|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_\_**ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_\_\_\_\_\_\_\_\_**КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

НАПРАВЛЕНИЕ ПОДГОТОВКИ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К КУРСОВОЙ РАБОТЕ***

***НА ТЕМУ:***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***Разработка учебного тренажера по булевым функциям «Булеан»\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

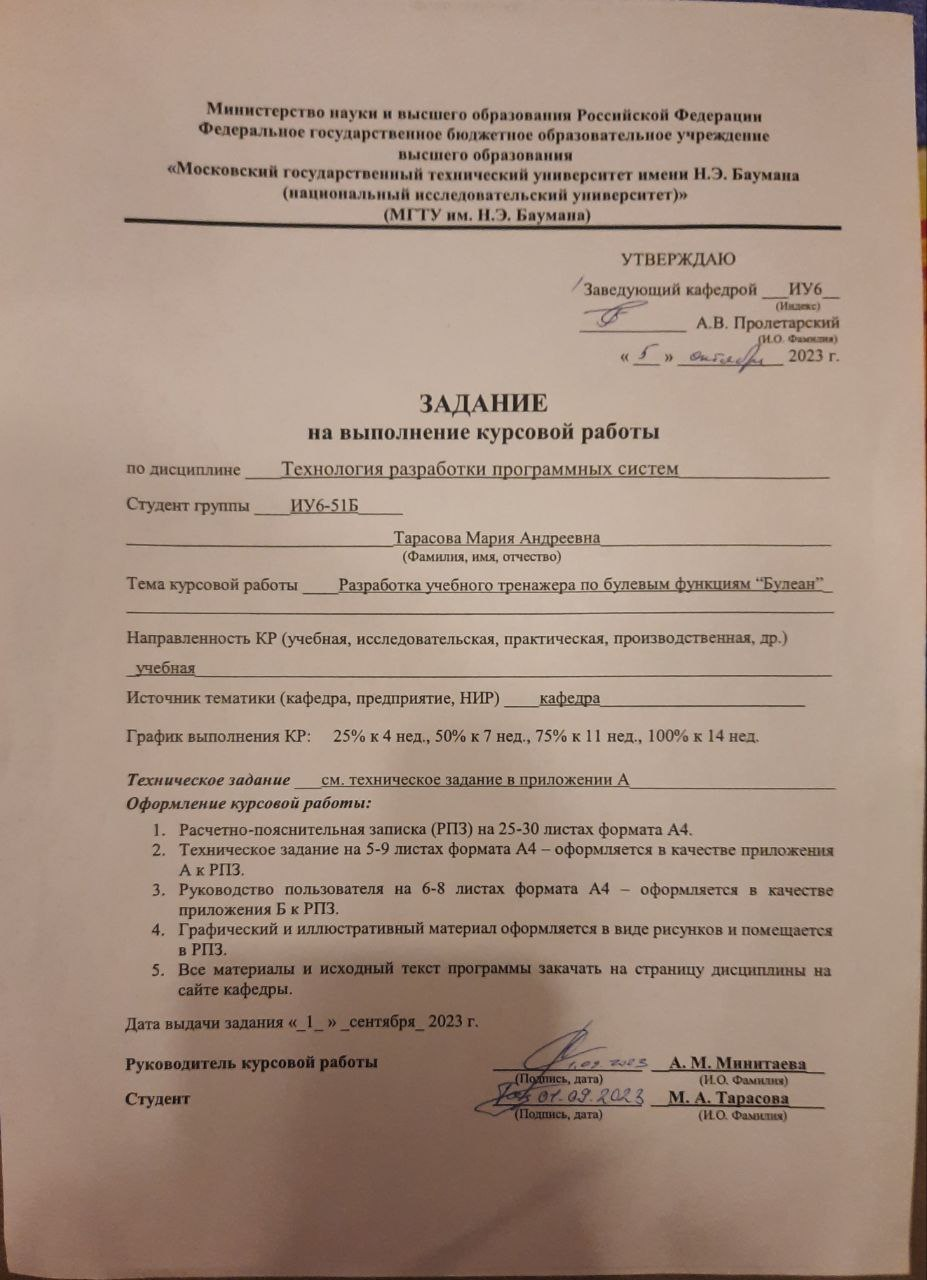
Студент \_\_\_\_\_ИУ6-51Б\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_М. А. Тарасова\_\_\_**

(Группа) (Подпись, дата) (И.О. Фамилия)

Руководитель курсовой работы **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_А. М. Минитаева\_\_**

(Подпись, дата) (И.О. Фамилия)

*2023 г.*



**РЕФЕРАТ**

РПЗ 63 страницы, 3 части, 16 рисунков, 9 таблиц, 5 источников, 3 приложения.

ТРЕНАЖЁР, БУЛЕВЫ ФУНКЦИИ, ПРЕОБРАЗОВАНИЯ.

Цель работы – проектирование и реализация приложения «Булеан», предназначенного для тренировки в преобразованиях булевых функций.

В результате разработки было спроектировано приложение, позволяющее пользователю сгенерировать булеву функцию и выполнить её преобразования, а также было проведено тестирование данного программного продукта методами чёрного ящика.

Приложение предназначено для использования студентами 1-2 курса для подготовки к контрольным мероприятиям и экзамену по дискретной математике.

ОГЛАВЛЕНИЕ

[ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ 5](#_Toc154023210)

[ВВЕДЕНИЕ 6](#_Toc154023211)

[1 Анализ требований и уточнение спецификаций 7](#_Toc154023212)

[1.1 Анализ задания и выбор технологии, языка и среды разработки 7](#_Toc154023213)

[1.2 Разработка диаграммы вариантов использования 7](#_Toc154023214)

[1.3 Разработка концептуальной модели предметной области 10](#_Toc154023215)

[1.4 Выбор методов решения задачи и разработка основных алгоритмов предметной области 11](#_Toc154023216)

[2 Проектирование структуры и компонентов программного продукта 14](#_Toc154023217)

[2.1 Разработка диаграммы состояний интерфейса 14](#_Toc154023218)

[2.2 Разработка структурной схемы программного продукта 15](#_Toc154023219)

[2.3 Разработка диаграммы классов и алгоритмы методов классов 17](#_Toc154023220)

[2.4 Разработка форм интерфейса 25](#_Toc154023221)

[3 Выбор стратегии тестирования и разработка тестов 30](#_Toc154023222)

[ЗАКЛЮЧЕНИЕ 36](#_Toc154023223)

[ПРИЛОЖЕНИЕ А 38](#_Toc154023224)

[ПРИЛОЖЕНИЕ Б 44](#_Toc154023225)

[ПРИЛОЖЕНИЕ В 53](#_Toc154023226)

# ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ТЗ – техническое задание.

Интерфейс – графическое представление приложения пользователю.

ООП – объектно-ориентированное программирование.

СДНФ – совершенная дизъюнктивная нормальная форма (дизъюнкции конъюнкций, при которых функция равна 1).

СКНФ – совершенная конъюктивная нормальная форма (конъюнкции дизъюнкций, при которых функция равна 0).

Минимальная ДНФ – дизъюнктивная нормальная форма, содержащая наименьшее число литералов.

Минимальная КНФ - конъюктивная нормальная форма, содержащая наименьшее число литералов.

DOM-дерево - интерфейс к HTML-документу.

# ВВЕДЕНИЕ

Данная работа выполнялась с целью подготовки студентов к экзаменам по дискретной математике и развитию у них навыков работы с булевыми функциями.

Её актуальность заключается в необходимости студентов потренироваться и проверить свои практические навыки в преобразовании булевых функций при написании работ по дискретной математике, построения схем, состоящих из логических элементов в схемотехнике и других областях.

Уникальность данной разработки состоит в том, что похожие тренажёры не были найдены на просторах Интернета, кроме тренажёра минимизации булевых функций на гиперкубе. В остальных случаях студентам предлагались лишь преобразователи булевых функций или генераторы булевых функций, предоставляющие только таблицу истинности на выходе программы, без возможности проверить, правильно ли студент вычислил минимальную ДНФ, записал функцию в базисе Жегалкина и так далее.

В данной же работе можно сгенерировать булеву функцию по выбранным параметрам, а также проверить результаты преобразований на корректность, что является необходимым пунктом в обучающем тренажёре.

# Анализ требований и уточнение спецификаций

## 1.1 Анализ задания и выбор технологии, языка и среды разработки

Для разработки данного тренажёра было решено использовать объектно-ориентированный подход, так как он позволяет проще взаимодействовать с командами пользователя, а также работать со структурами данных внутри самой программы.

В качестве языка разработки был использован JavaScript, так как он поддерживает технологию ООП и сам построен на этой технологии, а также с его помощи можно работать с DOM-деревом.

В качестве среды разработки был использован VisualStudio Code. Он позволяет вести разработку, открывая несколько файлов одновременно, а также поддерживает наглядное ведение дерева Git.

## 1.2 Разработка диаграммы вариантов использования

Одним из первых шагов при проектировании данного тренажёра является определение основных вариантов его использования, для этого были разработаны варианты использования и их диаграмма, которая отражает варианты взаимодействие пользователя с разрабатываемым приложением.

В процессе анализа технического задания были выявлены следующие варианты взаимодействия пользователя с приложением:

- генерация булевой функции;

- решение заданий по булевой функции.

Рассмотрим основные вариант использования, который описан в таблицах 1 - 4.

