBox1);
            treeView1.Nodes.Clear();
            treeView1.Nodes.Add(bftE.getTree());
            treeView1.ExpandAll();
        }

        // вставка нового бфт элемента и чтение всего текста заново
        public void insertText(string _text)
        {
            try
            {
                BFT_Elements bftE = new BFT_Elements();
                // запоминаем положение курсора
                int i = richTextBox1.SelectionStart;
                richTextBox1.Text = richTextBox1.Text.Insert(i, _text);
                // перемещаем курсор в конец тэга
                richTextBox1.SelectionStart = i + _text.Length;
                richTextBox1 = bftE.searchBFTtag(richTextBox1);
                treeView1.Nodes.Clear();
                treeView1.Nodes.Add(bftE.getTree());
                treeView1.ExpandAll();
            }
            catch (Exception exc)
            {
                MessageBox.Show(exc.ToString(), "Achtung!");
            }
        }

        // кнопка чтения тэгов
        private void button1_Click(object sender, EventArgs e)
        {
            try
            {
                BFT_Elements bftE = new BFT_Elements();
                richTextBox1 = bftE.searchBFTtag(richTextBox1);
                treeView1.Nodes.Clear();
                // добавление древа тэгов на вывод
                treeView1.Nodes.Add(bftE.getTree());
                treeView1.ExpandAll();
            }
            catch (Exception exc)
            {
                MessageBox.Show(exc.ToString(), "Achtung!");
            }
        }
    }
}

```



```

    }
    private void buttonConstCreate_Click(object sender, EventArgs e)
    {
        try
        {
            Form2 secondForm = new Form2();
            // открывает возможность передачи данных и изменения их, при вызове из
            другой формы
            secondForm.Owner = this;
            secondForm.Show();
            secondForm.setGroupBoxState(true, false, false);
        }
        catch (Exception exc)
        {
            MessageBox.Show(exc.ToString(), "Achtung!");
        }
    }

    private void buttonVarCreate_Click(object sender, EventArgs e)
    {
        try
        {
            Form2 secondForm = new Form2();
            secondForm.Owner = this;
            secondForm.Show();
            secondForm.setGroupBoxState(false, true, false);
        }
        catch (Exception exc)
        {
            MessageBox.Show(exc.ToString(), "Achtung!");
        }
    }

    private void buttonLinkCreate_Click(object sender, EventArgs e)
    {
        try
        {
            Form2 secondForm = new Form2();
            secondForm.Owner = this;
            secondForm.Show();
            secondForm.setGroupBoxState(false, false, true);
        }
        catch (Exception exc)
        {
            MessageBox.Show(exc.ToString(), "Achtung!");
        }
    }

    private void открытьToolStripMenuItem_Click(object sender, EventArgs e)
    {
        try
        {
            BFT_Elements bftE = new BFT_Elements();
            OpenFileDialog openFile1 = new OpenFileDialog();

            // Ищем файлы формата .rtf
            openFile1.DefaultExt = "*.rtf";
            openFile1.Filter = "RTF Files|*.rtf|All Files|*.*";

            // Был ли выбран файл
            if (openFile1.ShowDialog() == System.Windows.Forms.DialogResult.OK &&
                openFile1.FileName.Length > 0)
            {
                // Загружаем файл в rtb
            }
        }
    }

```

```

        richTextBox1.LoadFile(openFile1.FileName);
        richTextBox1 = bftE.searchBFTtag(richTextBox1);
        treeView1.Nodes.Clear();
        treeView1.Nodes.Add(bftE.getTree());
        treeView1.ExpandAll();
        // создаем итератор и заполняем его
        BFT_Iterator bftI = new BFT_Iterator();
        bftI.addTree(bftE);
        string message = null;
        // собираем из него все BFT элементы
        foreach (string tag in bftI)
        {
            message += tag + "\r\n";
        }
        MessageBox.Show(message, "Считанные BFT элементы:");
    }
}
catch (Exception exc)
{
    MessageBox.Show(exc.ToString(), "Achtung!");
}
}

