

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Волгоградский государственный технический университет»

Кафедра «Программное обеспечение автоматизированных систем»

УТВЕРЖДАЮ:
Зав. Кафедрой ПОАС
_____ Ю.А. Орлова
«_____» _____ 2023 г.

Логическая модель разрешения имен в программе

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

ВКРБ–09.03.04–10.19–13–23–91

Листов 24

Руководитель работы
_____ О. А. Сычев
«_____» _____ 2023 г.

Нормоконтролер
_____ А. С. Кузнецова
«_____» _____ 2023 г.

Исполнитель
студент группы ПрИн-467
_____ А. Г. Сидор
«_____» _____ 2023 г.

Волгоград, 2023

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Волгоградский государственный технический университет»

Кафедра «Программное обеспечение автоматизированных систем»

УТВЕРЖДАЮ:
Зав. Кафедрой ПОАС
_____ Ю.А. Орлова
« ____ » _____ 2023 г.

Логическая модель разрешения имен в программе

ЛИСТ УТВЕРЖДЕНИЯ

А.В.00001-01 91 01-1-ЛУ

Листов 1

Руководитель работы
_____ О. А. Сычев
« ____ » _____ 2023 г.

Нормоконтролер
_____ А. С. Кузнецова
« ____ » _____ 2023 г.

Исполнитель
студент группы ПрИн-467
_____ А. Г. Сидор
« ____ » _____ 2023 г.

Волгоград, 2023 г.

Аннотация

Документ представляет собой техническое задание к выпускной работе бакалавра на тему «Логическая модель разрешения имен в программе». В документе изложены основания и назначения разработки программы, требования к разрабатываемой программе, технико-экономические показатели. Документ включает в себя страниц – 24, приложений – 6, рисунков – 6.

Ключевые слова: логическая модель, дерево решений, разрешение имен в программе, тренажер.

Содержание

Введение	6
1.1 Наименование программы	6
1.2 Область применения	6
2 Основание для разработки	6
3 Назначение разработки	6
4 Требования к программе	7
4.1 Требования к функциональным характеристикам	7
4.2 Требования к надежности	8
4.2.1 Требования к обеспечению надежного функционирования программы	8
4.2.2 Время восстановления после отказа	8
4.2.3 Отказы из-за некорректных действий пользователя	9
4.3 Требования к условиям эксплуатации	9
4.3.1 Климатические условия эксплуатации	9
4.3.2 Требования к квалификации и численности персонала	10
4.4 Требования к составу и параметрам технических средств	10
4.5.1 Требования к информационным структурам и методам решения	10
4.5.2 Требования к исходным кодам и языкам программирования	11
4.5.3 Требования к программным средствам, используемым программой	11
4.5.4 Требования к защите информации	11
4.6 Требования к маркировке и упаковке	11
4.7 Требования к транспортированию и хранению	12
4.8 Специальные требования	12
5 Требования к программной документации	12
6 Техничко-экономические показатели	12
6.1 Экономические преимущества разработки	12

7 Стадии и этапы разработки	13
7.1 Стадии разработки	13
7.2 Содержание работ по этапам	13
8 Порядок контроля и приемки	14
8.1 Виды испытаний	14
Приложение А.1 - Виды определяемых ошибок	15
Приложение А.2 - Описание задачи	16
Приложение Б.1 - Диаграммы вариантов использования	17
Приложение Б.2 - Сценарии вариантов использования	18
Приложение Б.3 - Макеты экранных форм	19
Приложение Б.4 - Структура и формат данных	24

Введение

1.1 Наименование программы

Наименование программы: «Name resolution trainer».

1.2 Область применения

Данная программа будет разрабатываться для применения в области образования, а точнее, обучению программированию.

2 Основание для разработки

Приложение разрабатывается на основании приказа №1235-ст от 06.09.2022, утвержденного кафедрой ПОАС по теме «Логическая модель разрешения имен в программе».

3 Назначение разработки

Улучшить качество интеллектуального обучения программированию в разделе умений по разрешению имен в программе. Данная программа может использоваться, как обучающая, в образовательных учреждениях, а также в домашних условиях для саморазвития.