Таблица 1 – вариант использования генерации булевой функции.

|  |  |
| --- | --- |
| Название варианта | Генерация булевой функции и заданий к ней |
| Цель | Сгенерировать булеву функцию и задания к ней |
| Действующие лица | Пользователь |
| Краткое описание | Выбор типов заданий, количества переменных и операций, генерация на их основе функции и самих заданий |
| Тип | Основной |

Таблица 2 – описание варианта использования генерация булевой функции.

|  |  |
| --- | --- |
| Действие пользователя | Отклик системы |
| 1. Пользователь выбирает типы заданий, количество переменных и операций и нажимает кнопку «Сгенерировать».  3. Пользователь может посмотреть функцию и, если она ему не понравится, снова нажать кнопку «Сгенерировать». | 2. Приложение принимает введённые пользователем параметры, на их основе генерирует булеву функцию и выводит её на экран вместе с заданиями к ней.  4. Приложение генерирует функцию на основе уже введённых параметров и выводит на экран новую булеву функцию вместе с заданиями. |

Таблица 3 – вариант использования решения заданий по булевой функции.

|  |  |
| --- | --- |
| Название варианта | Решение заданий по булевой функции |
| Цель | Проверить ответы пользователя |
| Действующие лица | Пользователь |
| Краткое описание | Ввод ответов в нужные поля, проверка их на правильность, просмотр правильного ответа. |
| Тип | Основной |

Таблица 4 – описание варианта использования решения заданий по булевой функции.

|  |  |
| --- | --- |
| Действие пользователя | Отклик системы |
| 1. Пользователь решает задания, сгенерированные для данной булевой функции, вводит ответ в нужное поле и нажимает кнопку «Проверить».  3. Пользователь может перерешать задание и снова проверить ответ. Также пользователь может посмотреть правильный ответ. | 2. Приложение принимает введённые данные, сверяет их с рассчитанными преобразованиями функции внутри приложения и, если всё сходится, выводит на экран надпись «Правильно». Если ответ неверный, на экран выводится надпись «Неправильно». Также предлагается пользователю посмотреть правильный ответ. |

Диаграмма вариантов использования представлена на рисунке 1.

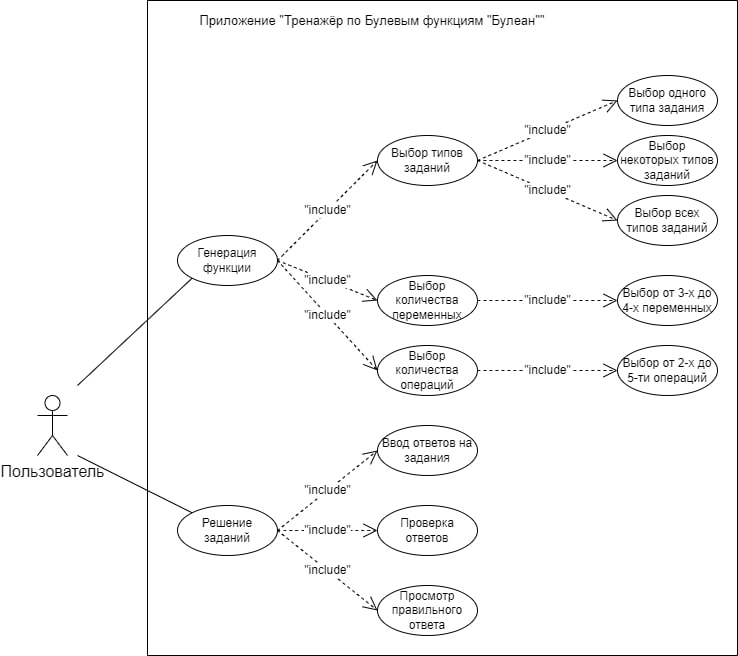


Рисунок 1 – Диаграмма вариантов использования

## 1.3 Разработка концептуальной модели предметной области

В процессе разработки приложения были выявлены основные объекты:

- задание;

- булева функция;

- операнды.

Из заданий строится список заданий, который в свою очередь потом выводится пользователю. Задания наследуются все от родительского класса Задание.

Сама булева функция является деревом, состоящим из операндов-нодов, которые в свою очередь могут быть операндами-переменными и операндами-операциями. Сами операнды наследуются от класса Операнд.

Задания используют дерево булевой функции для сравнения ответов пользователя с просчитанными внутренними методами дерева значениями.

На основе анализа основных объектов была создана концептуальная диаграмма классов предметной области, представленная на рисунке 2.

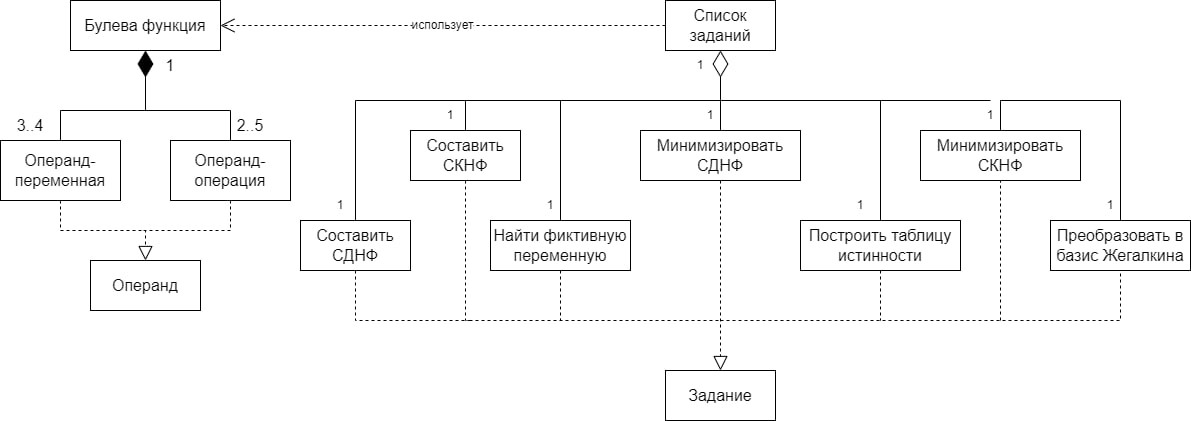


Рисунок 2 – Концептуальная диаграмма классов предметной области.

## 1.4 Выбор методов решения задачи и разработка основных алгоритмов предметной области

Как уже было сказано выше булева функция строится на основе такой структуры, как дерево. Данный выбор структуры обусловлен тем, что с его помощью значительно проще проводить с деревом операции преобразования и выводить функцию в полноценную строку, а также дерево даёт возможность легко сгенерировать и соединить нужные ноды, так как оно имеет ветвления и строится как пирамида – сверху вниз, от корня к листьям – в данном случае, что позволяет собирать булеву функцию как конструктор в зависимости от параметров пользователя, в то время, как для вставления нового элемента, допустим, в массив нужно проводить сдвиги, проходя по всему массиву или его части по нескольку раз, таким образом вычислительная сложность будет составлять О(n2). Со списковой структурой работать тоже не получится из-за её линейности и невозможности ветвления.

Дерево булевой функции является классом, у которого есть свои поля и методы. В полях содержится количество операций и переменных функции, её СДНФ, СКНФ, таблица значений функции при всех значениях переменных, минимальные ДНФ и КНФ, полином Жегалкина, в который переводится данная функция, фиктивная переменная этой функции (если есть).

Методы же дерева булевой функции генерируют само дерево функции, просчитывают значения её полей, а также формируют вывод полей в формате, который будет понятен пользователю.

Задания созданы классами и убраны в список из-за того, что у каждого задания есть свои методы, которые сверяются с разными полями булевой функции и имеют свои отдельные проверки ответов пользователя, что значительно облегчает выбор одного, некоторых или всех типов заданий одновременно.

Также существует отдельная функция, работающая непосредственно с пользовательским интерфейсом, что позволяет не нарушать целостность компонентов приложения.

Основным алгоритмом является алгоритм генерации дерева булевой функции. На нём строится всё остальное, так как после построения дерева, с ним могут работать уже другие методы этого дерева (вычисление значений функции при разных значениях переменных, приведение функции к СКНФ и СДНФ и так далее). Также именно после построения дерева и вычисления значений его полей, методы заданий могут с ними взаимодействовать.

# 2 Проектирование структуры и компонентов программного продукта

## 2.1 Разработка диаграммы состояний интерфейса

Для данного тренажёра был выбран вид интерфейса, состоящий из одной страницы с добавляющимися элементами.

Разработанная диаграмма интерфейса представлена на рисунке 3.

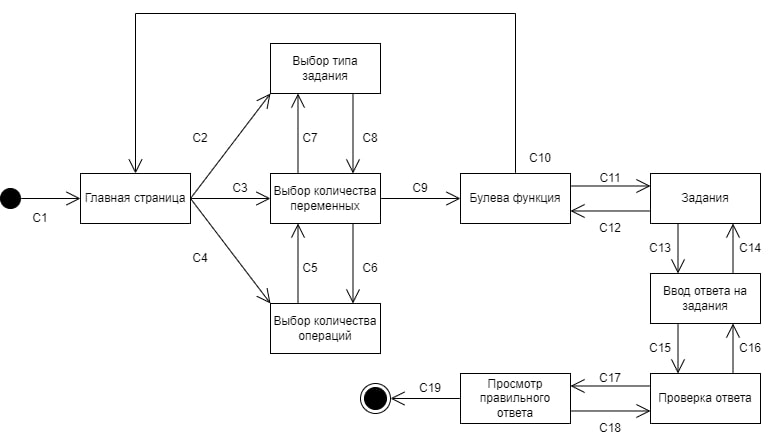


Рисунок 3 – Диаграмма состояний интерфейса

С1 – отображение главной страницы;

С2 – переход на выбор типов заданий;

С3 – переход на выбор количества переменных;

С4 – переход на выбор количества операций;

С5 – переход на выбор количества переменных;

С6 – переход на выбор количества операций;

С7 – переход на выбор типов заданий;

С8 – переход на выбор количества переменных;

С9 – генерация булевой функции;

С10 – возвращение на главную страницу;

С11 – переход к заданиям;

С12 – возвращение к булевой функции;

С13 – переход к вводу ответа на задание;

С14 – возвращение к другим заданиям;

С15 – переход к проверке ответа;

С16 – возвращение к вводу ответа на задание;

С17 – открытие правильного ответа;

С18 – скрытие правильного ответа;

С19 – выход из приложения.