private void сохранитьToolStripMenuItem_Click(object sender, EventArgs e)
{
    try
    {
        // Сохраняем файл в формате .rtf
        SaveFileDialog saveFile1 = new SaveFileDialog();
        saveFile1.DefaultExt = "*.rtf";
        saveFile1.Filter = "RTF Files|*.rtf";
        // если ввели название файла
        if (saveFile1.ShowDialog() == DialogResult.OK &&
saveFile1.FileName.Length > 0)
        {
            richTextBox1.SaveFile(saveFile1.FileName);
        }
    }
    catch (Exception exc)
    {
        MessageBox.Show(exc.ToString(), "Achtung!");
    }
}

private void шрифтToolStripMenuItem_Click(object sender, EventArgs e)
{
    try
    {
        // Вызываем диалог изменения шрифта
        BFT_Elements bftE = new BFT_Elements();
        FontDialog fd = new FontDialog();
        fd.Font = richTextBox1.Font;
        if (fd.ShowDialog() != DialogResult.Cancel)
        {
            richTextBox1.Font = fd.Font;
            richTextBox1 = bftE.searchBFTtag(richTextBox1);
            treeView1.Nodes.Clear();
            treeView1.Nodes.Add(bftE.getTree());
            treeView1.ExpandAll();
        }
    }
    catch (Exception exc)
    {

```

```

        MessageBox.Show(exc.ToString(), "Achtung!");
    }
}
}

```

## Б.7 Файл Form1.Designer.cs

```

namespace KP_APPO
{
    partial class Form1
    {
        /// <summary>
        /// Обязательная переменная конструктора.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Освободить все используемые ресурсы.
        /// </summary>
        /// <param name="disposing">истинно, если управляемый ресурс должен быть удален; иначе
ложно.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Код, автоматически созданный конструктором форм Windows

        /// <summary>
        /// Требуемый метод для поддержки конструктора — не изменяйте
        /// содержимое этого метода с помощью редактора кода.
        /// </summary>
        private void InitializeComponent()
        {
            System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(Form1));
            this.button1 = new System.Windows.Forms.Button();
            this.label1 = new System.Windows.Forms.Label();

```

```

this.treeView1 = new System.Windows.Forms.TreeView();
this.buttonConstCreate = new System.Windows.Forms.Button();
this.buttonVarCreate = new System.Windows.Forms.Button();
this.buttonLinkCreate = new System.Windows.Forms.Button();
this.splitContainer1 = new System.Windows.Forms.SplitContainer();
this.richTextBox1 = new System.Windows.Forms.RichTextBox();
this.label2 = new System.Windows.Forms.Label();
this.menuStrip1 = new System.Windows.Forms.MenuStrip();
this.файлToolStripMenuItem = new System.Windows.Forms.ToolStripItem();
this.открытьToolStripMenuItem = new System.Windows.Forms.ToolStripItem();
this.сохранитьToolStripMenuItem = new System.Windows.Forms.ToolStripItem();
this.шрифтToolStripMenuItem = new System.Windows.Forms.ToolStripItem();
this.label3 = new System.Windows.Forms.Label();
((System.ComponentModel.ISupportInitialize)(this.splitContainer1)).BeginInit();
this.splitContainer1.Panel1.SuspendLayout();
this.splitContainer1.Panel2.SuspendLayout();
this.splitContainer1.SuspendLayout();
this.menuStrip1.SuspendLayout();
this.SuspendLayout();
//
// button1
//
this.button1.Anchor
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top
System.Windows.Forms.AnchorStyles.Right)));
this.button1.Location = new System.Drawing.Point(874, 27);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(158, 24);
this.button1.TabIndex = 0;
this.button1.Text = "Считать метки";
this.button1.UseVisualStyleBackColor = true;
this.button1.Click += new System.EventHandler(this.button1_Click);
//
// label1
//
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(3, 4);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(92, 13);
this.label1.TabIndex = 1;
this.label1.Text = "Редактор текста";

```