4 Требования к программе

4.1 Требования к функциональным характеристикам

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

- Программа должна представлять пользователю и решать задачу (см. Приложение А.2);
- варианты ответов (см. Приложение Б.3, в поле с кодом названия переменных подчеркнутые и выделенные синим цветом) должны быть кликабельными, чтобы пользователь мог выбрать их посредством нажатия ЛКМ;
- программа должна проверять выбранный ответ пользователя (частичный ответ, один из вариантов ответов, упомянутый в пункте 2);
- программа должна выводить понятное сообщение об ошибке с пояснением в случае неверного частичного ответа пользователя;
- программа должна определять наиболее распространенные ошибки пользователей (см. Приложение А.1);
- программа должна проверять полный ответ пользователя (когда пользователь выбрал все частичные варианты ответов), и в случае неверного ответа выдавать ошибку;
- при запросе пользователя программа должна показывать подсказки по выполнению задачи;
- обеспечить возможность интеграции с кодом генерации наводящих вопросов и их показа;
- программа должна вычислять правильный ответ и использовать его при проверке ответов пользователя.

4.2 Требования к надежности

4.2.1 Требования к обеспечению надежного функционирования программы

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением пользователем совокупности организационно-технических мероприятий, перечень которых приведен ниже:

- организацией бесперебойного питания технических средств;
- использованием лицензионного программного обеспечения;
- регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г. «Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;
- регулярным выполнением требований ГОСТ 51188-98 «Защита информации. Испытания программных средств на наличие компьютерных вирусов».

4.2.2 Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать времени, необходимого на перезагрузку операционной системы и запуск программы, при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

4.2.3 Отказы из-за некорректных действий пользователя

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу конечного пользователя без предоставления ему административных привилегий.

4.3 Требования к условиям эксплуатации

4.3.1 Климатические условия эксплуатации

Климатические условия эксплуатации, при которых программа должна функционировать, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

4.3.2 Требования к квалификации и численности персонала

Для использования программного средства требуется один человек, являющийся пользователем ПК.

4.4 Требования к составу и параметрам технических средств

В состав технических средств должен входить персональный компьютер, включающий в себя:

- процессор Intel Core i5 4th-gen и лучше;
- оперативную память объемом 4 ГБ и более;
- операционную систему Windows 8.1 и выше.

В состав периферийных устройств должны входить:

- монитор;
- клавиатура;
- мышь.

4.5.1 Требования к информационным структурам и методам решения

Требования к информационным структурам и методам решения не предъявляются.

4.5.2 Требования к исходным кодам и языкам программирования

Требования к исходным кодам и языкам программирования не предъявляются.

4.5.3 Требования к программным средствам, используемым программой

Программные средства, используемые программой, должны быть представлены:

- лицензионной локализованной версией операционной системы Windows 8.1 и выше;
- виртуальной машиной JAVA с JDK 14 и выше.

4.5.4 Требования к защите информации

Требования к защите информации не предъявляются.

4.6 Требования к маркировке и упаковке

Требования к маркировке и упаковке не предъявляются.

4.7 Требования к транспортированию и хранению

Требования к транспортированию и хранению не предъявляются.

4.8 Специальные требования

Специальные требования не предъявляются.

5 Требования к программной документации

Состав разрабатываемой программной документации должен включать в себя:

- техническое задание, оформленное в соответствии с ГОСТ 19;
- пояснительную записку, оформленную в соответствии с СТП 24;
- руководство системного программиста, оформленное в соответствии с ГОСТ 19.

6 Техничко-экономические показатели

6.1 Экономические преимущества разработки

Ориентировочная экономическая эффективность не рассчитывается.

7 Стадии и этапы разработки

7.1 Стадии разработки

Разработка приложения должна быть проведена в 5 стадий:

- анализ и сбор информации по предметной области;
- проектирование приложения;
- разработка приложения;
- тестирование приложения;
- документирование приложения.

7.2 Содержание работ по этапам

Этапы работы:

- Анализ и сбор информации по предметной области:
 1. поиск и чтение статей по предметной области.
 2. поиск конкурентов по предметной области.
- Проектирование приложения:
 1. разработка диаграммы и сценариев использования.
 2. проектирование макета интерфейса.
- Разработка приложения:
 1. разработка модели решения:
 - a. разработка логических моделей в виде деревьев решений.
 - b. разработка модели данных.
 2. разработка клиента приложения.
 3. разработка сервера приложения.
- Тестирование приложения:

1. тестирование приложения на заранее подготовленных входных данных.