## 2.2 Разработка структурной схемы программного продукта

Данное приложение проектируется по объектному подходу, из чего следует, что должна быть разработана его структурная схема (объектная декомпозиция).

Так как в приложении у нас имеются несколько частей взаимодействия с пользователем, перечислим каждую из них, и за что они отвечают:

- главное меню:

а) выбор типа задания;

б) выбор количества переменных;

д) выбор количества операций;

- задания:

а) ввод ответа;

б) проверка ответа:

б.1) просмотр правильного варианта ответа.

Главное меню активирует выбор параметров пользователем.

Выбор параметров в свою очередь активирует генерацию функции и заданий. Функция генерируется внутри самого приложения, как и её СДНФ, СКНФ и так далее.

Задания высвечиваются пользователю после генерации функции и после ввода ответа пользователя активируют проверку, причём каждое свою по отдельности, без привязки друг к другу.

Сама проверка вне зависимости от правильности введённого пользователем ответа активирует просмотр правильного варианта ответа, который остаётся скрытым, пока пользователь не нажмёт на него.

Разработанная в связи с анализом частей, активирующих и использующих друг друга, структурная схема (объектная декомпозиция) представлена на рисунке 4.

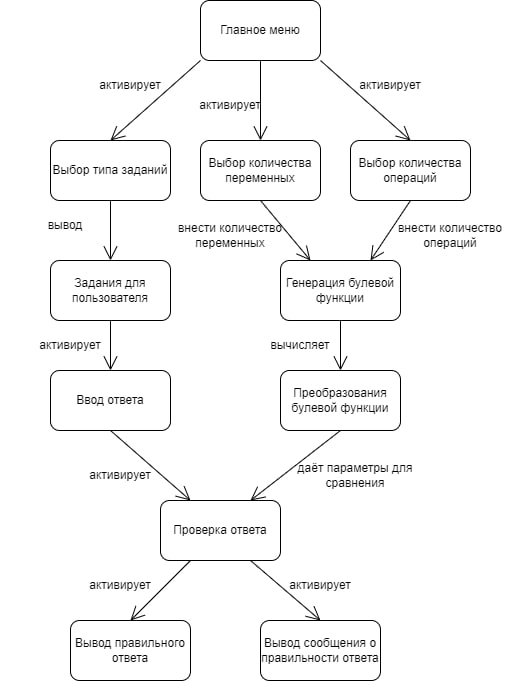


Рисунок 4 – Структурная схема программного продукта (объектная декомпозиция)

## 2.3 Разработка диаграммы классов и алгоритмы методов классов

Так как данное приложение реализовано с помощью технологии ООП, при его разработке использовались классы, иерархия которых показана на диаграмме классов (рисунок 5).

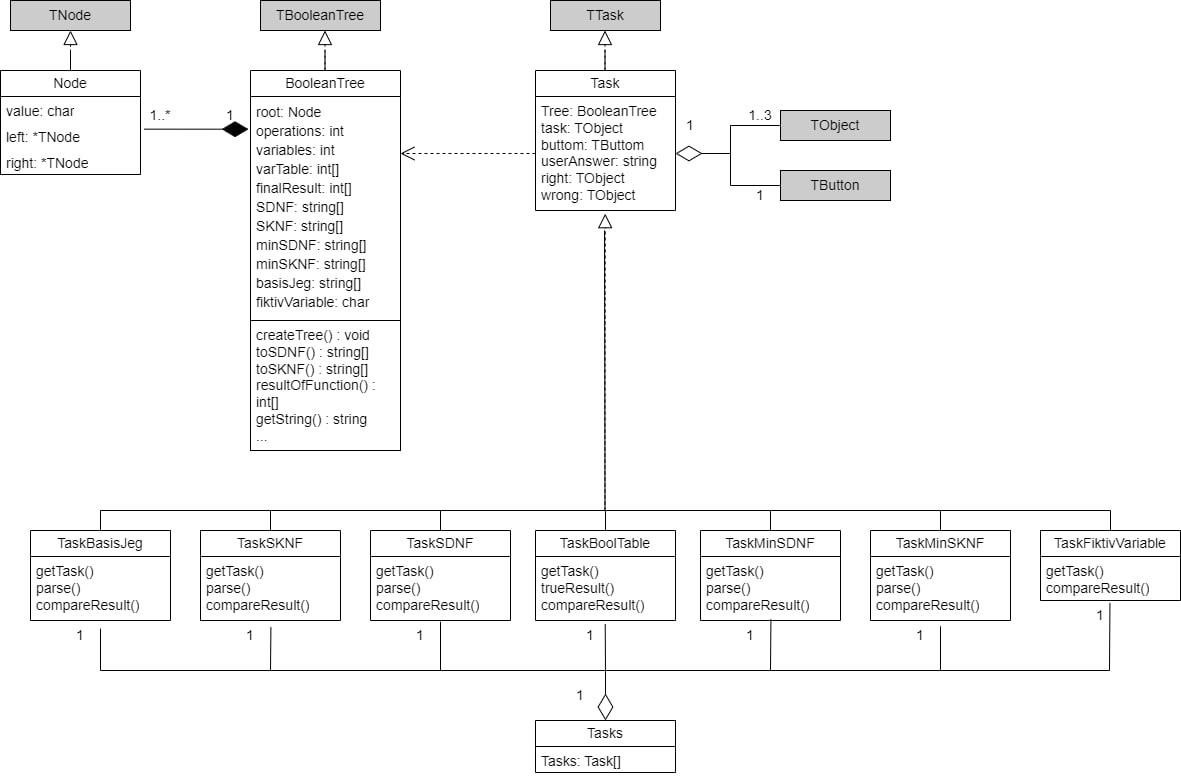


Рисунок 5 – Диаграмма классов

Схемы алгоритмов некоторых методов представлены на рисунках 6-7.

На рисунке 6 представлена схема алгоритма метода дерева булевой функции createTree(). Данный метод создаёт само дерево булевой функции, а также вызывает методы, просчитывающие значение полей дерева. На выходе метода мы получаем полноценное дерево булевой функции и указатель на его корень. Параметрами данного метода являются выбранные пользователем значения количества переменных и операций в функции. Данный метод вызывается при нажатии пользователем кнопки «Сгенерировать». Переменные алгоритма и их назначения указаны в таблице 5. Полный текст метода представлен в приложении B (страница 55).

Пользователь нажимает кнопку «Сгенерировать», в метод передаются выбранное пользователем количество переменных и операций. Эти числа присваиваются полям variables и operations дерева. Далее вызывается функция setVariables(), которая присваивает значение полю varTable дерева, которое в свою очередь является массивом хэшей со значениями, ключами к которым являются названия переменных функции. Далее создаётся нода со случайным значением из константного массива символов операций. Если значение равно “NOT”, то левая ветвь такой ноды будет указывать на ноду со значением “null”. Если же значение не “NOT”, то обе ветви ноды указывают на ноды с пустыми значениями Node. Счётчик операций уменьшается на единицу. Далее пустые ноды загружаются в очередь и начинается цикл, который не закончится, пока numOperations не станет равным нулю.

В данном цикле ноды достаются из очереди, им присваиваются значения из константного списка символов операций. Если значение операции “NOT”, в очередь добавляется только его пустая правая ветвь, левой ветви присваивается значение “null”. Счётчик при этом не уменьшается. Если значение не “NOT”, счётчик уменьшается на единицу и обе пустые ветви добавляются в очередь.

После выхода из цикла, у нас у дерева остаются только незаполненные листья. Их в счётном цикле заполняем значениями из константного списка символов переменных.

Далее уже вызываются процедуры и функции, заполняющие поля дерева.

Таблица 5 – Переменные алгоритма метода createTree() и их назначение

|  |  |
| --- | --- |
| Переменная | Назначение |
| queue | Это переменная-массив, которая используется как очередь при добавлении новых нод к дереву. |

Продолжение таблицы 5

|  |  |
| --- | --- |
| newNode | Переменная типа Node, создаваемая для присоединения новых нод к дереву и присваивания им значений операций или переменных. |
| numOperators | Счётчик, считающий количество операций, которые осталось внести в дерево булевой функции. |

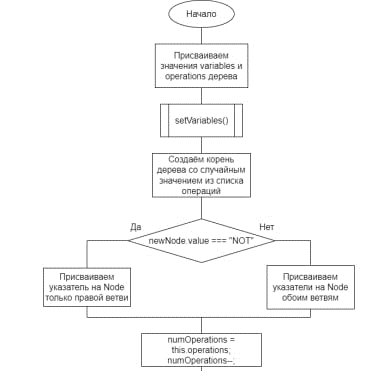


Рисунок 6 – Схема алгоритма метода createTree()

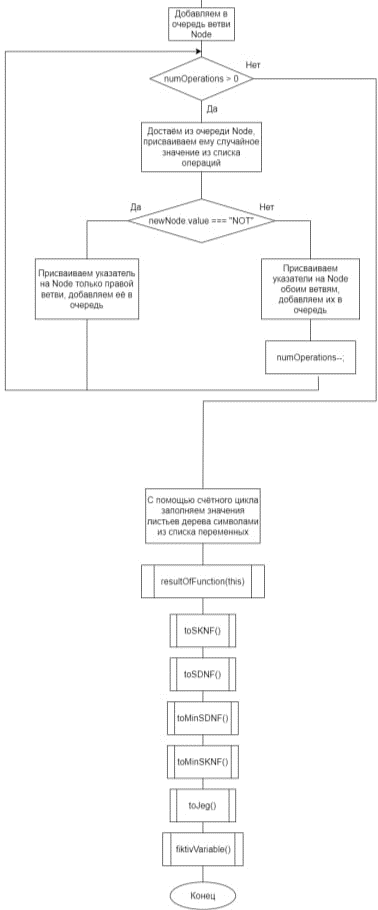


Рисунок 6 - Продолжение

На рисунке 7 представлена схема алгоритма метода compareResult() класса TaskBoolTable. Данный метод предназначен для сравнения введённых пользователем ответов с полем finalResult дерева булевой функции. Входными данными алгоритма являются введённые пользователем значения. Выходные данные – это сообщение о правильности ответа, которое выводится на экран пользователю, и скрытый правильный ответ, который пользователь может сам открыть. Данный метод не использует формальные параметры. Он вызывается при нажатии пользователем кнопки «Проверить». Переменные, используемые в методе, указаны в таблице 6. Полный текст метода представлен в приложении B (страница 57).