```

//
// treeView1
//
this.treeView1.Anchor
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top
System.Windows.Forms.AnchorStyles.Bottom)
| System.Windows.Forms.AnchorStyles.Left)
| System.Windows.Forms.AnchorStyles.Right)));
this.treeView1.Location = new System.Drawing.Point(-1, 22);
this.treeView1.Name = "treeView1";
this.treeView1.Size = new System.Drawing.Size(430, 623);
this.treeView1.TabIndex = 3;
//
// buttonConstCreate
//
this.buttonConstCreate.Anchor
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top
System.Windows.Forms.AnchorStyles.Right)));
this.buttonConstCreate.Location = new System.Drawing.Point(874, 57);
this.buttonConstCreate.Name = "buttonConstCreate";
this.buttonConstCreate.Size = new System.Drawing.Size(158, 24);
this.buttonConstCreate.TabIndex = 4;
this.buttonConstCreate.Text = "Создать константу";
this.buttonConstCreate.UseVisualStyleBackColor = true;
this.buttonConstCreate.Click += new System.EventHandler(this.buttonConstCreate_Click);
//
// buttonVarCreate
//
this.buttonVarCreate.Anchor
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top
System.Windows.Forms.AnchorStyles.Right)));
this.buttonVarCreate.Location = new System.Drawing.Point(874, 87);
this.buttonVarCreate.Name = "buttonVarCreate";
this.buttonVarCreate.Size = new System.Drawing.Size(158, 24);
this.buttonVarCreate.TabIndex = 5;
this.buttonVarCreate.Text = "Создать переменную";
this.buttonVarCreate.UseVisualStyleBackColor = true;
this.buttonVarCreate.Click += new System.EventHandler(this.buttonVarCreate_Click);
//
// buttonLinkCreate
//

```

```

        this.buttonLinkCreate.Anchor
        ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top
        System.Windows.Forms.AnchorStyles.Right)));
        this.buttonLinkCreate.Location = new System.Drawing.Point(874, 117);
        this.buttonLinkCreate.Name = "buttonLinkCreate";
        this.buttonLinkCreate.Size = new System.Drawing.Size(158, 24);
        this.buttonLinkCreate.TabIndex = 7;
        this.buttonLinkCreate.Text = "Создать связь";
        this.buttonLinkCreate.UseVisualStyleBackColor = true;
        this.buttonLinkCreate.Click += new System.EventHandler(this.buttonLinkCreate_Click);
        //
        // splitContainer1
        //
        this.splitContainer1.Anchor
        ((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top
        System.Windows.Forms.AnchorStyles.Bottom)
        | System.Windows.Forms.AnchorStyles.Left)
        | System.Windows.Forms.AnchorStyles.Right)));
        this.splitContainer1.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D;
        this.splitContainer1.Location = new System.Drawing.Point(12, 27);
        this.splitContainer1.Name = "splitContainer1";
        //
        // splitContainer1.Panel1
        //
        this.splitContainer1.Panel1.Controls.Add(this.richTextBox1);
        this.splitContainer1.Panel1.Controls.Add(this.label1);
        this.splitContainer1.Panel1.Margin = new System.Windows.Forms.Padding(0, 0, 2, 0);
        //
        // splitContainer1.Panel2
        //
        this.splitContainer1.Panel2.Controls.Add(this.label2);
        this.splitContainer1.Panel2.Controls.Add(this.treeView1);
        this.splitContainer1.Panel2.Margin = new System.Windows.Forms.Padding(2, 0, 0, 0);
        this.splitContainer1.Size = new System.Drawing.Size(856, 649);
        this.splitContainer1.SplitterDistance = 419;
        this.splitContainer1.TabIndex = 8;
        //
        // richTextBox1
        //

```