2. тестирование приложения с участием студентов ВолГТУ, обучающихся на направлении «Программная инженерия».

– Документирование приложения:

1. разработка ПЗ приложения.

8 Порядок контроля и приемки

8.1 Виды испытаний

Программа сдаётся на проверку заказчику 05.06.2023. При обнаружении в программе ошибок или недостатков исполнитель обязуется устранить их в недельный срок и предоставить программу на повторную проверку.

Программа сдаётся на проверку независимому тестировщику не позднее 09.06.2023. Результаты тестирования предоставляются на защите дипломного проекта членам ГАК.

Виды определяемых ошибок

– Ошибки связанные с областью видимости.

Примеры возможных ошибок:

– в начале префикса «::», а пользователь выбирает объявление из локальной области видимости.

– Ошибки связанные с видимостью в локальной области.

Примеры возможных ошибок:

– пользователь выбрал переменную, объявленную ниже места использования.

– пользователь выбрал переменную, объявленную выше места использования, но которая закончила свое существование.

– Ошибки связанные с перекрытием.

Примеры возможных ошибок:

– пользователь выбрал перекрываемую переменную (не заметил перекрытие).

– пользователь выбрал потенциально перекрывающую переменную.

– Ошибки связанные с названием префикса.

Примеры возможных ошибок:

– пользователь выбрал переменную, у которой префикс отличается от указанного в условии.

Описание задачи

Вопрос задачи: «Какие переменные видны в N строке и с префиксом P?».

Данные, предоставляемые пользователю, для решения задачи:

- код программы с выделенными именами переменных в их объявлении;

- номер строки N, которая фигурирует в условии;

- префикс P в строковой форме.

Задача поддерживает наличие в коде:

- функций;

- контекстов:

1. глобальный;

2. локальный;

- типов данных:

3. класс;

4. структура;

5. перечисление;

- пространств имен;

- переменных;

- типовых конструкций (например: ветвления, циклы, блоки и т.д.).

Задача не поддерживает наличие в коде:

- операторов using.

Префикс переменной строится из операторов «::», типов данных, перечисленных выше, и пространств имен.

Код для условия задачи пишется на языке высокого уровня C++ стандарта 2014 года.

Диаграммы вариантов использования

На рисунке Б.1.1 представлена диаграмма вариантов использования.