Пользователь нажимает кнопку «Проверить» и метод тут же считывает введённый ответ в переменную userAnswer. В отдельной функции checkAnswer() проверяется корректность значений (нет других символов кроме 1 или 0). Далее с помощью счётного цикла введённый массив строк преобразуется в массив целых чисел и помещается в переменную comparedResult. После этого переменной key присваивается значение true и начинается счётный цикл, в котором сравниваются значения comparedResult и поля дерева функции finalResult. Если есть хоть одно несовпадение, key становится равным false. Далее проверяется значение key. Если оно true, пользователю выводится надпись «Правильно», если оно false, надпись гласит «Неправильно».

Далее вне зависимости от правильности ответа, пользователю выводится скрытый правильный ответ, который он может при желании посмотреть.

Таблица 6 – Переменные алгоритма compareResult() и их значение

|  |  |
| --- | --- |
| Название переменной | Назначение |
| userAnswer | Переменная типа string, в которую загружается ответ пользователя. |
| comparedResult | Переменная-массив целых чисел для сравнения с полем finalResult дерева БФ. |
| key | Переменная типа bool. Нужна для проверки совпадения всех значений comparedResult и finalResult. |

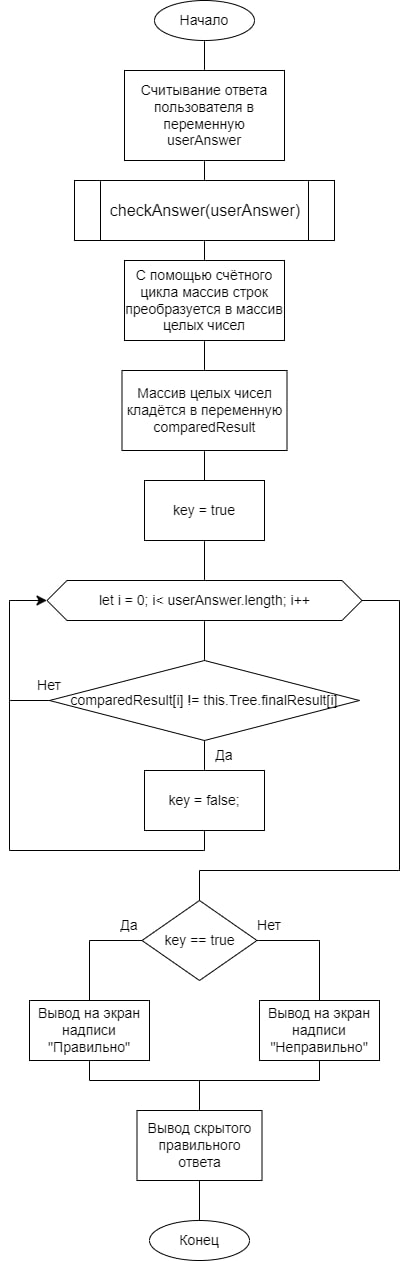


Рисунок 7 – Схема алгоритма метода compareResult()

## 2.4 Разработка форм интерфейса

Разработанные формы интерфейса проекта показаны на рисунках 8-15.

На главной странице (главном меню) пользователь видит список с множественным выбором типов заданий (рисунок 8). Он может выбрать один, несколько или все типы заданий.

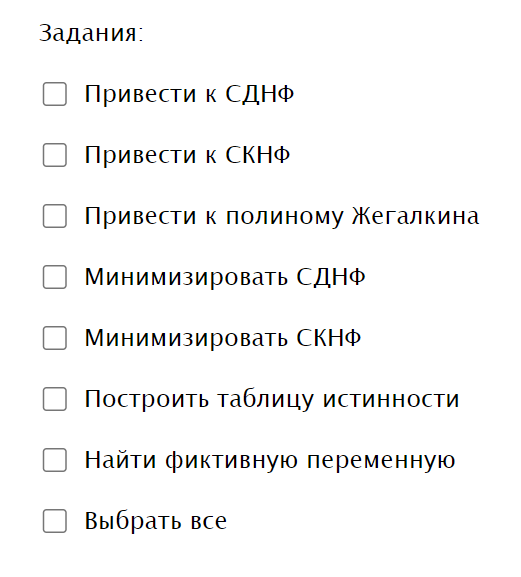


Рисунок 8 – Список с выбором типов заданий

Далее под этим списком располагаются две строчки с выбором количества переменных и операций, а также кнопка «Сгенерировать» для генерации функции (рисунок 9).

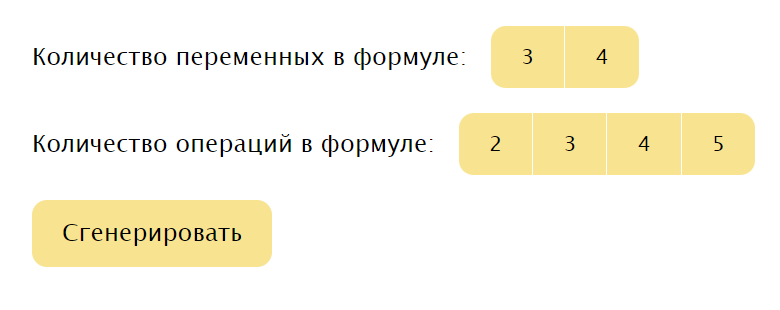


Рисунок 9 – Выбор количества переменных и операций в функции

После нажатия кнопки «Сгенерировать», ниже данной кнопки появляется сама функция (рисунок 10), а под ней располагаются выбранные задания.

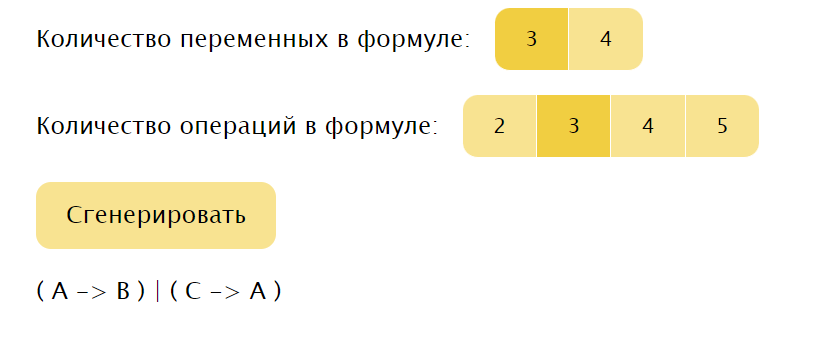


Рисунок 10 – Сгенерированная функция с выбранными параметрами

Задания по преобразованию функции в СДНФ, в СКНФ, приведение её к минимальным ДНФ и КНФ и к полиному Жегалкина выглядят одинаково: текст задания, поле для записи ответа, пример записи ответа и кнопка «Проверить» (рисунок 11).

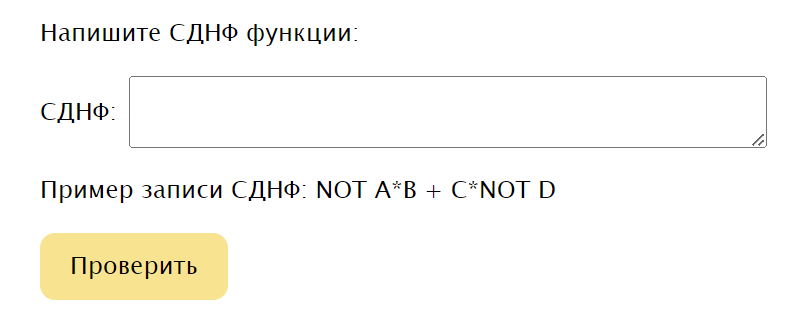


Рисунок 11 – Задание на приведение функции к СДНФ

Задания на таблицу истинности и на определение фиктивной переменной выглядят по-другому. Задание на таблицу истинности представляет из себя табличку, в строках которой под нужными переменными записаны все их значения, в конце строк есть поле для ввода значения функции при данной комбинации переменных (рисунок 12).

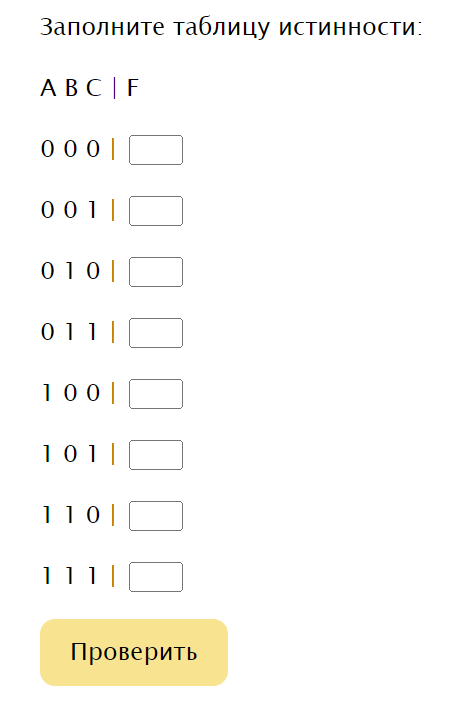


Рисунок 12 – Задание на таблицу истинности

Задание на фиктивную переменную выглядит так же, как и на СДНФ, только поле ввода ответа меньше (рисунок 13).