```

        this.richTextBox1.Anchor
        ((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top
        System.Windows.Forms.AnchorStyles.Bottom)
        | System.Windows.Forms.AnchorStyles.Left)
        | System.Windows.Forms.AnchorStyles.Right)));
        this.richTextBox1.Location = new System.Drawing.Point(-1, 22);
        this.richTextBox1.Name = "richTextBox1";
        this.richTextBox1.ScrollBars = System.Windows.Forms.RichTextBoxScrollBars.Vertical;
        this.richTextBox1.Size = new System.Drawing.Size(417, 623);
        this.richTextBox1.TabIndex = 10;
        this.richTextBox1.Text = "";
        //
        // label2
        //
        this.label2.AutoSize = true;
        this.label2.Location = new System.Drawing.Point(3, 4);
        this.label2.Name = "label2";
        this.label2.Size = new System.Drawing.Size(82, 13);
        this.label2.TabIndex = 4;
        this.label2.Text = "BFT Элементы";
        //
        // menuStrip1
        //
        this.menuStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
            this.файлToolStripMenuItem,
            this.шрифтToolStripMenuItem});
        this.menuStrip1.Location = new System.Drawing.Point(0, 0);
        this.menuStrip1.Name = "menuStrip1";
        this.menuStrip1.RenderMode = System.Windows.Forms.ToolStripRenderMode.System;
        this.menuStrip1.Size = new System.Drawing.Size(1045, 24);
        this.menuStrip1.TabIndex = 9;
        this.menuStrip1.Text = "menuStrip1";
        //
        // файлToolStripMenuItem
        //
        this.файлToolStripMenuItem.DropDownItems.AddRange(new
        System.Windows.Forms.ToolStripItem[] {
            this.открытьToolStripMenuItem,
            this.сохранитьToolStripMenuItem});
        this.файлToolStripMenuItem.Name = "файлToolStripMenuItem";
        this.файлToolStripMenuItem.Size = new System.Drawing.Size(48, 20);

```

```

        this.файлToolStripMenuItem.Text = "Файл";
        //
        // открытьToolStripMenuItem
        //
        this.открытьToolStripMenuItem.Name = "открытьToolStripMenuItem";
        this.открытьToolStripMenuItem.Size = new System.Drawing.Size(133, 22);
        this.открытьToolStripMenuItem.Text = "Открыть";
        this.открытьToolStripMenuItem.Click += new
System.EventHandler(this.открытьToolStripMenuItem_Click);
        //
        // сохранитьToolStripMenuItem
        //
        this.сохранитьToolStripMenuItem.Name = "сохранитьToolStripMenuItem";
        this.сохранитьToolStripMenuItem.Size = new System.Drawing.Size(133, 22);
        this.сохранитьToolStripMenuItem.Text = "Сохранить";
        this.сохранитьToolStripMenuItem.Click += new
System.EventHandler(this.сохранитьToolStripMenuItem_Click);
        //
        // шрифтToolStripMenuItem
        //
        this.шрифтToolStripMenuItem.Name = "шрифтToolStripMenuItem";
        this.шрифтToolStripMenuItem.Size = new System.Drawing.Size(58, 20);
        this.шрифтToolStripMenuItem.Text = "Шрифт";
        this.шрифтToolStripMenuItem.Click += new
System.EventHandler(this.шрифтToolStripMenuItem_Click);
        //
        // label3
        //
        this.label3.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom
System.Windows.Forms.AnchorStyles.Right)));
        this.label3.AutoSize = true;
        this.label3.Location = new System.Drawing.Point(912, 661);
        this.label3.Name = "label3";
        this.label3.Size = new System.Drawing.Size(120, 13);
        this.label3.TabIndex = 10;
        this.label3.Text = "Пустовой В. А. ПИ-20г";
        //
        // Form1
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

```