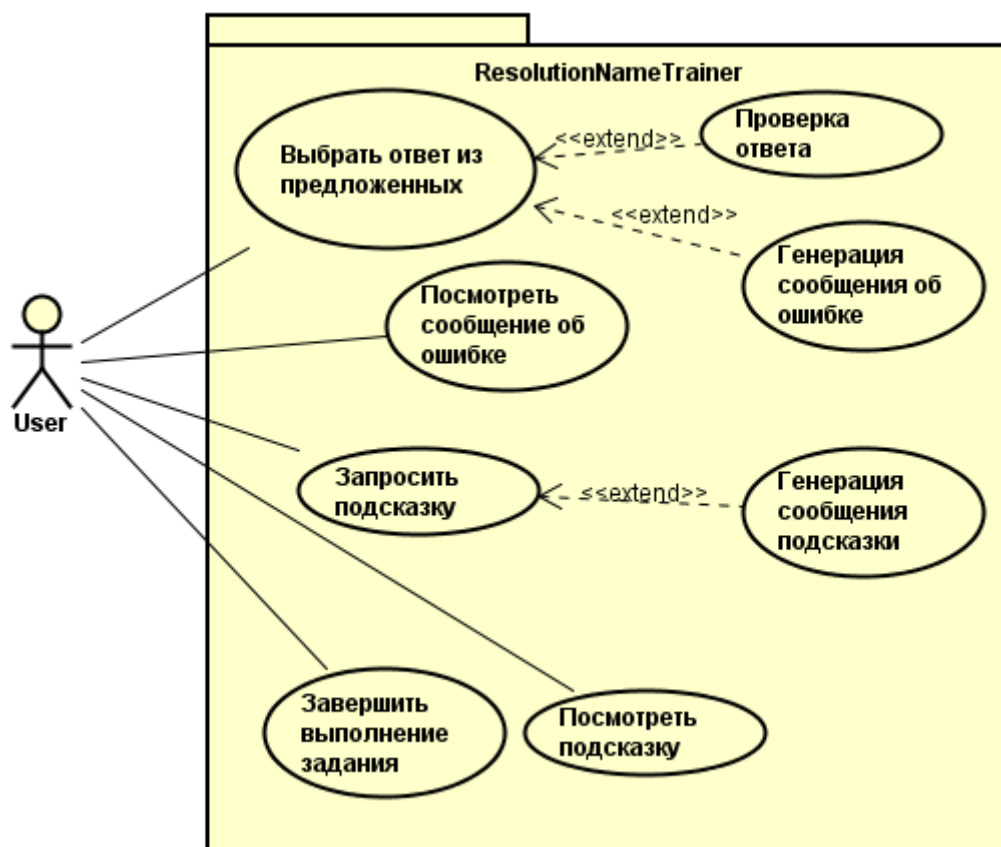


Рисунок Б.1.1 – Диаграмма вариантов использования программы

Сценарии вариантов использования

Пользователь выбирает вариант из предложенных:

- пользователь нажимает ЛКМ по выделенной переменной в коде;
- в текстовом поле под кодом условия показывается результат (верно / сообщение об ошибке с конкретным описанием).

Пользователь запрашивает подсказку:

- во время выполнения задания пользователь нажимает ЛКМ по кнопке «Подсказка»;
- в текстовом поле под кодом условия показывается подсказка, наводящая на один из ответов.

Пользователь завершает выполнение задания с ошибкой:

- пользователь нажимает ЛКМ по кнопке «Завершить выполнение»;
- кнопка «Завершить выполнение» блокируется до следующего первого выбора ответа пользователя;
- в текстовом поле под кодом условия показывается сообщение об ошибке.

Пользователь завершает выполнение задания с верным ответом:

- пользователь нажимает ЛКМ по кнопке «Завершить выполнение»;
- на экране всплывает сообщение о верном решении задачи;
- входные данные задачи меняются на следующие;
- все текстовые поля кроме условий задачи сворачиваются.

Пользователь завершает решение всех задач:

- пользователь отправляет верный ответ на последнюю задачу;
- экран меняется на завершающий с кнопкой «Прорешать еще раз».

Макеты экранных форм

Какие переменные видны в 11 строке с префиксом (a::b)?

```
1. #include <iostream>
2. namespace a {
3.     class b {
4.     public:
5.         static const int d = 0;
6.         static const int c = 1;
7.     };
8.     int value = 2;
9. }
10. void main() {
11.
12. }
```

Ошибка...

Подсказка

Завершить выполнение

Рисунок Б.3.1 - Макет основного экрана с текстовыми полем ошибки

Какие переменные видны в 11 строке с префиксом (a::b)?

```
1. #include <iostream>
2. namespace a {
3.     class b {
4.     public:
5.         static const int d = 0;
6.         static const int c = 1;
7.     };
8.     int value = 2;
9. }
10. void main() {
11.
12. }
```

Подсказка...

Подсказка

Завершить выполнение

Рисунок Б.3.2 - Макет основного экрана с текстовыми полем подсказки

Какие переменные видны в 11 строке с префиксом (a::b)?

```
1. #include <iostream>
2. namespace a {
3.     class b {
4.     public:
5.         static const int d = 0;
6.         static const int c = 1;
7.     };
8.     int value = 2;
9. }
10. void main() {
11.
12. }
```

Ответ верен!!!