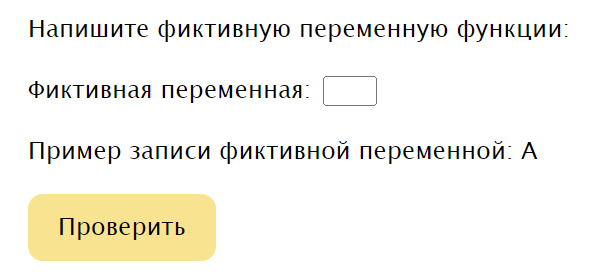


Рисунок 13 – Задание на фиктивную переменную

Далее пользователь вводит ответ и нажимает кнопку «Проверить», и вне зависимости от того, правильный он или нет, пользователю показывается скрытый правильный ответ (рисунок 14).

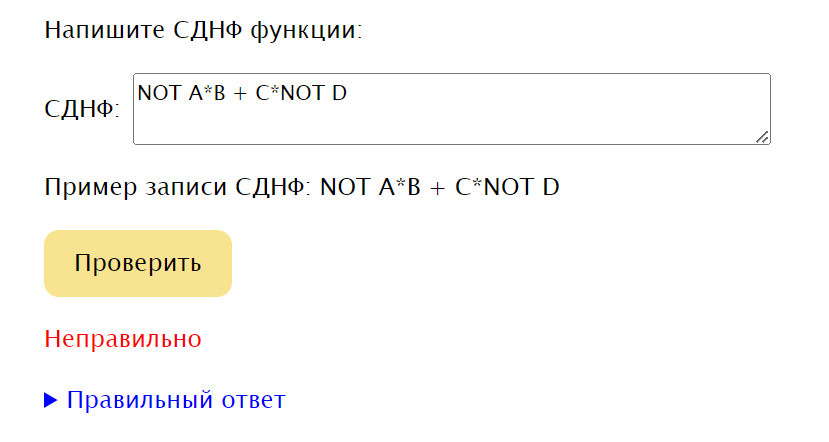


Рисунок 14 – Проверка ответа

Пользователь может раскрыть правильный ответ для просмотра, просто нажав на него, или не открывать (рисунок 15).

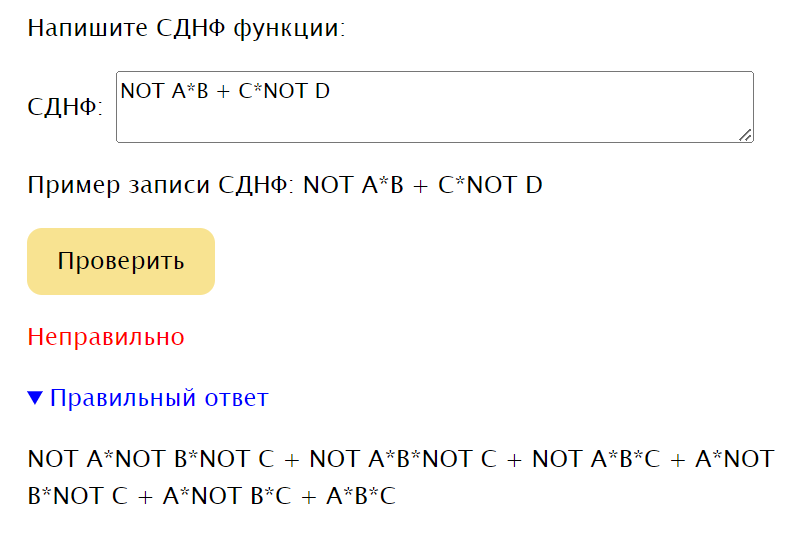


Рисунок 15 – Раскрытие правильного ответа

Правильный ответ у задания с таблицей истинности также представлен табличкой (рисунок 16).

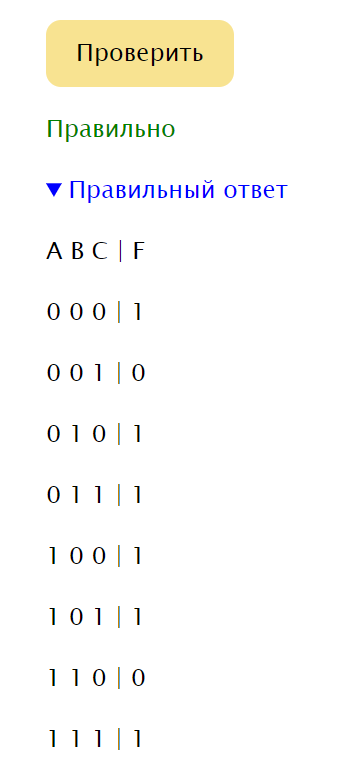


Рисунок 16 – Раскрытие правильного ответа в задании на таблицу истинности

# 3 Выбор стратегии тестирования и разработка тестов

Так как программа уже написана, и она в большей части взаимодействует с пользователем, то основным методом тестирования был выбран метод чёрного ящика, а именно метод причинно-следственных связей. Результаты данного тестирования представлены в таблице 7.

Таблица 7 – Тестирование методом причинно-следственных связей.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № теста | Назначение | Значение исходных данных | Ожидаемый результат | Реакция программы | Вывод |
| Главное меню приложения | | | | | |
| 1 | Работа множественного выбора | Пользователь ставит галочки напротив нужных типов заданий, выбирает количество переменных и операций и нажимает кнопку «Сгенерировать» | Приложение выведет на экран те задания, что указал пользователь | Приложение выводит на экран те задания, что указал пользователь | Корректная работа приложения |
| 2 | Работа множественного выбора | Пользователь не выбирает ни одного из заданий, выбирает количество переменных и операций и нажимает кнопку «Сгенерировать» | Приложение выведет на экран сообщение о том, что не выбрано ни одно из заданий. Функция не сгенерируется, задания не выведутся. | Приложение выводит сообщение об отсутствии выбора заданий. Функция не сгенерирована, задания не выведены. | Корректная работа приложения. |

Продолжение таблицы 7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 3 | Работа выбора количества переменных | Пользователь выбирает задания, выбирает количество переменных и операций и нажимает кнопку «Сгенерировать» | Приложение генерирует функцию с тем количеством переменных, которые указал пользователь | Приложение сгенерировало функцию с тем количеством переменных, что указал пользователь | Корректная работа приложения |
| 4 | Работа выбора количества переменных | Пользователь выбирает задания и количество операций, но не выбирает количество переменных и нажимает кнопку «Сгенерировать» | Приложение выводит сообщение о том, что не выбрано количество переменных | Приложение вывело сообщение о том, что не выбрано количество переменных | Корректная работа приложения |
| 5 | Работа выбора количества операций | Пользователь выбирает задания, выбирает количество переменных и операций и нажимает кнопку «Сгенерировать» | Приложение генерирует функцию с тем количеством операций, которые указал пользователь | Приложение сгенерировало функцию с тем количеством операций, что указал пользователь | Корректная работа приложения |

Продолжение таблицы 7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 6 | Работа выбора количества операций | Пользователь выбирает задания и количество переменных, но не выбирает количество операций и нажимает кнопку «Сгенерировать» | Приложение выводит сообщение о том, что не выбрано количество операций | Приложение вывело сообщение о том, что не выбрано количество операций | Корректная работа приложения |
| 7 | Работа кнопки «Сгенерировать» | Пользователь выбирает параметры и нажимает кнопку «Сгенерировать» | Приложение генерирует функцию и выводит её на экран | Приложение сгенерировало функцию и вывело её на экран | Корректная работа приложения |
| Часть страницы с заданиями | | | | | |
| 8 | Работа проверки ответа | Пользователь вводит ответ в поле и нажимает кнопку «Проверить» | Приложение обрабатывает ответ и выводит сообщение об его правильности на экран | Приложение обработало ответ и вывело сообщение об его правильности | Корректная работа приложения |
| 9 | Работа проверки ответа | Пользователь ничего не вводит в поле ответа и нажимает кнопку «Проверить» | Приложение выводит сообщение о том, что поля ответа пустое | Приложение вывело сообщение о том, что поле ответа пустое | Корректная работа приложения |
| 11 | Раскрытие правильного ответа | Пользователь открывает флажок «Правильный ответ» | На экран выводится скрытый ранее правильный ответ | На экран вывелся скрытый ранее правильный ответ | Корректная работа приложения |

Продолжение таблицы 7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 10 | Вывод правильного ответа | Пользователь вводит ответ и нажимает кнопку «Проверить» | Несмотря на ответ пользователя, выведется флажок просмотра правильного ответа | Приложение выводит флажок просмотра правильного ответа вне зависимости от правильности ответа пользователя | Корректная работа приложения |

На части страницы с заданиями пользователь имеет возможность вводить текст. Для тестирования работы данной части страницы выделим классы эквивалентности (таблица 8).

Таблица 8 – Эквивалентное разбиение

|  |  |  |
| --- | --- | --- |
| Входное условие | Правильные классы эквивалентности | Неправильные классы эквивалентности |
| Часть страницы с заданиями | | |
| Ответ пользователя на задание с фиктивной переменной | Введена переменная из возможных | Введён другой символ |
| Ответ пользователя на преобразование к СДНФ и минимальной ДНФ, СКНФ и минимальной КНФ | Введена корректная строка, содержащая операции отрицания, конъюнкции и дизъюнкции, а также переменные в зависимости от количества выбранных. | Введена другая последовательность |

Продолжение таблицы 8

|  |  |  |
| --- | --- | --- |
| Ответ пользователя на преобразование в полином Жегалкина | Введена корректная строка, содержащая сложение по модулю два и переменные в зависимости от количества выбранных. | Введена другая последовательность |
| Ответ пользователя на задание с таблицей истинности | Введены корректные значения: 0 или 1 | Введены другие символы |

В таблице 9 приведены результаты тестирования методом эквивалентных разбиений.