```

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(1045, 688);
this.Controls.Add(this.label3);
this.Controls.Add(this.splitContainer1);
this.Controls.Add(this.buttonLinkCreate);
this.Controls.Add(this.button1);
this.Controls.Add(this.buttonConstCreate);
this.Controls.Add(this.buttonVarCreate);
this.Controls.Add(this.menuStrip1);
this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
this.MainMenuStrip = this.menuStrip1;
this.Name = "Form1";
this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
this.Text = "Редактор BFT";
this.Load += new System.EventHandler(this.Form1_Load);
this.splitContainer1.Panel1.ResumeLayout(false);
this.splitContainer1.Panel1.PerformLayout();
this.splitContainer1.Panel2.ResumeLayout(false);
this.splitContainer1.Panel2.PerformLayout();
((System.ComponentModel.ISupportInitialize)(this.splitContainer1)).EndInit();
this.splitContainer1.ResumeLayout(false);
this.menuStrip1.ResumeLayout(false);
this.menuStrip1.PerformLayout();
this.ResumeLayout(false);
this.PerformLayout();

}

```

#endregion

```

private System.Windows.Forms.Button button1;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.TreeView treeView1;
private System.Windows.Forms.Button buttonConstCreate;
private System.Windows.Forms.Button buttonVarCreate;
private System.Windows.Forms.Button buttonLinkCreate;
private System.Windows.Forms.SplitContainer splitContainer1;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.MenuStrip menuStrip1;
private System.Windows.Forms.ToolStripMenuItem файлToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem шрифтToolStripMenuItem;

```

```

        private System.Windows.Forms.ToolStripMenuItem открытьToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem сохранитьToolStripMenuItem;
        private System.Windows.Forms.RichTextBox richTextBox1;
        private System.Windows.Forms.Label label3;
    }
}

```

## Б.8 Файл Form2.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace KP_APP0
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();

            public void setGroupBoxState(bool gb1, bool gb2, bool gb3)
            {
                groupBox1.Visible = gb1;
                groupBox2.Visible = gb2;
                groupBox3.Visible = gb3;
            }

            private void buttonCancel_Click(object sender, EventArgs e)
            {
                this.Close();
            }

            private void buttonCreateConst_Click(object sender, EventArgs e)
            {
                try
                {
                    BFT_Creator bftC = new BFT_Creator();
                    Form1 firstForm = this.Owner as Form1;
                    if (String.IsNullOrEmpty(textBoxConstData.Text)
                        || textBoxConstData.Text.Contains("[") ||
                        textBoxConstData.Text.Contains("]"))
                        MessageBox.Show("Строка пуста или содержит запрещённые [ ]",
                            "Achtung!");
                    else
                    {
                        firstForm.insertText(bftC.createConst(textBoxConstData.Text));
                        this.Close();
                    }
                }
                catch (Exception exc)
                {
                    MessageBox.Show(exc.ToString(), "Achtung!");
                }
            }
        }
    }
}

```

```

    }
}

private void buttonCreateVar_Click(object sender, EventArgs e)
{
    try
    {
        BFT_Creator bftC = new BFT_Creator();
        Form1 firstForm = this.Owner as Form1;
        if (String.IsNullOrEmpty(textBoxVarName.Text) ||
String.IsNullOrEmpty(textBoxVarData.Text)
        || textBoxVarName.Text.Contains("[") ||
textBoxVarName.Text.Contains("]")
        || textBoxVarData.Text.Contains("[") ||
textBoxVarData.Text.Contains("]"))
            MessageBox.Show("Строки пусты или содержат запрещённые [ ]",
"Achtung!");
        else
        {
            firstForm.insertText(bftC.createVariable(textBoxVarName.Text,
textBoxVarData.Text));
            this.Close();
        }
    }
    catch (Exception exc)
    {
        MessageBox.Show(exc.ToString(), "Achtung!");
    }
}

private void buttonCreateLink_Click(object sender, EventArgs e)
{
    try
    {
        BFT_Creator bftC = new BFT_Creator();
        Form1 firstForm = this.Owner as Form1;
        if (String.IsNullOrEmpty(textBoxLinkName.Text) ||
String.IsNullOrEmpty(textBoxLinkData.Text)
        || textBoxLinkName.Text.Contains("[") ||
textBoxLinkName.Text.Contains("]")
        || textBoxLinkData.Text.Contains("[") ||
textBoxLinkData.Text.Contains("]"))
            MessageBox.Show("Строки пусты или содержат запрещённые [ ]",
"Achtung!");
        else
        {
            firstForm.insertText(bftC.createLink(textBoxLinkName.Text,
textBoxLinkData.Text));
            this.Close();
        }
    }
    catch (Exception exc)
    {
        MessageBox.Show(exc.ToString(), "Achtung!");
    }
}
}
}