Подсказка

Завершить выполнение

Рисунок Б.3.3 - Макет основного экрана с текстовыми полем верного ответа

Какие переменные видны в 11 строке с префиксом (a::b)?

```
1. #include <iostream>
2. namespace a {
3.     class b {
4.     public:
5.         static const int d = 0;
6.         static const int c = 1;
7.     };
8.     int value = 2;
9. }
10. void main() {
11.
12. }
```

Подсказка

Завершить выполнение

Рисунок Б.3.4 - Макет основного экрана без дополнительных текстовых полей

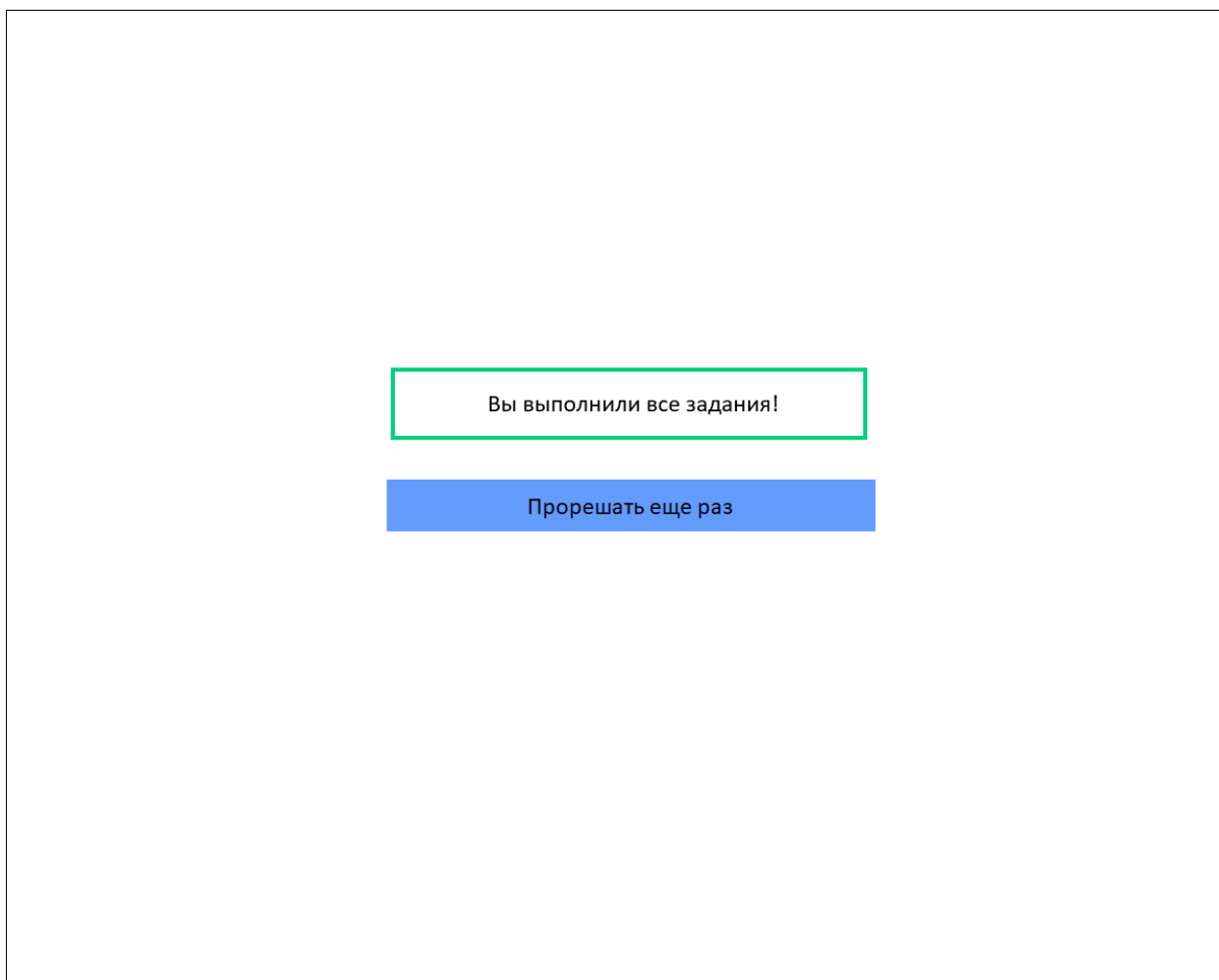


Рисунок Б.3.5 - Макет завершающего экрана

Структура и формат данных

Данные необходимые для решения задачи (см. Приложение А.2) программой:

- Упрощенное дерево кода в виде RDF графа;
- Префикс в строковой форме;
- Номер строки в виде числа.