Таблица 9 – Тестирование методом эквивалентных разбиений.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер теста | Проверяемый параметр | Проверяемый класс | Ожидаемый результат | Полученный результат |
| Часть страницы с заданиями | | | | |
| 1 | Ответ пользователя на задание с СДНФ, СКНФ, минимальными ДНФ и КНФ | Правильный класс (корректная строка) | Отсутствие всплывающего сообщения о том, что последовательность символов неправильная | Сообщение о неправильной последовательности не всплывает |
| 2 | Ответ пользователя на задание с СДНФ, СКНФ, минимальными ДНФ и КНФ | Неправильный класс (другая последовательность символов) | Всплывающее сообщение о неправильной последовательности | Сообщение о неправильной последовательности всплывает |

Продолжение таблицы 9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | Ответ пользователя на задание с полиномом Жегалкина | Правильный класс (корректная строка) | Отсутствие всплывающего сообщения о том, что последовательность символов неправильная | Сообщение о неправильной последовательности не всплывает |
| 4 | Ответ пользователя на задание с полиномом Жегалкина | Неправильный класс (другая последовательность символов) | Всплывающее сообщение о неправильной последовательности | Сообщение о неправильной последовательности всплывает |
| 5 | Ответ пользователя на задание с таблицей истинности | Правильный класс (введены 0 или 1) | Нет всплывающего сообщения о некорректных символах | Сообщение о некорректных символах не всплывает |
| 6 | Ответ пользователя на задание с таблицей истинности | Неправильный класс (введены другие символы) | Всплывающее сообщение о некорректных символах | Сообщение о некорректных символах всплывает |
| 7 | Ответ пользователя на задание с фиктивной переменной | Правильный класс (введена переменная из возможных) | Нет всплывающего сообщения о некорректных символах | Сообщение о некорректных символах не всплывает |
| 8 | Ответ пользователя на задание с фиктивной переменной | Неправильный класс (введён другой символ) | Всплывающее сообщение о некорректных символах | Сообщение о некорректных символах всплывает |

# ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы было спроектировано и реализовано приложение-тренажёр для выполнения операций над булевыми функциями. Разработанная система позволяет сгенерировать функцию по заданным параметрам в реальном времени и проверить корректность выполнения операций над данной функцией, что поможет при подготовке к экзамену и контрольным мероприятиям по дискретной алгебре.

Данную систему можно улучшить и расширить. В будущих версиях будет добавлено:

- возможность авторизации пользователей для составления рейтинга решённых заданий;

- возможность добавления своих функций и заданий;

- возможность преобразования введённых функций;

В ходе разработки был изучен язык JavaScript, приобретён опыт проектирования программных продуктов, работы с системой контроля версий Git, составления технической документации, а также получены навыки отладки приложения и тестирования.

СПИСОК ИСТОЧНИКОВ

1 Иванова Г.С. – Технология программирования: учебник / Г.С. Иванова. – 3-е изд., стер. – М. : КНОРУС, 2016. – 334 с. – (Бакалавриат)

1. Иванова Г.С., Ничушкина Т.Н., Пугачёв Е.К., Самарёв Р.С., Фетисов М.В. – Методические указания по выполнению курсовой работы по дисциплине «Технология разработки программных систем»: Электронное учебное издание. – МГТУ им. Н. Э. Баумана, 2021. – 43 с.
2. Прокопенко Н.Ю. - Математическая логика и булевы функции: учеб.-метод. пос. / Н.Ю. Прокопенко; Нижегор. гос. архитектур. - строит. ун-т – Н. Новгород: ННГАСУ, 2021. – 107 с.
3. Современный учебник JavaScript [Электронный ресурс] URL:

<https://learn.javascript.ru/>

1. Иванова Г.С., Ничушкина Т.Н., Пугачёв Е.К. - Методы обработки данных и оценки программ : учебное пособие / Г. С. Иванова, Т. Н. Ничушкина, Е. К. Пугачев. — Москва : Издательство МГТУ им. Н. Э. Баумана, 2020. — 72, [2] с. : ил.

# ПРИЛОЖЕНИЕ А

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

Листов 9

**Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Информационные системы и сети

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01

УЧЕБНЫЙ ТРЕНАЖЁР ПО БУЛЕВЫМ ФУНКЦИЯМ «БУЛЕАН»

Техническое задание на курсовую работу

по дисциплине Технология разработки программных систем

Листов 9

Студент гр. ИУ6-51Б \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**М. А. Тарасова**

(Группа) (Подпись, дата) (И.О. Фамилия)

Руководитель курсовой работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **А. М. Минитаева**

(кандидат технических наук, доцент (Подпись, дата) (И.О. Фамилия)

доцент кафедры Информационных

технологий)

Москва, 2023

1 ВВЕДЕНИЕ

Настоящее техническое задание распространяется на разработку учебного тренажёра по булевым функциям «Булеан» [УТБФБ], используемой для тренировки студентами навыка приведения булевых функций к КНФ, ДНФ, СКНФ и СДНФ, и предназначенной для закрепления знаний дискретной математики.

Актуальность данного тренажёра состоит в том, что студентам, изучающим дискретную математику, иногда нужно закрепить пройденный материал или подготовиться к контрольной, включающей тему булевых функций, и для того, чтобы они долго не искали задачки на повторение и ответы к ним, будет разработан специальный учебный тренажёр.

Данная разработка является одной из уникальных, так как данный тренажёр в отличие от калькулятора булевых функций несёт в себе цель обучить студента самому проводить операции с булевыми функциями.

Аналогом данного тренажёра является учебный тренажёр по булевым функциям на гиперкубе, в отличие от которого данный тренажёр не будет использовать гиперкуб.

2 ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

Учебный тренажёр по булевым функциям «Булеан» разрабатывается по тематике кафедры ИУ6 «Компьютерные системы и сети» факультета ИУ «Информатика и системы управления» МГТУ им. Н.Э. Баумана.

3 НАЗНАЧЕНИЕ РАЗРАБОТКИ

Основное назначение учебного тренажёра по булевым функциям «Булеана» заключается в закреплении навыка студентов приводить булевы функции к СКНФ и СДНФ и минимизировать СДНФ и СКНФ.

4 ТРЕБОВАНИЯ К ПРОГРАММНОМУ ИЗДЕЛИЮ

* 1. Т р е б о в а н и я к ф у н к ц и он а л ь н ы м х а р а к т е р и с т и к а м
     1. В ы п о л н я е м ы е ф у н к ц и и
        1. Д л я п о л ь з о в а т е л я :
* Выбор типа задания: приведение функции к СДНФ/ СКНФ, представление булевой функции в виде полинома Жегалкина, минимизация СДНФ, минимизация СКНФ, построение таблицы истинности. Возможность выбрать все варианты заданий для одной булевой функции, поиск фиктивной переменной;
* Выбор количества переменных, от которых зависит функция: три или четыре;
* Выбор количества операций, из которых состоит функция: 2 – 5;
* Ввод ответа;
* Просмотр правильного ответа.
  + 1. И с х о д н ы е д а н н ы е :
* Тип задания;
* Количество переменных функции;
* Количество операций в функции;
* Ответ на задание.
  + 1. Р е з у л ь т а т ы :
* Сгенерированная функция;
* Сообщение о правильности введённого ответа;
* Показ правильного ответа.
  1. Т р е б о в а н и я к н а д е ж н ос т и
     1. Предусмотреть контроль вводимой информации.
     2. Предусмотреть блокировку некорректных действий пользователя.
  2. У с л о в и я э к с п л у а т а ц и и
     1. Условия эксплуатации в соответствии с СанПиН 2.2.2/2.4.1340-03.
     2. Специальное обслуживание не требуется
  3. Т р е б о в а н и я к с о с т а в у и п а р а м е т р а м т е х н и че с к и х с р е д с т в
     1. Минимальная конфигурация технических средств:
        1. Тип процессора ntel Pentium 4
        2. Объем ОЗУ. 4 Гб.
        3. Одна из следующих операционных систем: Windows 7, Ubuntu 10 или их более поздние версии
     2. Минимальная конфигурация сервера
        1. Операционная система из семейства Linux
        2. Тип процессора ntel Pentium 4
        3. Объём ОЗУ 2 Гб
        4. Объём ПЗУ 10 Гб
  4. Тре бова ни я к и нф орма ци онн ой и прог ра ммн ой с овмес ти мос ти
     1. Программное обеспечение должно работать под управлением операционных систем семейств Windows 7, Ubuntu 10 или их более поздних версий.

4.6 С п е ц и а л ь н ые т р е б о в ан и я

Специальные требования отсутствуют.

5 ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

* 1. Разрабатываемые программные модули должны быть самодокументированы, т.е. тексты программ должны содержать все необходимые комментарии.
  2. В состав сопровождающей документации должны входить:
     1. Расчетно-пояснительная записка на 25-30 листах формата А4 (без приложений 5.2.2, 5.2.3 и 5.2.4).
     2. Техническое задание (Приложение А).
     3. Руководство пользователя (Приложение Б).
  3. Графическая часть должна быть включена в расчетно-пояснительную записку в качестве иллюстраций:
     1. Диаграмма вариантов использования.
     2. Схемы взаимодействия объектов, объектная декомпозиция.
     3. Схема структурная программного обеспечения.
     4. Диаграммы классов предметной области и/или интерфейсной части программного обеспечения.
     5. Граф состояний интерфейса.
     6. Формы интерфейса.
     7. Схемы алгоритмов модулей (подпрограмм).
     8. Таблицы тестов.