```

## Б.9 Файл Form2.Designer.cs

```

namespace KP_ARPO
{
    partial class Form2
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.groupBox1 = new System.Windows.Forms.GroupBox();
            this.buttonCreateConst = new System.Windows.Forms.Button();
            this.textBoxConstData = new System.Windows.Forms.TextBox();
            this.label1 = new System.Windows.Forms.Label();
            this.buttonCancel = new System.Windows.Forms.Button();
            this.groupBox2 = new System.Windows.Forms.GroupBox();
            this.buttonCreateVar = new System.Windows.Forms.Button();
            this.textBoxVarData = new System.Windows.Forms.TextBox();
            this.button4 = new System.Windows.Forms.Button();
            this.label3 = new System.Windows.Forms.Label();
            this.textBoxVarName = new System.Windows.Forms.TextBox();
            this.label2 = new System.Windows.Forms.Label();
            this.groupBox3 = new System.Windows.Forms.GroupBox();
            this.buttonCreateLink = new System.Windows.Forms.Button();
            this.textBoxLinkData = new System.Windows.Forms.TextBox();
            this.buttonCancelLink = new System.Windows.Forms.Button();
            this.label4 = new System.Windows.Forms.Label();
            this.label5 = new System.Windows.Forms.Label();
            this.textBoxLinkName = new System.Windows.Forms.TextBox();
            this.groupBox1.SuspendLayout();
            this.groupBox2.SuspendLayout();
            this.groupBox3.SuspendLayout();
            this.SuspendLayout();
            //
            // groupBox1
            //
            this.groupBox1.Anchor =
                ((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top |
                System.Windows.Forms.AnchorStyles.Bottom)

```

```

| System.Windows.Forms.AnchorStyles.Left)
| System.Windows.Forms.AnchorStyles.Right));
this.groupBox1.Controls.Add(this.buttonCreateConst);
this.groupBox1.Controls.Add(this.textBoxConstData);
this.groupBox1.Controls.Add(this.label1);
this.groupBox1.Controls.Add(this.buttonCancel);
this.groupBox1.Location = new System.Drawing.Point(12, 12);
this.groupBox1.Name = "groupBox1";
this.groupBox1.Size = new System.Drawing.Size(202, 161);
this.groupBox1.TabIndex = 1;
this.groupBox1.TabStop = false;
this.groupBox1.Text = "Константа";
this.groupBox1.Visible = false;
//
// buttonCreateConst
//
this.buttonCreateConst.Location = new System.Drawing.Point(5, 135);
this.buttonCreateConst.Name = "buttonCreateConst";
this.buttonCreateConst.Size = new System.Drawing.Size(90, 20);
this.buttonCreateConst.TabIndex = 1;
this.buttonCreateConst.Text = "Создать";
this.buttonCreateConst.UseVisualStyleBackColor = true;
this.buttonCreateConst.Click += new System.EventHandler(this.buttonCreateConst_Click);
//
// textBoxConstData
//
this.textBoxConstData.Location = new System.Drawing.Point(5, 40);
this.textBoxConstData.Name = "textBoxConstData";
this.textBoxConstData.Size = new System.Drawing.Size(186, 20);
this.textBoxConstData.TabIndex = 0;
//
// label1
//
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(5, 20);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(55, 13);
this.label1.TabIndex = 1;
this.label1.Text = "Значение";
//
// buttonCancel
//
this.buttonCancel.Location = new System.Drawing.Point(103, 135);
this.buttonCancel.Name = "buttonCancel";
this.buttonCancel.Size = new System.Drawing.Size(90, 20);
this.buttonCancel.TabIndex = 2;
this.buttonCancel.Text = "Отмена";
this.buttonCancel.UseVisualStyleBackColor = true;
this.buttonCancel.Click += new System.EventHandler(this.buttonCancel_Click);
//
// groupBox2
//
this.groupBox2.Anchor =
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Bottom)
| System.Windows.Forms.AnchorStyles.Left)
| System.Windows.Forms.AnchorStyles.Right)));
this.groupBox2.Controls.Add(this.buttonCreateVar);
this.groupBox2.Controls.Add(this.textBoxVarData);
this.groupBox2.Controls.Add(this.button4);
this.groupBox2.Controls.Add(this.label3);
this.groupBox2.Controls.Add(this.textBoxVarName);