6 СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Этап** | **Содержание этапа** | **Сроки и объем** | **Представляемые результаты** | |
| **Спецификации и программный продукт** | **Документы** |
| 1. | Выбор темы, составле ние задания, решение организационных вопросов | 1..2 недели  (10 %) | - | **Заполненный бланк задания на курсовую работу – вывешивается на сайт кафедры для получения утверждающей подписи**  **заведующего кафедрой** |
| 2. | Анализ предметной об- ласти, разработка ТЗ. Исследование методов решения, выбор основ- ных проектных решений | 3..4 недели | Результаты декомпозиции предметной области.  Эскизный проект: интерфейс, схемы, возможно, часть програм мы (выбранные готовые решения). | Фрагмент расчетно- пояснительной записки с обоснованием выбора средств и подходов к раз- работке |
| **3.** | **Сдача ТЗ** | **4 неделя**  **(25 %)** | **-** | **Техническое задание – утверждается руководителем** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Этап** | **Содержание этапа** | **Сроки и объем** | **Представляемые результаты** | |
| **Спецификации и программный продукт** | **Документы** |
| 4. | Проектирование и реа-  лизация основных ком понентов – ядра про- граммы | 5..7 недели | Технический проект  основной части: структура программы, алго- ритмы программ, описания структур данных, диаграмма классов – в зависимости от вы- бранной технологии разработки.  Программный продукт, реализующий основ- ные функции (демон- стрируется руководителю) | Фрагмент расчетно-  пояснительной записки с обоснованием разработан ных спецификаций Тексты части программно-  го продукта, реализующего основные функции. |
| **5.** | **Сдача прототипа про граммного продукта** | **7 неделя**  **(50 %)** | **Прототип программного продукта – де-**  **монстрируется руководителю** |  |
| 6. | Разработка компонентов, обеспечивающих функциональную пол-  ноту | 8..10 | Рабочий проект программы.  Готовая программа | Черновик расчетно- пояснительной записки. Тексты программного продукта. |
| **7.** | **Сдача программного продукта** | **11 неделя**  **(75 %)** | **Готовая программа – оценивается руководителем в баллах** | **-** |
| 8. | Тестирование програм мы и подготовка доку-  ментации | 12..14 | Тесты и результаты тестирования. | РПЗ и Руководство пользователя. |
| **9.** | **Оформление и сдача документации** | **14 неделя**  **(90 %)** | **–** | **Расчетно-пояснительная записка и Руководство пользователя – проверя- ются и подписываются руководителем** |
| 10. | Защита курсовой работы | 15..16 не-  дели (100%) | – | Доклад (3-5 минут). Защита курсовой работы.  Подписанная документация – вывешивается на сайт кафедры |

7 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

* 1. П о р я д о к к о н т р о л я

Контроль выполнения осуществляется руководителем еженедельно.

* 1. П о р я д о к з а щ и т ы

Защита осуществляется комиссии преподавателей кафедры.

* 1. С р о к з а щ и т ы

Срок защиты: 15-16 недели.

8 ПРИМЕЧАНИЕ

В процессе выполнения работы возможно уточнение отдельных требований технического задания по взаимному согласованию руководителя и исполнителя.

# ПРИЛОЖЕНИЕ Б

**РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

Листов 8

**Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Информационные системы и сети

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01

УЧЕБНЫЙ ТРЕНАЖЁР ПО БУЛЕВЫМ ФУНКЦИЯМ «БУЛЕАН»

Руководство пользователя

Листов 8

Студент гр. ИУ6-51Б \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **М. А. Тарасова**

(Подпись, дата) (И. О. Фамилия)

Руководитель курсовой работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **А. М. Минитаева**

(кандидат технических наук, доцент, (Подпись, дата) (И.О. Фамилия)

доцент кафедры Информационных

технологий)

Москва, 2023

ОГЛАВЛЕНИЕ

[1 Общие сведения о программном продукте 47](#_Toc154015143)

[2 Начало работы с тренажёром 47](#_Toc154015144)

[3 Инструкция по работе с приложением 4](#_Toc154015145)8

[3.1 Выбор параметров 4](#_Toc154015146)8

[3.2 Генерация функции 5](#_Toc154015147)0

[4 Завершение работы с приложением 52](#_Toc154015148)

**1 Общие сведения о программном продукте**

Тренажёр по булевым функциям «Булеан» предназначен для подготовки студентов к экзамену и другим контрольным мероприятиям по дискретной математике, содержащим задания на преобразование булевых функций. В приложении доступны выбор типов заданий, которые могут попасться на контрольном мероприятии, а также генерация самой булевой функции по выбранным пользователем параметрам, а именно по количеству переменных и операций в функции. В части с заданиями проверяются вводимые ответы, а также показывается скрытый правильный ответ.

Работа с тренажёром «Булеан» осуществляется с помощью персонального компьютера, имеющего доступ к сети Интернет.

**2 Начало работы с тренажёром**

Для начала работы с приложением необходимо иметь стабильное интернет-соединение на персональном компьютере. При загрузке приложения отобразится основная страница, на которой пользователь может увидеть список типов заданий, количество переменных и операций, которое он может выбрать, а также кнопку генерации заданий и булевой функции.

Пользователю не нужно авторизоваться, он может сразу приступить к выполнению заданий.

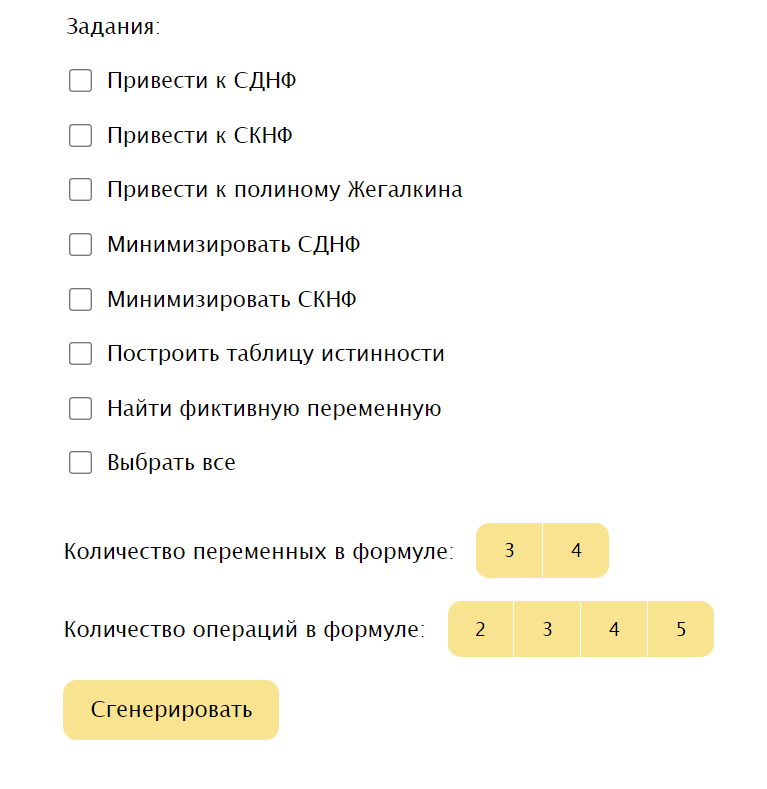


Рисунок Б.1 – Внешний вид основной страницы

**3 Инструкция по работе с приложением**

**3.1 Выбор параметров**

На основной странице пользователь может выбрать типы заданий (один или несколько), а также количество переменных и операций в будущей функции.

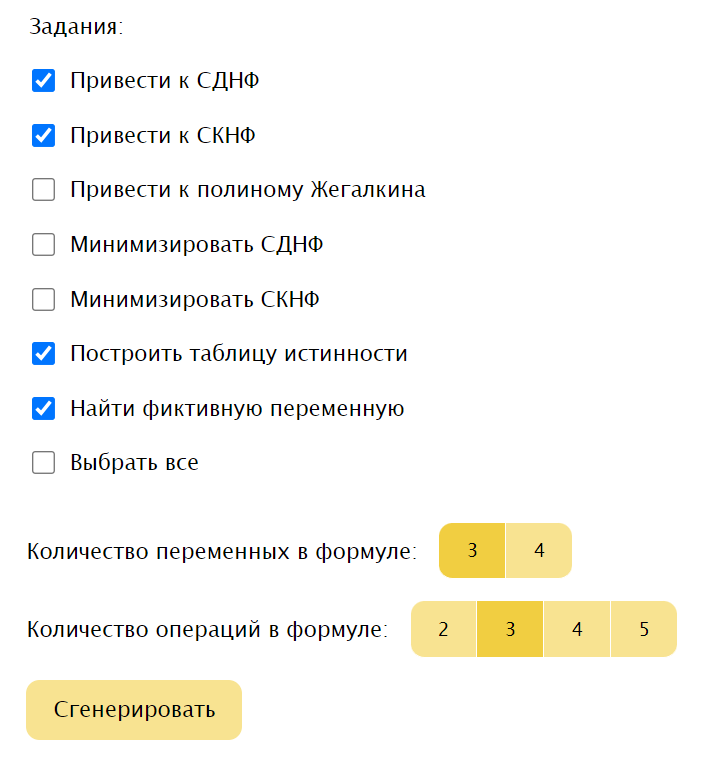


Рисунок Б.2 – Выбор параметров заданий и функции

В случае если хотя бы один параметр останется не выбранным, всплывёт сообщение либо о том, что не выбраны задания, либо о том, что не выбрано количество переменных или операций, в зависимости от пустых параметров.

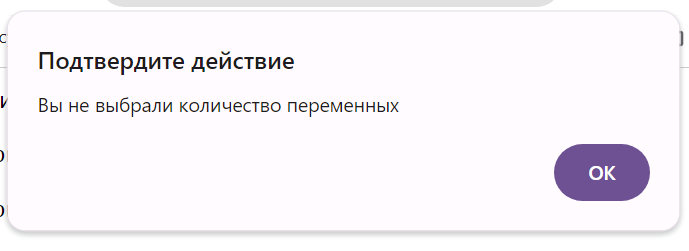


Рисунок Б.3 – Предупреждение о пустом параметре

**3.2 Генерация функции**

Чтобы сгенерировать функцию и задания, необходимо выбрать все параметры и нажать «Сгенерировать». После этого под нажатой кнопкой появится функция, а под функцией выбранные задания.

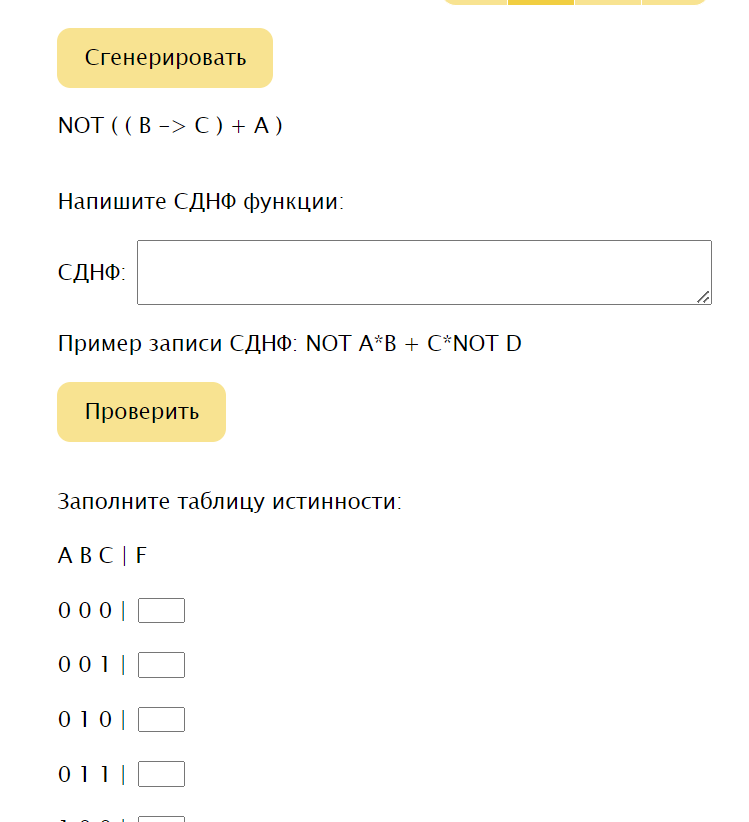


Рисунок Б.4 – Появившиеся задания

После появления заданий, можно приступить к их выполнению. Занести ответы можно в поля ответов. Для проверки нажать кнопку «Проверить».

Если ответ правильный, высветится надпись «Правильно» (рисунок Б.5). Если ответ неверный, высветится надпись «Неправильно» (рисунок Б.6).

Вне зависимости от правильности ответа пользователю будет предложено посмотреть правильный ответ (рисунок Б.7).

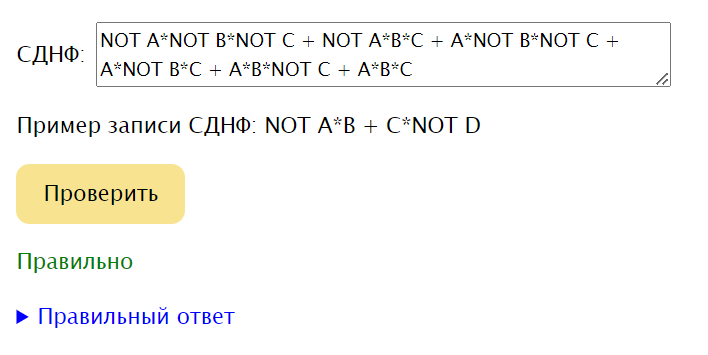


Рисунок Б.5 – Правильный ответ

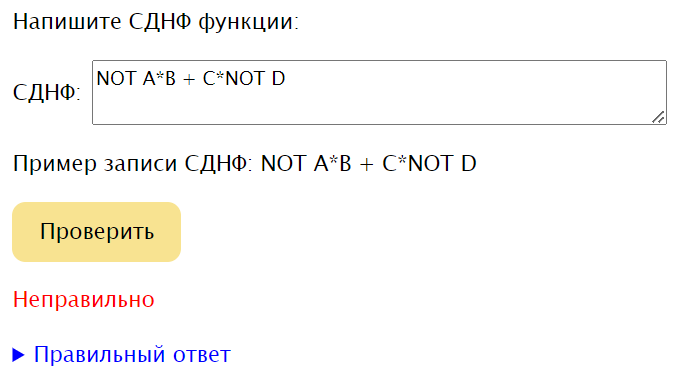


Рисунок Б.6 – Неправильный ответ

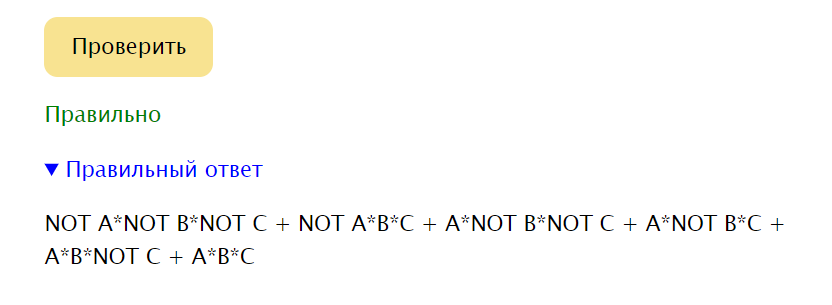


Рисунок Б.7 – Просмотр правильного ответа

**4 Завершение работы с приложением**

Для завершения работы с приложением достаточно закрыть вкладку веб-браузера. После выхода параметры обнулятся и функция с заданиями пропадёт.

# ПРИЛОЖЕНИЕ В

**ФРАГМЕНТ ИСХОДНОГО ТЕКСТА ПРОГРАММЫ**

Листов 5

**Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Информационные системы и сети

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01

УЧЕБНЫЙ ТРЕНАЖЁР ПО БУЛЕВЫМ ФУНКЦИЯМ «БУЛЕАН»

Фрагмент исходного текста программы

Листов 5

Студент гр. ИУ6-51Б \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **М. А. Тарасова**

(Подпись, дата) (И. О. Фамилия)

Руководитель курсовой работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **А. М. Минитаева**

(кандидат технических наук, доцент, (Подпись, дата) (И. О. Фамилия)

доцент кафедры Информационных

технологий)

Москва, 2023

В качестве примеров приведены тексты методов классов BooleanTree –createTree() и TaskBoolTable – compareResult().

Листинг В.1 – BooleanTree.createTree()

    //Функция собирает дерево

    createTree(numVariables, numOperators){

        this.variables = numVariables;

        this.setVariables(this.variables);

        this.operations = numOperators;

        let queue = [];

        let newNode = new Node(operators[Math.floor(Math.random() \* operators.length)]);

        numOperators--;

        this.root = newNode;

        if(newNode.value === "NOT")

        {

            newNode.right = new Node(null);

            queue.push(newNode.right)

        }

        else {

            newNode.left = new Node(null);

            newNode.right = new Node(null);

            queue.push(newNode.left, newNode.right);

        }

        while(numOperators){

            newNode = queue.shift();

            newNode.value = operators[Math.floor(Math.random()\*operators.length)];

            if(operators.includes(newNode.value))

            {

                if(newNode.value === "NOT")

                {

                    newNode.right = new Node(null);

                    queue.push(newNode.right);

                }

                else{

                    numOperators--;

                    newNode.left = new Node(null);

                    newNode.right = new Node(null);

                    queue.push(newNode.left, newNode.right);

                }

            }

        }

        let k = 0;

        for(let i = 0; i < queue.length; i++)

        {

            if(k === this.variables)

            k = 0;

            queue[i].value = usedVariables[k];

            k++;

        }

        this.resultOfFunction(this);

        console.log(this.finalResult);

        this.toSDNF();

        this.toSKNF();

this.toMinSDNF();

this.toMinSKNF();

this.toJeg();

this.fiktivVariable();

    }

Листинг В.2 – TaskBoolTable.compareResult()

    compareResult(){

        let userAnswer = document.getElementsByClassName('SDNF')[0].value;

        console.log(userAnswer);

checkAnswer(userAnswer);

        userAnswer = this.parse(userAnswer);

        let counter = 0;

        if(userAnswer.length === this.Tree.SDNF.length)

        for(let i = 0; i < userAnswer.length; i++)

        {

            if(userAnswer[i] == this.Tree.SDNF[i])

            counter++

        }

        if(this.task.lastChild.previousSibling === this.right || this.task.lastChild.previousSibling === this.wrong)

        {

            this.task.lastChild.previousSibling.remove();

            this.task.lastChild.remove();

        }

        if(counter === this.Tree.SDNF.length)

        this.task.append(this.right);

        else

        this.task.append(this.wrong);

        let table = document.createElement('details');

        table.innerHTML = "<summary style='color: blue'>Правильный ответ</summary>";

        table.innerHTML += '<p>' + this.Tree.SDNF.join(' + ') + '</p>';

        if(this.task.lastChild != table)

        this.task.append(table);

    }

Полный текст проекта размещён по ссылке: <https://github.com/Lily55/TermWork_TRPS>