```

```

this.groupBox2.Controls.Add(this.label2);
this.groupBox2.Location = new System.Drawing.Point(12, 12);
this.groupBox2.Name = "groupBox2";
this.groupBox2.Size = new System.Drawing.Size(202, 161);
this.groupBox2.TabIndex = 1;
this.groupBox2.TabStop = false;
this.groupBox2.Text = "Переменная";
this.groupBox2.Visible = false;
//
// buttonCreateVar
//
this.buttonCreateVar.Location = new System.Drawing.Point(5, 135);
this.buttonCreateVar.Name = "buttonCreateVar";
this.buttonCreateVar.Size = new System.Drawing.Size(90, 20);
this.buttonCreateVar.TabIndex = 2;
this.buttonCreateVar.Text = "Создать";
this.buttonCreateVar.UseVisualStyleBackColor = true;
this.buttonCreateVar.Click += new System.EventHandler(this.buttonCreateVar_Click);
//
// textBoxVarData
//
this.textBoxVarData.Location = new System.Drawing.Point(5, 90);
this.textBoxVarData.Name = "textBoxVarData";
this.textBoxVarData.Size = new System.Drawing.Size(186, 20);
this.textBoxVarData.TabIndex = 1;
//
// button4
//
this.button4.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bottom |
System.Windows.Forms.AnchorStyles.Right)));
this.button4.Location = new System.Drawing.Point(107, 135);
this.button4.Name = "button4";
this.button4.Size = new System.Drawing.Size(90, 20);
this.button4.TabIndex = 3;
this.button4.Text = "Отмена";
this.button4.UseVisualStyleBackColor = true;
this.button4.Click += new System.EventHandler(this.buttonCancel_Click);
//
// label3
//
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(5, 70);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(55, 13);
this.label3.TabIndex = 0;
this.label3.Text = "Значение";
//
// textBoxVarName
//
this.textBoxVarName.Location = new System.Drawing.Point(5, 40);
this.textBoxVarName.Name = "textBoxVarName";
this.textBoxVarName.Size = new System.Drawing.Size(186, 20);
this.textBoxVarName.TabIndex = 0;
//
// label2
//
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(5, 20);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(57, 13);
this.label2.TabIndex = 0;

```

```

        this.label2.Text = "Название";
        //
        // groupBox3
        //
        this.groupBox3.Anchor =
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Bottom)
| System.Windows.Forms.AnchorStyles.Left)
| System.Windows.Forms.AnchorStyles.Right)));
        this.groupBox3.Controls.Add(this.buttonCreateLink);
        this.groupBox3.Controls.Add(this.textBoxLinkData);
        this.groupBox3.Controls.Add(this.buttonCancelLink);
        this.groupBox3.Controls.Add(this.label4);
        this.groupBox3.Controls.Add(this.label5);
        this.groupBox3.Controls.Add(this.textBoxLinkName);
        this.groupBox3.Location = new System.Drawing.Point(12, 12);
        this.groupBox3.Name = "groupBox3";
        this.groupBox3.Size = new System.Drawing.Size(202, 161);
        this.groupBox3.TabIndex = 2;
        this.groupBox3.TabStop = false;
        this.groupBox3.Text = "Связь";
        this.groupBox3.Visible = false;
        //
        // buttonCreateLink
        //
        this.buttonCreateLink.Location = new System.Drawing.Point(5, 135);
        this.buttonCreateLink.Name = "buttonCreateLink";
        this.buttonCreateLink.Size = new System.Drawing.Size(90, 20);
        this.buttonCreateLink.TabIndex = 2;
        this.buttonCreateLink.Text = "Создать";
        this.buttonCreateLink.UseVisualStyleBackColor = true;
        this.buttonCreateLink.Click += new System.EventHandler(this.buttonCreateLink_Click);
        //
        // textBoxLinkData
        //
        this.textBoxLinkData.Location = new System.Drawing.Point(5, 90);
        this.textBoxLinkData.Name = "textBoxLinkData";
        this.textBoxLinkData.Size = new System.Drawing.Size(186, 20);
        this.textBoxLinkData.TabIndex = 1;
        //
        // buttonCancelLink
        //
        this.buttonCancelLink.Location = new System.Drawing.Point(103, 135);
        this.buttonCancelLink.Name = "buttonCancelLink";
        this.buttonCancelLink.Size = new System.Drawing.Size(90, 20);
        this.buttonCancelLink.TabIndex = 3;
        this.buttonCancelLink.Text = "Отмена";
        this.buttonCancelLink.UseVisualStyleBackColor = true;
        this.buttonCancelLink.Click += new System.EventHandler(this.buttonCancel_Click);
        //
        // label4
        //
        this.label4.AutoSize = true;
        this.label4.Location = new System.Drawing.Point(5, 70);
        this.label4.Name = "label4";
        this.label4.Size = new System.Drawing.Size(55, 13);
        this.label4.TabIndex = 3;
        this.label4.Text = "Значения";
        //
        // label5
        //
        this.label5.AutoSize = true;

```

```

        this.label5.Location = new System.Drawing.Point(5, 20);
        this.label5.Name = "label5";
        this.label5.Size = new System.Drawing.Size(57, 13);
        this.label5.TabIndex = 4;
        this.label5.Text = "Названия";
        //
        // textBoxLinkName
        //
        this.textBoxLinkName.Location = new System.Drawing.Point(5, 40);
        this.textBoxLinkName.Name = "textBoxLinkName";
        this.textBoxLinkName.Size = new System.Drawing.Size(186, 20);
        this.textBoxLinkName.TabIndex = 0;
        //
        // Form2
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(225, 181);
        this.Controls.Add(this.groupBox3);
        this.Controls.Add(this.groupBox1);
        this.Controls.Add(this.groupBox2);
        this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.Fixed3D;
        this.Name = "Form2";
        this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
        this.Text = "Создание";
        this.groupBox1.ResumeLayout(false);
        this.groupBox1.PerformLayout();
        this.groupBox2.ResumeLayout(false);
        this.groupBox2.PerformLayout();
        this.groupBox3.ResumeLayout(false);
        this.groupBox3.PerformLayout();
        this.ResumeLayout(false);
    }

    #endregion

    private System.Windows.Forms.GroupBox groupBox1;
    private System.Windows.Forms.Button buttonCancel;
    private System.Windows.Forms.GroupBox groupBox2;
    private System.Windows.Forms.GroupBox groupBox3;
    private System.Windows.Forms.Button buttonCreateLink;
    private System.Windows.Forms.TextBox textBoxLinkData;
    private System.Windows.Forms.Button buttonCancelLink;
    private System.Windows.Forms.Label label4;
    private System.Windows.Forms.Label label5;
    private System.Windows.Forms.TextBox textBoxLinkName;
    private System.Windows.Forms.Button buttonCreateVar;
    private System.Windows.Forms.TextBox textBoxVarData;
    private System.Windows.Forms.Button button4;
    private System.Windows.Forms.Label label3;
    private System.Windows.Forms.TextBox textBoxVarName;
    private System.Windows.Forms.Label label2;
    private System.Windows.Forms.TextBox textBoxConstData;
    private System.Windows.Forms.Label label1;
    private System.Windows.Forms.Button buttonCreateConst;
}
}

```